June 2018 Volume 13 Number 1 ISSN 1557-6418

JAIF 18

Journal of Advances in Information Fusion

A semi-annual archival publication of the International Society of Information Fusion

Regular Papers Page

Statistical Efficiency of Simultaneous Target State and Sensor Bias Estimation 3
Djedjiga Belfadel, Fairfield University, USA
Yaakov Bar-Shalom, University of Connecticut, USA
Peter Willett, University of Connecticut, USA

Distributed Fusion Algorithm for Passive Localization of Multiple Transient Emitters 13
Wenbo Dou, University of Connecticut, USA
Yaakov Bar-Shalom, University of Connecticut, USA
Lance Kaplan, U.S. Army Research Laboratory, USA
Jemin George, U.S. Army Research Laboratory, USA

Maneuvering Target Tracking Using Continuous Wave Bistatic Sonar with Propagation Delay 36
Rong Yang, DSO National Laboratories, Singapore
Yaakov Bar-Shalom, University of Connecticut, USA
Claude Jauffret, TOULON Cedex, France
Annie-Claude Pérez, TOULON Cedex, France
Gee Wah Ng, DSO National Laboratories, Singapore

Approaches to Obtain a Large Number of Ranked Solutions to 3-Dimensional Assignment Problems 50
Lingyi Zhang, University of Connecticut, USA
David Sidoti, University of Connecticut, USA
Spandana Vallabhaneni, University of Connecticut, USA
Krishna R. Pattipati, University of Connecticut, USA
David A. Castañón, Boston University, USA

General Multivariate Polynomial Target Localization and Initial Estimation 68
David Frederic Crouse, Naval Research Laboratory, USA

Mutual Information for Optimal Asset Allocation 92
K. C. Chang, George Mason University, USA
Zhi Tian, George Mason University, USA

Semi-Blind Secure Watermarking based on integration of AES and ECC in DCT Domain 106
Vineet Mehan, Maharaja Agrasen University, India

Volumes 1–12 Index 123

INTERNATIONAL SOCIETY OF INFORMATION FUSION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied INFORMATION FUSION technologies. Technical areas of interest include target tracking, detection theory, applications for information fusion methods, image fusion, fusion systems architectures and management issues, classification, learning, data mining, Bayesian and reasoning methods.

Journal of Advances in Information Fusion (ISSN 1557-6418) is published semi-annually by the International Society of Information Fusion. The responsibility for the contents rests upon the authors and not upon ISIF, the Society, or its members. ISIF is a California Nonprofit Public Benefit Corporation at P.O. Box 4631, Mountain View, California 94040.

Statistical Efficiency of Simultaneous Target State and Sensor Bias Estimation

DJEDJIGA BELFADEL
YAAKOV BAR-SHALOM
PETER WILLETT

In this paper we provide a new methodology using an exoatmospheric target of opportunity seen in a satellites borne sensor's field of view to estimate the sensor's biases simultaneously with the state of the target. Each satellite is equipped with an Infra Red (IR) sensor that provides the Line Of Sight (LOS) measurements azimuth and elevation to the target. The measurements provided by these sensors are assumed to be noisy but perfectly associated, i.e., it is known perfectly that they belong to the same target. The evaluation of the Cramér-Rao Lower Bound (CRLB) on the covariance of the bias estimates, and the statistical tests on the results of simulations show that both the target trajectory and the biases are observable and this method is statistically efficient.

Manuscript received October 15, 2016; revised November 2, 2016; released for publication December 12, 2016.

Refereeing of this contribution was handled by Paolo Braca.

Authors' addresses: D. Belfadel, Electrical and Computer Engineering, Fairfield University, Fairfield, CT, U.S.A. (E-mail: dbelfadel@fairfield.edu). Y. Bar-Shalom and P. Willett, Electrical and Computer Engineering, University of Connecticut, Storrs, CT, U.S.A. (E-mail: {ybs, willett}@engr.uconn.edu).

I. INTRODUCTION

A space-based tracking system provides many advantages for missile defense as well as space situational awareness as a part of a system of systems that contribute to an overall picture. It can cover gaps in terrestrial radar coverage and expand the capabilities of a Ballistic Missile Defense System (BMDS), allow interceptors to engage enemy missiles earlier in their trajectories, discriminate between warheads and decoys, and provide warhead hit assessment. However, systemic errors in sensing systems hinder accurate threat identification and target state estimation, and, in this way, the space-based tracking systems present some unique challenges [7].

Multisensor systems use fusion of data from multiple sensors to form accurate estimates of a target track. To fuse multiple sensor data the individual sensor data must be expressed in a common reference frame. A problem encountered in multisensor systems is the presence of errors due to sensor bias. Bias error in a spacecraft and sensors can result from a number of different sources [8], including:

- Errors in spacecraft position (spacecraft navigation bias).
- Errors in spacecraft attitude (wheel assembly controller error, coordinate system translation round-off error).
- Errors in sensor calibration (residual pointing error, degradation of sensor alignment).
- Errors in timing caused by bias in the clocks of the sensors.

In [9] time varying bias estimation based on a nonlinear least squares formulation and the singular value decomposition using truth data was presented. However, this work did not discuss the CRLB for bias estimation. An approach using maximum a posteriori (MAP) data association for concurrent bias estimation and data association based on sensor-level track state estimates was proposed in [10] and extended in [11].

For angle-only sensors, imperfect registration leads to LOS angle measurement biases in azimuth and elevation. If not corrected, the registration errors can seriously degrade the global surveillance system performance by increasing the tracking errors and even introducing ghost targets. In [6] the effect of sensor and timing bias error on the tracking quality of a space-based IR tracking system that utilizes a Linearized Kalman Filter (LKF) for the highly non-linear problem of tracking a ballistic missile was presented. This was extended in [7] by proposing a method of using stars observed in the sensor background to reduce the sensor bias error. In [4] simultaneous sensors bias and targets position estimation using fixed passive sensors was proposed. A solution to the related observability issues discussed in [4]

TABLE I
Symbols associated with coordinate systems and measurements:

Symbol	Definition
r	The range from the sensor to the target
α, ϵ	Azimuth and Elevation angles
$\mathbf{b}$	Bias vector
ϕ, ψ, ρ	Roll, pitch and yaw
$\boldsymbol{\omega}$	Orientation of a sensor
T	A rotation matrix
x, y, z	Target Positions in Cartesian coordinates
$\dot{x}, \dot{y}, \dot{z}$	Target velocities in Cartesian coordinates
$\boldsymbol{\theta}$	Parameters vector
$\mathbf{z}$	Measurements vector
ξ, η, ζ	Sensor Locations
R	Covariance matrix
H	Jacobian matrix
F	Transition matrix

is proposed in [5] using space based sensors. In [3] a simultaneous target state and passive sensors bias estimation was proposed. However, this work did not discuss the statistical efficiency of the estimates. The new bias estimation algorithm developed in this paper is validated using a hypothetical scenario created using System Tool Kit (STK) [1]. The tracking system consists of two optical sensors (space based) tracking a ballistic target. We assume the sensors are synchronized, their locations are known, and the data association is correct and we estimate their orientation biases (assumed constant during the entire tracking time) while simultaneously estimating the state of the target (position and velocity). We evaluate the Cramér-Rao lower bound (CRLB) on the covariance of the bias estimates, which is the quantification of the available information on the sensor biases, and show via statistical tests that the estimation is statistically efficient—it meets the CRLB.

Section II presents the problem formulation and solution in detail. Section III describes the simulations performed and gives the results. Finally, Section IV discusses the conclusions and future work.

II. PROBLEM FORMULATION AND ANNOTATIONS

A. List of Symbols and Acronyms

Table I is a list of the symbols used throughout the paper. In many sections, symbols are given additional subscripts or superscripts to make them more specific.

B. Problem Formulation

In order to fuse measurements from multiple sensors, all the sensors measurements must be expressed with respect to a common frame of reference. The fundamental frame of reference used in this paper is the Earth Centered Inertial (ECI) Coordinate System.

The sensor reference frame associated with sensor platform s (measurement frame of the sensor) is defined by the orthogonal set of unit vectors $(e_{\xi_s}, e_{\eta_s}, e_{\zeta_s})$. The origin of the measurement frame of the sensor is a translation of the ECI origin, and its axes are rotated with respect to the ECI axes. The rotation between these frames can be described by a set of Euler angles. We will refer to these angles $\phi_s + \phi_s^{\mathrm{n}}$, $\rho_s + \rho_s^{\mathrm{n}}$, $\psi_s + \psi_s^{\mathrm{n}}$ of sensor s, as roll, pitch and yaw respectively, where ϕ_s^{n} is the nominal roll angle, ϕ_s is the roll bias, etc.

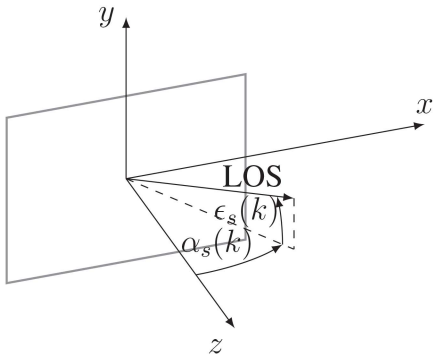


Fig. 1. Optical sensor coordinate system with the origin in the center of the focal plane.

Each angle defines a rotation about a prescribed axis, in order to align the sensor frame axes with the ECI axes. The xyz rotation sequence is chosen, which is accomplished by first rotating about the x axis by ϕ_s^{n}, then rotating about the y axis by ρ_s^{n}, and finally rotating about the z axis by ψ_s^{n}. The rotations sequence can be expressed by the matrices

$$T_s(\psi_s^{\mathrm{n}}, \rho_s^{\mathrm{n}}, \phi_s^{\mathrm{n}}) = T_z(\psi_s^{\mathrm{n}}) T_y(\rho_s^{\mathrm{n}}) T_x(\phi_s^{\mathrm{n}}) \tag{1}$$

The explicit expressions of the elements of (1) can be found in [3]. Assume there are N_S synchronized passive sensors, with known positions in ECI coordinates,

$$\boldsymbol{\xi}_s(k) = [\xi_s(k), \eta_s(k), \zeta_s(k)]', \quad s = 1, 2, \ldots, N_S, \quad k = 0, 1, 2, \ldots, K \tag{2}$$

where K is the final tracking time. The sensors get biased noisy measurements for tracking a single target at unknown positions

$$\mathbf{x}_{\mathrm{p}}(k) = [x(k), y(k), z(k)]' \tag{3}$$

also in ECI coordinates. With the previous convention, the operations needed to transform the position of the target location expressed in ECI coordinates into the sensor s coordinate system (based on its nominal orientation) is

$$\mathbf{x}_s^{\mathrm{n}}(k) = T(\boldsymbol{\omega}_s(k))(\mathbf{x}_{\mathrm{p}}(k) - \boldsymbol{\xi}_s(k)) \quad s = 1, 2, \ldots, N_S, \quad k = 0, 1, 2, \ldots, K \tag{4}$$

where $\boldsymbol{\omega}_s(k) = [\phi_s^{\mathrm{n}}(k), \rho_s^{\mathrm{n}}(k), \psi_s^{\mathrm{n}}(k)]'$ is the nominal orientation of sensor s, $T(\boldsymbol{\omega}_s(k))$ is the appropriate rotation matrix, and the translation $(\mathbf{x}_{\mathrm{p}}(k) - \boldsymbol{\xi}_s(k))$ is the difference between the vector position of the target and the vector position of the sensor s, both expressed in ECI coordinates. The superscript "n" in (8) indicates that the rotation matrix is based on the nominal sensor orientation.

Each passive sensor provides LOS measurements of the target position. As shown in Figure 1, the azimuth angle $\alpha_s(k)$ is the angle in the sensor xz plane between the sensor z axis and the line of sight to the target, while the elevation angle $\epsilon_s(k)$ is the angle between the line of sight to the target and its projection onto the xz plane, i.e.,

$$\begin{bmatrix} \alpha_s(k) \\ \epsilon_s(k) \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\dfrac{x_s(k)}{z_s(k)}\right) \\ \tan^{-1}\left(\dfrac{y_s(k)}{\sqrt{x_s^2(k)+z_s^2(k)}}\right) \end{bmatrix} \tag{5}$$

The model for the biased noise-free LOS measurements is then

$$\begin{bmatrix} \boldsymbol{\alpha}_s^b(k) \\ \boldsymbol{\epsilon}_s^b(k) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}(k), \boldsymbol{\xi}_s(k), \boldsymbol{\omega}_s(k), \mathbf{b}_s) \\ h_2(\mathbf{x}(k), \boldsymbol{\xi}_s(k), \boldsymbol{\omega}_s(k), \mathbf{b}_s) \end{bmatrix} \triangleq \mathbf{h}(\mathbf{x}(k), \boldsymbol{\xi}_s(k), \boldsymbol{\omega}_s(k), \mathbf{b}_s) \tag{6}$$

where h_1 and h_2 denote the sensor Cartesian coordinates-to-azimuth/elevation angle mapping that can be found by inserting (8) and (5) into (6)

$$\begin{bmatrix} h_1(\mathbf{x}(k), \boldsymbol{\xi}_s(k), \boldsymbol{\omega}_s(k), \mathbf{b}_s) \\ h_2(\mathbf{x}(k), \boldsymbol{\xi}_s(k), \boldsymbol{\omega}_s(k), \mathbf{b}_s) \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\dfrac{x_s^{\mathrm{b}}(k)}{z_s^{\mathrm{b}}(k)}\right) \\ \tan^{-1}\left(\dfrac{y_s^{\mathrm{b}}(k)}{\sqrt{(x_s^{\mathrm{b}}(k))^2+(z_s^{\mathrm{b}}(k))^2}}\right) \end{bmatrix} \tag{7}$$

and

$$\mathbf{x}_s^{\mathrm{b}}(k) = T(\boldsymbol{\omega}_s^b(k))(\mathbf{x}_{\mathrm{p}}(k) - \boldsymbol{\xi}_s(k)) \quad s = 1,2,\ldots,N_S, \quad k = 0,1,2,\ldots,K, \tag{8}$$

where

$$\boldsymbol{\omega}_s^b(k) = [\phi_s^{\mathrm{n}}(k) + \phi_s, \rho_s^{\mathrm{n}}(k) + \rho_s, \psi_s^{\mathrm{n}}(k) + \psi_s]' \tag{9}$$

is the biased orientation of sensor s, and the bias vector of sensor s is

$$\mathbf{b}_s = [\phi_s, \rho_s, \psi_s]' \tag{10}$$

At time k, each sensor provides the noisy LOS measurements

$$\mathbf{z}_s(k) = \mathbf{h}(\mathbf{x}_{\mathrm{p}}(k), \boldsymbol{\xi}_s(k), \boldsymbol{\omega}_s(k), \mathbf{b}_s) + \mathbf{w}_s(k) \tag{11}$$

Let $\mathbf{z}$ be an augmented vector consisting of the batch stacked measurements from all the sensors up to time K

$$\mathbf{z} = [z_1(1), z_2(1), \ldots, z_{N_S}(1), \ldots, z_1(K), z_2(K), \ldots, z_{N_S}(K)] \tag{12}$$

and

$$\mathbf{w}_s(k) = [w_s^{\alpha}(k), w_s^{\epsilon}(k)]' \tag{13}$$

The measurement noises $\mathbf{w}_s(k)$ are zero-mean, white Gaussian with

$$R_s = \begin{bmatrix} (\sigma_s^{\alpha})^2 & 0 \\ 0 & (\sigma_s^{\epsilon})^2 \end{bmatrix} \quad s = 1,2,\ldots,N_S \tag{14}$$

and are assumed mutually independent. The problem is to estimate the bias vectors for all sensors and the state vector (position and velocity) of the target of opportunity, i.e.,

$$\boldsymbol{\theta} = [x(K), y(K), z(K), \dot{x}(K), \dot{y}(K), \dot{z}(K), \mathbf{b}_1', \ldots, \mathbf{b}_{N_S}']' \tag{15}$$

from

$$\mathbf{z} = \mathbf{h}(\boldsymbol{\theta}) + \mathbf{w} \tag{16}$$

where

$$\mathbf{h}(\boldsymbol{\theta}) = [h_{11}(\boldsymbol{\theta})', h_{21}(\boldsymbol{\theta})', \ldots, h_{N_S 1}(\boldsymbol{\theta})', \ldots, h_{1K}(\boldsymbol{\theta})', h_{2K}(\boldsymbol{\theta})', \ldots, h_{N_S K}(\boldsymbol{\theta})']' \tag{17}$$

$$\mathbf{w} = [\mathbf{w}_1(1)', \mathbf{w}_2(1)', \ldots, \mathbf{w}_{N_S}(1)', \ldots, \mathbf{w}_1(K)', \mathbf{w}_2(K)', \ldots, \mathbf{w}_{N_S}(K)']' \tag{18}$$

and the covariance of the stacked process noise (18) is the $(N_s K \times N_s K)$ block-diagonal matrix

$$R = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & R_{N_S} \end{bmatrix} \tag{19}$$

C. Space target dynamics

The state space model for a noiseless discrete-time system[1] is of the general form

$$\mathbf{x}(k+1) = f[\mathbf{x}(k), \mathbf{u}(k)] \quad k = 0,1,2,\ldots,K-1 \tag{20}$$

With small time steps ($\leq 10s$) we can approximate the motion model with the discrete-time dynamic equation

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + G\mathbf{u}(k) \tag{21}$$

where

$$\mathbf{x}(k) = [x(k), y(k), z(k), \dot{x}(k), \dot{y}(k), \dot{z}(k)]', \quad k = 0,1,2,\ldots,K \tag{22}$$

is the 6 dimensional state vector at time k, F is the state transition matrix, and $\mathbf{u}$ is a known input representing the gravitational effects acting on the target (given in (25)). The state transition matrix for a target with acceleration due to gravity is

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

[1] Since we are dealing with exoatmospheric motion it is reasonable to assume that it is noiseless.

and the known input gain matrix (multiplying the appropriate components of the gravity vector) is

$$G = \begin{bmatrix} \Delta t^2/2 & 0 & 0 \\ 0 & \Delta t^2/2 & 0 \\ 0 & 0 & \Delta t^2/2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \tag{24}$$

where Δt is the sampling interval. The gravity term is given by

$$\mathbf{u}(k) = g \frac{\mathbf{x}_{\mathrm{p}}(k)}{a(\mathbf{x}_{\mathrm{p}}(k))} \tag{25}$$

where $\mathbf{x}_{\mathrm{p}}$ is the position part of the state $\mathbf{x}$ in (22), $g = 9.8$ m/s^2, and

$$a = \sqrt{x(k)^2 + y(k)^2 + z(k)^2} \tag{26}$$

is the distance from the target to the origin of the coordinates system. For simplicity we assume g to be constant. The ratio $\mathbf{x}_{\mathrm{p}}/a$ yields the time-varying components of the gravity acting on the target and provides the scaling factor for the gravity term. Note that in view of (25), the state model (21) is not linear.

We shall obtain the maximum likelihood (ML) estimate of the augmented parameter vector (15) consisting of the (unknown) target position, velocity and sensor biases, by maximizing the likelihood function (LF) of $\boldsymbol{\theta}$ based on $\mathbf{z}$

$$\Lambda(\boldsymbol{\theta}; \mathbf{z}) = p(\mathbf{z} \mid \boldsymbol{\theta}) \tag{27}$$

where

$$p(\mathbf{z} \mid \boldsymbol{\theta}) = |2\pi R|^{-1/2} \exp(-\tfrac{1}{2}[\mathbf{z} - \mathbf{h}(\boldsymbol{\theta})]' R^{-1} [\mathbf{z} - \mathbf{h}(\boldsymbol{\theta})]) \tag{28}$$

and $\mathbf{h}$ is defined in (17)

The ML estimate (MLE) is then

$$\hat{\boldsymbol{\theta}}(\mathbf{z})^{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \Lambda(\boldsymbol{\theta}; \mathbf{z}) \tag{29}$$

In order to find the MLE, one has to solve a nonlinear least squares problem. This will be done using a numerical search via the Batch Iterated Least Squares (ILS) technique.

D. Bias Estimability

Intuitively, the observability of a system guarantees that the sensor measurements provide sufficient information for estimating the unknown parameters. As discussed in [3] the two requirement for bias estimability are:

First requirement for bias estimability. Each sensor provides a two-dimensional measurement (the two LOS angles to the target) at time K. We assume that each sensor sees the target at all the times $0, 1, 2, \ldots, K$. Stacking together all the measurements results in an overall measurement vector of dimension $2KN_S$. Given that the position, velocity of the target and bias vectors of each sensor are three-dimensional, and knowing that the number of equations (size of the stacked measurement vector) has to be at least equal to the number of parameters to be estimated (target state and biases), we must have

$$2KN_S \geq 3N_S + 6 \tag{30}$$

This is a necessary condition but not sufficient because (29) has to have a unique solution, i.e., the parameter vector has to be estimable. This is guaranteed by the second requirement.

Second requirement of bias estimability. This is the invertibility of the Fisher Information Matrix (FIM). In order to have parameter observability, the FIM must be invertible. If the FIM is not invertible (i.e., it is singular), then the CRLB (the inverse of the FIM) will not exist—the FIM will have one or more infinite eigenvalues, which means total uncertainty in a subspace of the parameter space, i.e., ambiguity [2].

For the example of bias estimability discussed in the sequel, to estimate the biases of 2 sensors (6 bias components) and 6 target components (3 position and 3 velocity components), i.e., the search is in an 12-dimensional space in order to meet the necessary requirement (30). As stated previously, the FIM must be invertible, so the rank of the FIM has to be equal to the number of parameters to be estimated ($6 + 6 = 12$, in the previous example). The full rank of the FIM is a necessary and sufficient condition for estimability. There exists, however, a subtle unobservability for this example that will necessitate the use of more measurements than the strict minimum number of measurements given by (30).

E. Iterated Least Squares for Maximization of the LF of $\boldsymbol{\theta}$

Given the estimate $\hat{\boldsymbol{\theta}}^j$ after j iterations, the batch ILS estimate after the $(j+1)$th iteration will be

$$\hat{\boldsymbol{\theta}}^{j+1} = \hat{\boldsymbol{\theta}}^j + [(H^j)' R^{-1} H^j]^{-1} (H^j)' R^{-1} [\mathbf{z} - \mathbf{h}(\hat{\boldsymbol{\theta}}^j)] \tag{31}$$

where

$$\mathbf{h}(\hat{\boldsymbol{\theta}}^j) = [h_{11}(\hat{\boldsymbol{\theta}}^j)', h_{21}(\hat{\boldsymbol{\theta}}^j)', \ldots, h_{N_S 1}(\hat{\boldsymbol{\theta}}^j)', \ldots, h_{1K}(\hat{\boldsymbol{\theta}}^j)', h_{2K}(\hat{\boldsymbol{\theta}}^j)', \ldots, h_{N_S K}(\hat{\boldsymbol{\theta}}^j)'] \tag{32}$$

where

$$H^j = \left. \frac{\partial \mathbf{h}(\boldsymbol{\theta}^j)}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}^j} \tag{33}$$

is the Jacobian matrix of the vector consisting of the stacked measurement functions (32) w.r.t. (15) evaluated at the ILS estimate from the previous iteration j. In this case, the Jacobian matrix is, with the iteration index omitted for conciseness,

$$H = [H_{11} \quad H_{21} \quad H_{N_S 1} \cdots H_{1K} \quad H_{2K} \quad H_{N_S K}]' \tag{34}$$

where

$$H_{sk} = \begin{bmatrix} \frac{h_{1_s}(k)}{\partial x(k)} & \frac{h_{1_s}(k)}{\partial y(k)} & \frac{h_{1_s}(k)}{\partial z(k)} & \frac{h_{1_s}(k)}{\partial \dot{x}(k)} & \frac{h_{1_s}(k)}{\partial \dot{y}(k)} & \frac{h_{1_s}(k)}{\partial \dot{z}(k)} & \frac{h_{1_s}(k)}{\partial b_{\alpha_1}} & \frac{h_{1_s}(k)}{\partial b_{\epsilon_1}} & \frac{h_{1_s}(k)}{\partial b_{\rho_1}} & \cdots & \frac{h_{1_s}(k)}{\partial b_{\alpha_{N_S}}} & \frac{h_{1_s}(k)}{\partial b_{\epsilon_{N_S}}} & \frac{h_{1_s}(k)}{\partial b_{\rho_{N_S}}} \\ \frac{h_{2_s}(k)}{\partial x(k)} & \frac{h_{2_s}(k)}{\partial y(k)} & \frac{h_{2_s}(k)}{\partial z(k)} & \frac{h_{2_s}(k)}{\partial \dot{x}(k)} & \frac{h_{2_s}(k)}{\partial \dot{y}(k)} & \frac{h_{2_s}(k)}{\partial \dot{z}(k)} & \frac{h_{2_s}(k)}{\partial b_{\epsilon_1}} & \frac{h_{2_s}(k)}{\partial b_{\epsilon_1}} & \frac{h_{2_s}(k)}{\partial b_{\rho_1}} & \cdots & \frac{h_{2_s}(k)}{\partial b_{\epsilon_{N_S}}} & \frac{h_{2_s}(k)}{\partial b_{\epsilon_{N_S}}} & \frac{h_{2_s}(k)}{\partial b_{\rho_{N_S}}} \end{bmatrix} \tag{35}$$

The appropriate partial derivatives with respect to the target positions and the bias terms can be found in [3], and the partial derivatives with respect to the target velocity components are:

$$\frac{\partial h_{1_s}(k)}{\partial \dot{x}_s(k)} = \Delta t \frac{\partial h_{1_s}(k)}{\partial x_s(k)} \tag{36}$$

$$\frac{\partial h_{1_s}(k)}{\partial \dot{y}_s(k)} = 0 \tag{37}$$

$$\frac{\partial h_{1_s}(k)}{\partial \dot{z}_s(k)} = \Delta t \frac{\partial h_{1_s}(k)}{\partial z_s(k)} \tag{38}$$

$$\frac{\partial h_{2_s}(k)}{\partial \dot{x}_s(k)} = \Delta t \frac{\partial h_{2_s}(k)}{\partial x_s(k)} \tag{39}$$

$$\frac{\partial h_{2_s}(k)}{\partial \dot{y}_s(k)} = \Delta t \frac{\partial h_{2_s}(k)}{\partial y_s(k)} \tag{40}$$

$$\frac{\partial h_{2_s}(k)}{\partial \dot{z}_s(k)} = \Delta t \frac{\partial h_{2_s}(k)}{\partial z_s(k)}. \tag{41}$$

F. Initialization

Assuming that the biases are null, the LOS measurements from the first and the second sensor $\alpha_1(k)$, $\alpha_2(k)$ and $\epsilon_1(k)$ can be used to solve for each initial Cartesian target position, in ECI coordinates, using (42)–(44). The two Cartesian positions formed from (42)–(44) can then be differenced to provide an approximate velocity. This procedure is analogous to two-point differencing [2] and will provide a full six-dimensional state to initialize the ILS algorithm.

$$x(k)^0 = \frac{\xi_2(k) - \xi_1(k) + \zeta_1(k)\tan\alpha_1(k) - \zeta_2(k)\tan\alpha_2(k)}{\tan\alpha_1(k) - \tan\alpha_2(k)} \tag{42}$$

$$y(k)^0 = \frac{\tan\alpha_1(k)(\xi_2(k) + \tan\alpha_2(k)(\zeta_1(k) - \zeta_2(k))) - \xi_1(k)\tan\alpha_2(k)}{\tan\alpha_1(k) - \tan\alpha_2(k)} \tag{43}$$

$$z(k)^0 = \eta_1(k) + \tan\epsilon_1(k)\left|\frac{(\xi_1(k) - \xi_2(k))\cos\alpha_2(k) + (\zeta_2(k) - \zeta_1(k))\sin\alpha_2(k)}{\sin(\alpha_1(k) - \alpha_2(k))}\right| \tag{44}$$

G. Cramér-Rao Lower Bound

In order to evaluate the efficiency of the estimator, the CRLB must be calculated. The CRLB provides a lower bound on the covariance matrix of an unbiased estimator as [2]

$$E\{(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})'\} \geq J(\boldsymbol{\theta})^{-1} \tag{45}$$

where J is the FIM, $\boldsymbol{\theta}$ is the true parameter vector to be estimated, and $\hat{\boldsymbol{\theta}}$ is the estimate. The FIM is

$$J(\boldsymbol{\theta}) = E\{[\nabla_{\boldsymbol{\theta}} \ln\Lambda(\boldsymbol{\theta})][\nabla_{\boldsymbol{\theta}} \ln\Lambda(\boldsymbol{\theta})]'\}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{true}}} \tag{46}$$

where the log-likelihood function is

$$\lambda(\boldsymbol{\theta}) \triangleq \ln\Lambda(\boldsymbol{\theta}) \tag{47}$$

$$J(\boldsymbol{\theta}) = H'R^{-1}H|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{true}}} \tag{48}$$

where H is the Jacobian matrix (34). Since $\boldsymbol{\theta}_{\text{true}}$ is not available in practice, J will be evaluated at the estimate, and, as it is shown later, the two results are practically the same.

H. Statistical Test for Efficiency with Monte Carlo Runs

Another measure of performance involves weighting the estimate error by the inverse of the covariance matrix P. The normalized estimation error squared (NEES) for the parameter θ under the hypothesis of efficiency, i.e.,

$$P = J^{-1} \tag{49}$$

is defined as

$$\epsilon_\theta = (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})'P^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) = (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})'J(\boldsymbol{\theta})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \tag{50}$$

and is chi-square distributed with n_θ (the dimension of θ) degrees of freedom, that is,

$$\epsilon_\theta \sim \chi^2_{n_\theta} \tag{51}$$

The hypothesis test for efficiency whether (51) can be accepted, as discussed in [2] and outlined next. The NEES is used in simulations to check whether the estimator is efficient, that is, the errors are statistically consistent with the covariance given by the CRLB—this is the efficiency check. Thus the efficiency check of the estimator (in simulation—because this is the only

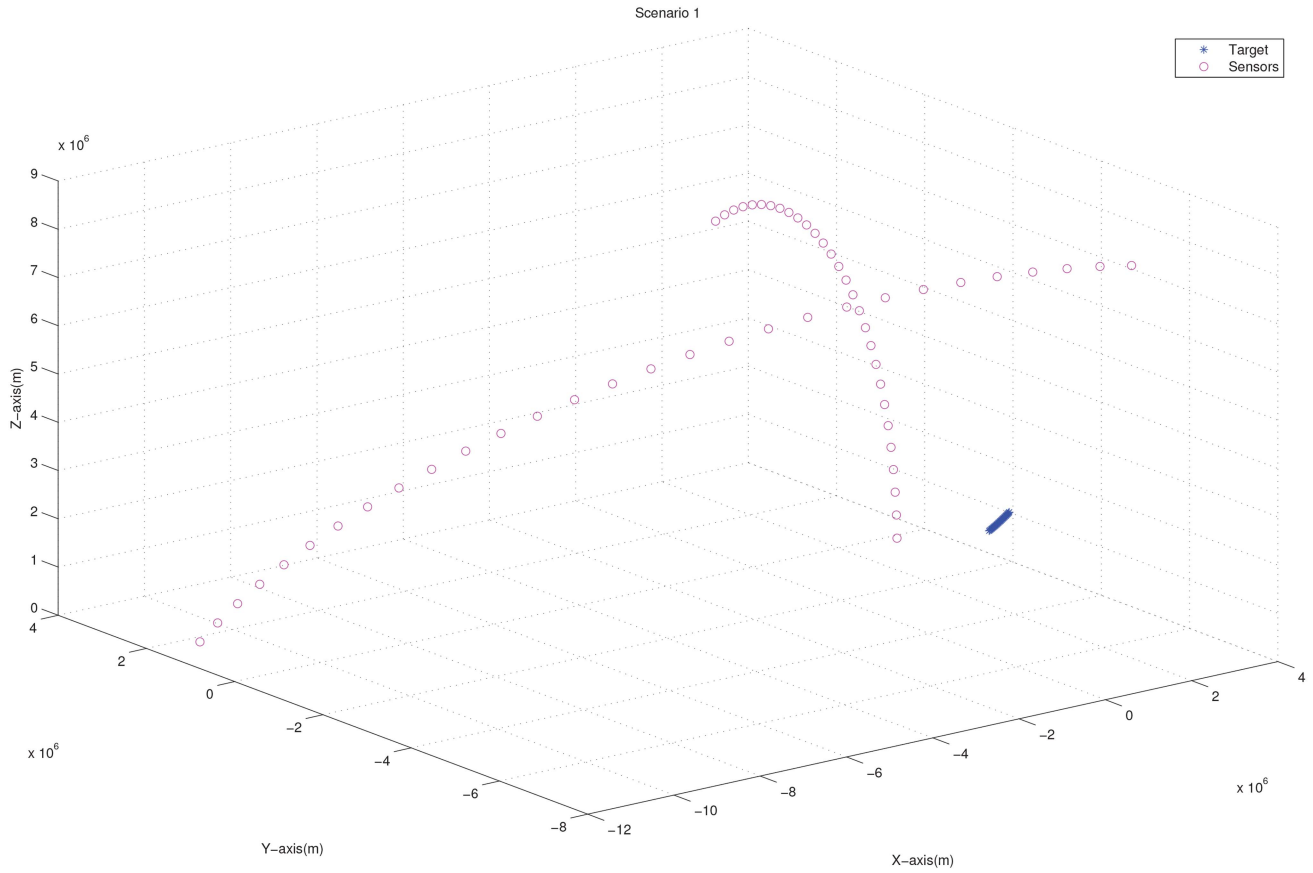


Fig. 2. Target and satellite trajectories for the two-sensor case

situation where θ is available) consists of verifying whether (51) holds. The practical procedure to check the estimator efficiency is using the sample average NEES from N independent Monte Carlo runs defined as

$$\bar{\epsilon}_\theta = \frac{1}{N}\sum_{i=1}^{N}\epsilon_\theta^i \tag{52}$$

The quantity $N\bar{\epsilon}_\theta$ is chi-square distributed with Nn_θ degrees of freedom.

Let Q be the type I error probability of the test. The $1-Q$ two-sided probability region for $N\bar{\epsilon}_\theta$ is the interval $[\epsilon_1', \epsilon_2']$.

$$\epsilon_1' = \chi^2_{Nn_\theta}\left(\frac{Q}{2}\right) \tag{53}$$

$$\epsilon_2' = \chi^2_{Nn_\theta}\left(1-\frac{Q}{2}\right) \tag{54}$$

where in view of the division by N in (52), one has

$$\epsilon_i = \frac{\epsilon_i'}{N} \tag{55}$$

Thus, if the estimator is efficient, one has to have

$$P\{\bar{\epsilon}_\theta \in [\epsilon_1', \epsilon_2']\} = 1-Q \tag{56}$$

III. SIMULATIONS

In this paper we used a hypothetical scenario to test our new methodology. The missile and satellite trajectories are generated using System Tool Kit (STK). The sensor satellites are in a circular orbits of 600 km and 700 km altitude with 0°, 60° degrees inclination, respectively. The target modeled represents a long range ballistic missile with a flight time of about 20 minutes. STK provides the target and sensor positions in three dimensional Cartesian coordinates at 1 s intervals. The measurement noise standard deviation σ_s (identical across sensors for both azimuth and elevation measurements, $\sigma_s^\alpha = \sigma_s^\epsilon = \sigma_s$) was assumed to be 30 μrad. The target launch time was chosen so that the satellite sensors were able to follow the missile trajectory throughout its flight path. As shown in Figure 3, these satellite orbits enabled maximum visibility of the missile trajectory from multiple angles. The missile and satellite trajectories displayed in Figure 3 represent 5 minutes of flight time (exoatmospheric). In order to establish a baseline for evaluating the performance of our method, we also ran the simulations without biases and with biases, but without bias estimation. As discussed in the previous section, the three sensor biases were roll, pitch and yaw angle offsets. Table II summarizes the bias values (in mrad).

In order to test for the statistical efficiency of the estimate (of the 12 dimensional vector (15)), the NEES is used, with the CRLB as the covariance matrix. The sample average NEES over 100 Monte Carlo runs calculated using the FIM evaluated at the true bias values, target

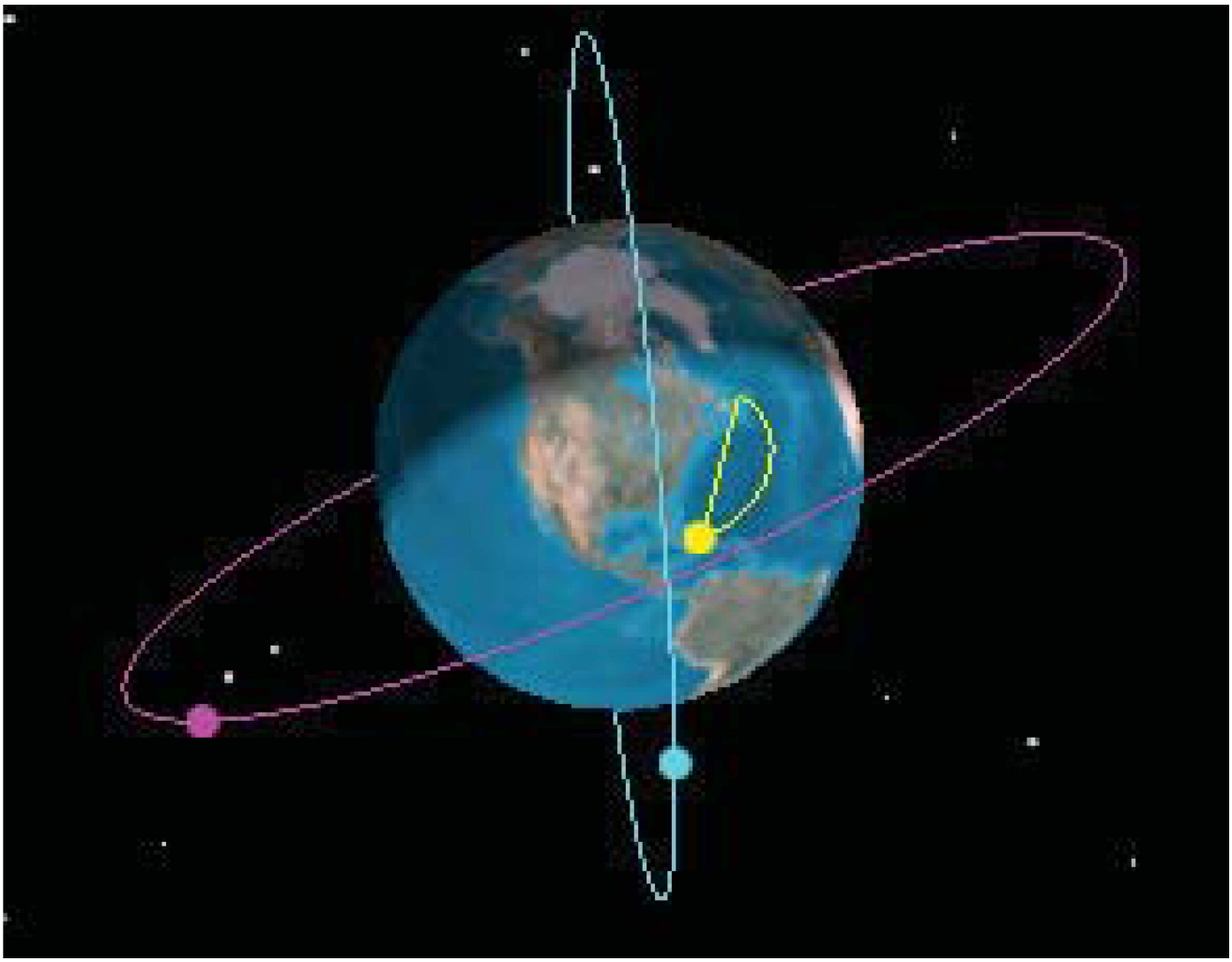


Fig. 3. Target and satellite trajectories for the two-sensor case

TABLE II
Sensor Biases (mrad).

	ψ	ρ	ϕ
Sensor 1	5.7596	4.3633	−3.8397
Sensor 2	4.8869	5.4105	−5.0615

TABLE III
Sample average bias NEES (CRLB evaluated at the estimate), for each of the 6 biases, over 100 Monte Carlo runs.

Biases	ψ_1	ρ_1	ϕ_1	ϕ_2	ψ_2	ρ_2
NEES	1.0326	0.9723	1.0239	1.0248	1.2009	0.8922

position, and velocity is approximately 11.52, and the sample average NEES calculated using the FIM evaluated at the estimated biases, target position and velocity is approximately 11.63 and both fall in the interval given below. According to the CRLB, the FIM has to be evaluated at the true parameter. Since this is not available in practice, however, it is useful to evaluate the FIM also at the estimated parameter, the only one available in real world implementations [12]. The results are practically identical regardless of which values are chosen for evaluation of the FIM. The 95% probability region for the 100 sample average NEES of the 12 dimensional parameter vector is [11.20, 12.81]. This NEES is found to be within this interval and the MLE is therefore statistically efficient. Table III shows the individual bias component NEES. The 95% probability region for the 100 sample average single component NEES is [0.74, 1.29]. These NEES are found to be within this interval.

The RMS errors for the target position and velocity are summarized in Table IV. In this table, the first estimation scheme was established as a baseline using bias-free LOS measurements to estimate the target position and velocity. For the second scheme, we used biased LOS measurements but we only estimated target position and velocity. In the last scheme, we used biased LOS measurements and we simultaneously estimated the target position, velocity, and sensor biases. Once again, bias estimation yields significantly improved target RMS position and velocity errors in the presence of biases.

Each component of $\boldsymbol{\theta}$ should also be individually consistent with its corresponding σ_{CRLB} (the square root

TABLE IV
Sample average RMSE for the target position (m) and velocity (ms^{-1}), over 100 Monte Carlo runs, for the 3 estimation schemes.

Scheme	Position RMSE	Velocity RMSE
1	107.44	5.16
2	47,161.10	25,149.32
3	494.49	19.55

TABLE V
Sample average bias (mrad) RMSE over 100 Monte Carlo runs and the corresponding bias standard deviation from the CRLB.

	RMSE	σ_{CRLB}
ψ_1	0.0326	0.0334
ρ_1	0.0239	0.0211
ϕ_1	0.0239	0.0261
ψ_2	0.0248	0.0252
ρ_2	0.0099	0.0096
ϕ_2	0.0122	0.0122

of the corresponding diagonal element of the inverse of FIM). In this case, the sample average bias RMSE over 100 Monte Carlo runs should be within 15% of its corresponding bias standard deviation from the CRLB (σ_{CRLB}) with 95% probability. The utmost limit ("existing information") for the scenario considered is around 10–33 μrad standard deviation for the bias errors, i,e., of the order of σ_s. Table V demonstrates the efficiency of the individual bias estimates.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we presented a new algorithm that uses a target of opportunity for estimation of measurement biases together with target state. The first step was formulating a general bias model for synchronized space-based optical sensors at known locations. The association of measurements is assumed to be perfect. Based on this, we used an ML approach that led to a batch nonlinear least-squares estimation problem for simultaneous estimation of the 3D Cartesian position and velocity components of the target of opportunity and the angle measurement biases of the sensors. The bias estimates, obtained via ILS, were shown to be unbiased and statistically efficient. For future work we plan to relax the no process noise assumption, reformulate the problem and again evaluate the statistical efficiency of the algorithm.

REFERENCES

[1] Sytem Tool Kit
registered trademark of Analytical Graphics Inc.
https://www.agi.com.

[2] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.
J. Wiley and Sons, 2001.

[3] D. Belfadel, Y. Bar-Shalom, and P. Willett
"Simultaneous target state and passive sensors bias estimation,"
in *Proc. FUSION Conf.*, Heidelberg, Germany, July 2016.

[4] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
"Bias Estimation and Observability for Optical Sensor Measurements with Targets of Opportunity,"
Journal of Advances in Information Fusion, vol. 9, no. 2, pp. 59–74, Dec. 2014.

[5] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
"Bias Estimation for Moving Optical Sensor Measurements with Targets of Opportunity,"
Journal of Advances in Information Fusion, vol. 10, no. 2, Dec. 2015.

[6] T. M. Clemons and K.-C. Chang
"Effect of Sensor Bias on Space-based Bearings-only tracker,"
in *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, #6968, San Diego, California, Aug. 2008.

[7] T. M. Clemons and K.-C. Chang
"Sensor Calibration using In-Situ Celestial Observations to Estimate Bias in Space-Based Missile Tracking,"
IEEE Trans. on Aerospace and Electronic Systems, vol. 48, no. 2, pp. 1403–1427, April 2012.

[8] D. F. Crouse
"Basic tracking using nonlinear 3D monostatic and bistatic measurements,"
in *IEEE Aerospace and Electronic Systems Magazine*, vol. 29, no. 8, Part II, pp. 4â\53, Aug. 2014.

[9] B. D. Kragel, S. Danford, S. M. Herman, and A. B. Poore
"Bias Estimation Using Targets of Opportunity,"
Proc. SPIE Conf. on Signal and Data Processing of Small Targets, #6699, Aug. 2007.

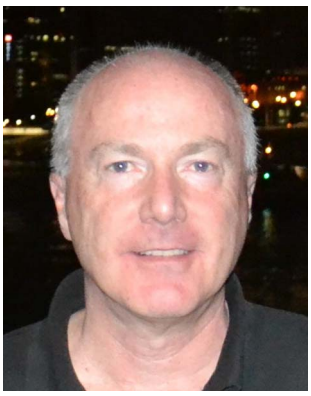
[10] B. D. Kragel, S. Danford, S. M. Herman, and A. B. Poore
"Joint MAP Bias Estimation and Data Association: Algorithms,"
Proc. SPIE Conf. on Signal and Data Processing of Small Targets, #6699-1E, Aug. 2007.

[11] B. D. Kragel, S. Danford, and A. B. Poore
"Concurrent MAP Data Association and Absolute Bias Estimation with an Arbitrary Number of Sensors,"
Proc. SPIE Conf. on Signal and Data Processing of Small Targets, #6969-50, May 2008.

[12] R. W. Osborne, III, and Y. Bar-Shalom
"Statistical Efficiency of Composite Position Measurements from Passive Sensors,"
IEEE Trans. on Aerospace and Electronic Systems, vol. 49, no. 4, pp. 2799–2806, Oct. 2013.

Djedjiga Belfadel is an Assistant Professor in the Electrical and Computer Engineering department at Fairfield University, Fairfield, CT. She obtained her B.S., degrees from the University of Mouloud Mammeri in 2003, her M.S., degrees from the University of New Haven in 2008, and her Ph.D. degree from University of Connecticut in 2015, all in electrical engineering. From 2009 to 2011, she worked, as an Electrical Engineer, at Evax Systems Inc. in Branford, Connecticut. Her research interests include target tracking, data association, sensor fusion, machine vision, and other aspects of estimation.

Yaakov Bar-Shalom received the B.S. and M.S. degrees from the Technion in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in EE. From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. His current research interests are in estimation theory, target tracking and data fusion. He has published over 550 papers and book chapters. He coauthored/edited 8 books, including *Tracking and Data Fusion* (YBS Publishing, 2011), He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He served as Associate Editor of the *IEEE Transactions on Automatic Control* and *Automatica*. He was General Chairman of the 1985 ACC. He served as Chairman of the Conference Activities Board of the IEEE CSS and member of its Board of Governors. He served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications during 2004–13. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE TAESystems in 1995 and 2000. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008 he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012 the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in Aerospace Engineering based on the citations of his work. He is the recipient of the 2015 ISIF Award for a Lifetime of Excellence in Information Fusion. This award has been renamed in 2016 as the Yaakov Bar-Shalom Award for a Lifetime of Excellence in Information Fusion.



Peter Willett received his B.A.Sc. (engineering science) from the University of Toronto in 1982, and his Ph.D. degree from Princeton University in 1986. He has been a faculty member at the University of Connecticut ever since, and since 1998 has been a Professor. His primary areas of research have been statistical signal processing, detection, machine learning, data fusion and tracking. He also has interests in and has published in the areas of change/abnormality detection, optical pattern recognition, communications and industrial/security condition monitoring. He was editor-in-chief of *IEEE Signal Processing Letters* (2014–2016). He was editor-in-chief for *IEEE Transactions on Aerospace and Electronic Systems* (2006–2011), and was Vice President for Publications for AESS (2012–2014). He is a member of the IEEE AESS Board of Governors 2003–2009, 2011–2016. He was General Co-Chair for the 2006 IEEE/ISIF Fusion Conference in Florence, Italy, and again for both 2008 in Cologne, Germany and 2011 in Chicago IL. He was Program Co-Chair for the 2016 ISIF/IEEE Fusion Conference in Heidelberg, Germany, the 2003 IEEE Conference on Systems, Man & Cybernetics in Washington DC, and Program Co-Chair for the 1999 Fusion Conference in Sunnyvale CA.

Distributed Fusion Algorithm for Passive Localization of Multiple Transient Emitters

WENBO DOU
YAAKOV BAR-SHALOM
LANCE KAPLAN
JEMIN GEORGE

This paper investigates the problem of deploying a network of passive sensors to estimate the positions of an unknown number of stationary transient emitters. Since a completely connected network, which has a link between every pair of nodes, is not feasible because of the power and bandwidth constraints, we developed a distributed algorithm that relies only on local communications between neighboring sensors. This distributed algorithm requires information diffusion within the network with the goal that every node achieves all target location estimates as accurate as a fusion center with centralized access to all information. The locations of the emitters are not completely observable by any single sensor since bearings and times of arrival with origin uncertainty are the only available measurements. These measurements are modeled as a realization of a Poisson point process at each sensor. The problem is formulated as a constrained optimization problem, which is solved via an alternating direction method of multipliers in a distributed manner based on the expectation maximization and averaging consensus algorithms. Consensus on the number of candidate targets as well as the inter-node estimate association are addressed so that the distributed algorithm converges to the maximum likelihood estimate. A likelihood function based approach using the estimated probability of detection is presented to determine the number of targets. Simulation results show that the distributed algorithm converges very fast and the root mean square error of target locations is almost as small as that obtained using the centralized algorithm. It is also shown that one can accurately determine the number of targets using the estimated probability of detection.

Manuscript received September 18, 2016; revised January 26, 2017 and March 1, 2017; released for publication March 7, 2017.

Refereeing of this contribution was handled by Chee-Yee Chong.

Authors' addresses: W. Dou and Y. Bar-Shalom, Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269 (E-mail: wenbo.dou@uconn.edu, ybs@engr.uconn.edu). L. Kaplan and J. George, U.S. Army Research Laboratory, 2800 Powder Mill Rd., Adelphi, MD 20783 (E-mail: {lance.m.kaplan.civ, jemin.george.civ}@mail.mil).

Supported by ARO Grant W991NF-10-1-0369.

1. INTRODUCTION

1.1. Background

This paper considers the problem of multiple transient emitter (target) localization using a wireless sensor network (WSN). One particular application is to utilize a network of acoustic gunfire detection systems mounted on a group of soldiers to localize adversaries in a battlefield [16][17]. It is assumed that the targets are stationary during the time window of interest but the number of the targets is unknown. The sensors can measure the line of sight (LOS) angles to the targets by detecting their emitted acoustic signals and record the times of arrival (TOAs) of the detected signals. This implies incomplete target location observability for any single sensor. Missed detections and false alarms are present due to the imperfection of the sensors. Furthermore, the associations between the measurements and the targets are unknown, that is, each sensor does not know from which target (or clutter) a particular measurement originates. Before estimating the position of any target, one has to associate the measurements from all the sensors. Therefore, the quality of data association is critical to the overall localization performance.

Two different fusion algorithms developed in our previous work [13] solved this problem using a centralized approach, i.e., we assumed that there is a fusion center collecting all the information from individual sensors either directly or by multi-hop relay, typically by wireless communication. Centralized access to all information can be difficult. For example, it requires a high transmission power to deliver the information from a single sensor directly to a fusion center in applications covering a large area. Moreover, the fusion center based approach is not robust, i.e., if the fusion center fails, the whole system fails. This has motivated a lot of work on distributed fusion or distributed optimization algorithms including the one presented in this paper.

One straightforward distributed solution is flooding, i.e., broadcasting the actual sensor measurements through the links in the network. In [7], a communication strategy of broadcasting new measurements was presented to allow distributed measurement fusion, which produces the optimal estimate at each node given all the measurements received up to any time for a linear dynamic system. For the localization problem considered in this paper, one has a nonlinear static system. The flooding approach still applies, by careful bookkeeping and a number of iterations of information exchange, each sensor would have all the information and can act as a fusion center to find the same global solution as a centralized approach. This method requires a large amount of data communication, storage memory, and bookkeeping overhead. For instance, it requires about S (the number of sensors) times the memory storage of the average consensus (AC) based approach.

TABLE I
Classification of the various versions of the shooter localization problem.

	Single target	Multiple targets
No missed detections or false alarms	P1	P5
Only missed detections exist	P2	P6
Only false alarms exist	P3	P7
Missed detections and false alarms exist	P4	P8

When it is used for the localization problem, the flooding approach is distributed in the sense that the information (all the measurements) is communicated in a distributed manner but it is centralized in the sense that the estimation algorithm including all computations is applied on all the information collected at every node, i.e., the flooding approach is a multiple replica of the centralized approach. In this paper, we present a consensus based algorithm that is different from the flooding approach and is distributed in the sense that both communication and estimation are performed in a distributed manner.

One of our approaches in [13] formulated the localization problem as an optimization problem and solved it using the expectation maximization (EM) algorithm. We observe that two types of subproblem are solved in the EM algorithm. One is to compute the average of variables with one variable from one sensor and the other is to solve a nonlinear least squares problem. Both subproblems can be formulated to optimize a global objective function, which can be written as a sum of local objective functions. Such problems can be solved using distributed optimization approaches whose goal is to recover the optimal global solution without any global coordination or interactions (like using a fusion center). Their solutions often contain a step where the sum or average of some quantity needs to be calculated and this can be achieved by an average consensus (AC) based approach.

The average consensus based approach with communication only between the one-hop neighbors scales well in that the communication overhead per sensor can be kept at an affordable level as the size of the network increases. Unlike the full flooding approach, which requires the local variables labeled with their origins, the average consensus approach does not need such labels and therefore uses less storage. If new nodes join the network, our consensus based distributed algorithm does not need to restart the whole process because the local variables can be updated following a (mini) flooding of only the new information.

In this paper, we assume that centralized access of all the information is not possible and we are interested in solving the problem of multiple transient emitter localization using an alternative algorithm that is different from the flooding approach and that is distributed in the sense that both communication and estimation are performed in a distributed manner. Since the goal is to have each sensor obtain a global estimate (which is a vector consisting of the number of targets and the position estimates of all targets) as good (or almost as good) as can be obtained by a fusion center using a centralized algorithm, information diffusion either in the form of raw measurements or in the form of some intermediate estimates (a function of raw measurements) within the network is necessary. Instead of using the raw measurement diffusion approach as in the flooding approach, we diffuse the intermediate estimates using the average consensus approach, i.e., the estimation is also performed in a distributed manner.

Without a fully connected network (each node can reach each other node via one or multiple "hops"), sending raw measurements to all nodes in order to achieve global optimal solution is a difficult task which requires "subnetwork" coordination, which is beyond the scope of this paper (multiple layers would be necessary). Therefore, we assume that the network is fully connected, i.e., there is a (not necessarily direct) path between every two sensors. If the network is not connected and has more than one connected subnetwork due to node or link failures, each subnetwork can be processed by our distributed algorithm independently. In such case, the consensus is achieved within each connected subnetwork.

Table I presents a classification of the various versions of the shooter localization problem. In view of the above discussion, it is necessary to develop a distributed algorithm to solve the problem P8 in Table I relying solely on local communications between one-hop neighboring sensors. Problems P3, P4, P6 and P7 are special cases of P8, therefore can be solved by the same distributed algorithm. Problem P1 is addressed in Section 2.7. Problems P2 are P5 are special extensions of P1 and will not be covered here.

1.2. Related Work

Distributed data fusion strategies, such as methods in [6], [8], [9], [10], [14], and [21] among others, are available for joint state estimation and data association in multi-sensor multi-target tracking scenarios. Since they are recursive algorithms that require sequential measurements and provide solutions to dynamic data association problems, they cannot be employed to solve joint parameter estimation and data association in a multi-sensor multi-target localization scenario (with incomplete observability at each sensor) considered in this paper. While most of the distributed estimation work in the literature assumes linear measurement models, our paper deals with nonlinear and incomplete target location measurements (direction of arrival and delayed arrival time). Although, one could imagine linearizing the localization problem and sharing messages between the nodes, we suspect that the linearization will probably

cause more errors than the distributed ADMM and will investigate this in our future work. Related work from robotics can be found in [19]. A recent comparison of optimal distributed estimation and consensus filtering for dynamic systems was done in [7].

A multi-dimensional assignment formulation assuming a Bernoulli measurement generation model that the number of measurements from each target received at each sensor is a Bernoulli random variable with parameter equal to the probability of detection as well as a cardinality selection formulation assuming a Poisson measurement generation model that the number of measurements from each target received at each sensor is a Poisson random variable with parameter equal to the probability of detection were considered in the centralized fusion algorithms [13] to solve the same problem of multiple transient emitter localization. This paper only considers developing a distributed algorithm to solve the cardinality selection problem assuming a Poisson measurement generation model[1] and leaves distributed multi-dimensional assignment algorithms for future work. While a list of measurements at each sensor was modeled as either realizations of a random variable with a mixture density or a Poisson point process (PPP) in [13], only PPP modeling is considered in this paper due to its simpler mathematical solution expression. Since the centralized algorithm solving the cardinality selection problem, which combines expectation maximization (EM) algorithm to estimating target parameters given a fixed number of targets and information criterion for selection of the best possible number of targets, is not amendable to a distributed implementation, it is necessary to develop a distributed EM algorithm.

Distributed EM algorithms have attracted a lot of attentions in sensor network applications for density estimation, data clustering and target tracking. For a fixed number of target, the localization problem can be considered as a density estimation problem. An incremental distributed EM algorithm presented in [23] is the first known scheme for density estimation and clustering in distributed sensor network. A distributed EM algorithm based on the averaging consensus filtering was developed in [18] for particle filter based target tracking. A distributed EM algorithm based on alternating direction method of multipliers (ADMM) was proposed in [15] for distributed data clustering. However, all these works assumes a *linear* generative model for their respective applications, which does not apply to a *nonlinear* generative model (see the measurement model in (78)) considered here due to the incomplete position measurement based on bearings and TOAs in the emitter localization scenario considered in this paper. Moreover, the parameters in these distributed EM algorithms are initialized to be either fixed values (zeros) or random values. This initialization approach was shown to be useless for our centralized EM algorithm, which requires an initialization based on the sequential m-best 2-D assignment algorithm applied on the lists of measurements from all sensors for the convergence to the global maximum.

1.3. Contributions

In this paper, we develop a distributed EM algorithm to solve the same problem as considered in [13] but in a distributed manner. The distributed processing introduces a number of challenges.

Firstly, the convergence of an EM algorithm (whether being centralized or distributed) depends highly on the initialization step. Previous studies on developing distributed EM algorithms assumed a linear measurement model and thus the initialization with fixed values (such as zeros) or random values, which is commonly used, works fine. This initialization does not work in the problem considered in this paper where the measurements (incomplete position observations) are nonlinear functions of target locations. Our earlier work shows that the assignment based initialization leads to global convergence. However, due to limited connections in a distributed setting, each sensor can only obtain a different EM initialization, which is a set of vectors, using the sequential m-best 2-D assignment algorithm on the measurement lists of its own and its neighbors (a subset of all the lists of measurements). For the global convergence of the EM algorithm, we developed a distributed set consensus algorithm ensuring that every node has the same initialization (the same number of targets and the same target locations).

Secondly, the maximization step in the standard EM algorithm has to be evaluated in a distributed manner. Although the probability of detection can be estimated by a distributed averaging consensus subroutine and the locations of the targets can be estimated by a distributed ADMM subroutine, this would result in a nested iterative algorithm with two subroutines being iterative algorithms themselves. Even more challenging, these two subroutines are needed for a number of iterations and at each iteration both of them requires local communications between sensors for a number of times, which would result in a very high communication cost. Instead, we manage to formulate a constrained optimization problem with equality conditions that force all local variables to be identical and developed a new distributed ADMM algorithm enabling a lower communication cost at the expense of additional local computation. The EM and AC based distributed ADMM algorithm is a generalization of previous distributed algorithms allowing

[1]The Bernoulli measurement generation model is more realistic than the Poisson measurement generation model. Therefore, the Bernoulli model is used to generate the synthetic data for the evaluation of the developed algorithm, whereas the Poisson model is assumed in the derivation of the developed algorithm. Using the Bernoulli model in the algorithm would make it excessively complicated because of the need to use multidimensional assignment.

the handling of the nonlinear and incomplete measurement models such as bearings in the passive sensing applications as here.

Last but not least, since we feel that a Bernoulli measurement generation model is a more realistic assumption and it reflects best the physical process of measurement generation, we used a likelihood function based thresholding approach to determine the number of targets.

1.4. Paper Organization

The remaining sections of this paper are organized as follows. Section 2 presents some preliminaries required for the development of the desired distributed algorithm. These include (i) graph modeling, (ii) a distributed AC algorithm for both single parameter estimation and multiple parameter estimation, (iii) data association test for two estimates as well as two sets of estimates, (iv) an algorithm of alternating direction method of multipliers and (v) a distributed nonlinear least squares algorithm, which can solve problem P1 in Table I. Section 3 formulates the problem by modeling each measurement set as a realization of a Poisson point process. Section 4 reviews a recently developed centralized algorithm that uses an EM algorithm to estimate the location and emission time parameters for a fixed number of targets. The distributed algorithm for problem P8 is presented in Section 5. The initialization issues of this algorithm—how to reach the consensus on the number of targets and how to reach the consensus on the target-estimate association—are discussed in Sections 5.1 and 5.2, respectively. An EM and AC based distributed ADMM algorithm is developed in Section 5.3. Section 5.4 describes a thresholding approach to distinguish real target estimates from false target estimates using the estimated probability of detection values. Section 6 presents and analyzes simulation results and Section 7 concludes the paper.

2. PRELIMINARIES

2.1. Graph Model

A wireless sensor network with S nodes (sensors) is deployed to collect data and perform data association and parameter estimation task. Every node is only able to communicate with its neighbors. Mathematically, this network can be modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the set of nodes

$$\mathcal{V} = \{v_1, v_2, \ldots, v_S\} \tag{1}$$

and the set of edges $\mathcal{E}$, where an edge $(v_i, v_j) \in \mathcal{E}$ is an unordered pair of distinct nodes, representing a two-way communication link between v_i and v_j. The graph $\mathcal{G}$ is assumed connected, meaning that there is a path between any two nodes. The set of neighbors of node v_i is defined as

$$\mathcal{N}_i = \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\} \tag{2}$$

The degree of node v_i is defined as

$$d_i = |\mathcal{N}_i| \tag{3}$$

where $|\cdot|$ denotes the set cardinality. The maximum degree of the graph $\mathcal{G}$ is defined as

$$d_{\max} = \max_i d_i \tag{4}$$

The Laplacian matrix L of the graph $\mathcal{G}$ is defined as

$$L_{ij} = \begin{cases} -1 & \text{if } v_j \in \mathcal{N}_i \\ d_i & \text{if } j = i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

2.2. Distributed Averaging Consensus Algorithm

Suppose a wireless sensor network with S nodes is deployed to estimate an unknown constant parameter $x \in \mathbf{R}^n$. Each node v_i makes a measurement

$$z_i = x + w_i \tag{6}$$

where w_i are independent, identically distributed, normal, zero mean, and with a known identity covariance matrix I. The maximum likelihood estimate of x is $(1/S)\sum_{i=1}^{S} z_i$, which is the mean vector of all measurements z_i. This estimate can be obtained by the following distributed averaging consensus algorithm.

Let us denote an initial value (z_i for the estimate problem) at node v_i by $u_i(0) \in \mathbf{R}^n$ at time $t = 0$. The matrix formed by the column vectors at all nodes is denoted as

$$\mathbf{U}(0) = [u_1(0) \quad u_2(0) \quad \ldots \quad u_S(0)]^T \in \mathbf{R}^{S\times n} \tag{7}$$

The goal of distributed averaging consensus is to make every node obtain the mean vector $(1/S)\sum_{i=1}^{S} u_i(0)$ eventually after gradually updating its value with a linear combination of its previously stored value and the values of its neighbors. One iteration of the process can be represented with a weight matrix W as

$$u_i(t+1) = W_{ii}u_i(t) + \sum_{j\in\mathcal{N}_i} W_{ij}u_j(t) \quad i = 1, \ldots, S \tag{8}$$

where $t = 0, 1, \ldots$ is the discrete time index, and W_{ij} is the weight on u_j at node v_i. Setting $W_{ij} = 0$ for $j \notin \mathcal{N}_i$, this iteration can be written in matrix form as

$$\mathbf{U}(t+1) = W\mathbf{U}(t) \tag{9}$$

and W is selected such that

$$\lim_{t\to\infty} \mathbf{U}(t) = \frac{1}{S}\mathbf{1}\mathbf{1}'\mathbf{U}(0) \tag{10}$$

The best constant edge weight matrix is given by [27]

$$W = I - \beta L \tag{11}$$

with

$$\beta = \frac{2}{\eta_1(L) + \eta_{S-1}(L)} \tag{12}$$

where $\eta_1(L)$ and $\eta_{S-1}(L)$ are the largest and second smallest eigenvalues of L, respectively.

In some cases, each node only has the knowledge of its neighbors rather than the connectivity of the entire network. It is more suitable to use the Metropolis weight matrix, which is defined as [28]

$$W_{ij} = \begin{cases} \dfrac{1}{1+\max\{d_i,d_j\}} & \text{if } v_j \in \mathcal{N}_i \\ 1-\sum_{v_k\in\mathcal{N}_i} W_{ik} & \text{if } j=i \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

2.3. Distributed Averaging Consensus Algorithm for Multiple Parameter Estimation with Unknown Data Association

Suppose a WSN with S nodes is used to estimate a set of N unknown constant parameters

$$X = \{x_1, x_2, \ldots, x_N\} \tag{14}$$

with each $x_j \in \mathbf{R}^n$. Each node v_i has a set of N measurements

$$Z_i = \{z_{i1}, z_{i2}, \ldots, z_{iN}\} \tag{15}$$

with one for each x_j. Let Π_N denote all permutations of the set $\{1,2,\ldots,N\}$, then the jth measurement of node v_i is

$$z_{ij} = x_{\pi_i(j)} + w_i \tag{16}$$

where $\pi_i \in \Pi_N$ is a permutation[2] at node v_i, and w_i are independent, identically distributed, normal, zero mean measurement noises with a known identity covariance matrix I.

Since the second index j of z_{ij} in the set Z_i contains no labeling information, one needs to perform data association and weighted averaging update (8) simultaneously for multiple parameter estimation. Let us denote the stacked vector at node v_i at time t as

$$\mathbf{u}_i(t) = [u_{i1}^T(t), u_{i2}^T(t), \ldots, u_{iN}^T(t)]^T \tag{17}$$

and $u_{ij}(0)$ is initialized as z_{ij}. At time t, node v_i calculates an optimal permutation[3] π_{ji} for each of its neighbor nodes v_j as

$$\pi_{ji} = \arg\min_{\pi\in\Pi_N} \sum_{k=1}^{N} \|u_{ik}(t) - u_{j\pi(k)}(t)\|^2 \tag{18}$$

Then node v_i updates each segment of its stacked vector (17) as

$$u_{ik}(t+1) = W_{ii}u_{ik}(t) + \sum_{j\in\mathcal{N}_i} W_{ij}u_{j\pi_{ji}(k)}(t) \tag{19}$$

where the index $\pi_{ji}(k)$ refers to the segment of the stacked vector at node v_j that associates with the kth segment of the stacked vector at node v_i according to the permutation π_{ji} (18), and the weight matrix is given by (13).

[2]It is a one-to-one mapping function from an ordered set $\{1,2,\ldots,N\}$ to a particular permutation of this set.

[3]The second index i of π_{ji} indicates that the optimal permutation is obtained with respect to $\mathbf{u}_i(t)$.

2.4. Association Test for Two Estimates

Suppose that sensor v_i has an unbiased estimate $\hat{x}_i$ of the n-dimensional (unknown) parameter x_i with a covariance matrix P_i and sensor v_j has an unbiased estimate $\hat{x}_j$ of the n-dimensional (unknown) parameter x_j with a covariance matrix P_j. We are interested in testing whether $x_i = x_j$. Let us denote the difference of the two estimates as

$$\hat{\Delta}_{ij} = \hat{x}_i - \hat{x}_j \tag{20}$$

which is the estimate of the difference of the parameters

$$\Delta_{ij} = x_i - x_j \tag{21}$$

Since the estimation errors

$$\tilde{x}_i = x_i - \hat{x}_i \tag{22}$$

$$\tilde{x}_j = x_j - \hat{x}_j \tag{23}$$

are zero-mean, the estimation error of the difference of the parameters

$$\tilde{\Delta}_{ij} = \Delta_{ij} - \hat{\Delta}_{ij} = \tilde{x}_i - \tilde{x}_j \tag{24}$$

is also zero-mean and it has the covariance matrix

$$\begin{aligned} T_{ij} &= E\{\tilde{\Delta}_{ij}\tilde{\Delta}_{ij}^T\} = E\{(\tilde{x}_i - \tilde{x}_j)(\tilde{x}_i - \tilde{x}_j)^T\} \\ &= P_i + P_j - E\{\tilde{x}_i\tilde{x}_j^T\} - E\{\tilde{x}_j\tilde{x}_i^T\} \end{aligned} \tag{25}$$

If $\tilde{x}_i$ and $\tilde{x}_j$ are independent, then we have

$$T_{ij} = P_i + P_j \tag{26}$$

Assuming that $\tilde{x}_i$ and $\tilde{x}_j$ are Gaussian, the normalized estimation error squared (NEES) [2] for Δ

$$\epsilon_{ij} \triangleq \tilde{\Delta}_{ij}^T T_{ij}^{-1} \tilde{\Delta}_{ij} \tag{27}$$

is chi-square distributed with n degrees of freedom.

The null hypothesis that the two parameters are the same and the alternative hypothesis are

$$H_0 : \Delta = 0 \tag{28}$$

$$H_1 : \Delta \neq 0 \tag{29}$$

Under H_0 ($\Delta = 0$), we have the following

$$\tilde{\Delta}_{ij} = -\hat{\Delta}_{ij} \tag{30}$$

$$\epsilon_{ij} = \hat{\Delta}_{ij}^T T_{ij}^{-1} \hat{\Delta}_{ij} \tag{31}$$

Therefore, the test of H_0 vs. H_1 is as follows. If

$$\hat{\Delta}_{ij}^T T_{ij}^{-1} \hat{\Delta}_{ij} \leq F_{\chi_n^2}^{-1}(1-\alpha) \tag{32}$$

where $F_{\chi_n^2}^{-1}(\cdot)$ is the inverse of the cumulative distribution function (cdf) of a chi-square random variable with n degrees of freedom, we will not reject H_0 at a significance level of α. Then it is likely that $\hat{x}_i$ and $\hat{x}_j$ are estimates of the same parameter.

2.5. Association Test for Two Sets of Estimates

Suppose that there are N unknown n-dimensional constant parameters

$$X = \{x_1, x_2, \ldots, x_N\} \tag{33}$$

Sensor v_i has a set of N_i estimates with corresponding covariance matrices

$$\hat{x}_i = \{\hat{x}_{i1}, \hat{x}_{i2}, \ldots, \hat{x}_{iN_i}\} \tag{34}$$

$$\mathcal{P}_i = \{P_{i1}, P_{i2}, \ldots, P_{iN_i}\} \tag{35}$$

Similarly, sensor v_j has N_j estimates with corresponding covariance matrices

$$\hat{x}_j = \{\hat{x}_{j1}, \hat{x}_{j2}, \ldots, \hat{x}_{jN_j}\} \tag{36}$$

$$\mathcal{P}_j = \{P_{j1}, P_{j2}, \ldots, P_{jN_j}\} \tag{37}$$

We assume that each sensor has at most one estimate for a particular parameter and the estimation errors are independent.

If $\hat{x}_{ik}$ and $\hat{x}_{j\ell}$ are estimates of the same parameter, then the NEES

$$d_{k\ell} = (\hat{x}_{ik} - \hat{x}_{j\ell})^T (P_{ik} + P_{j\ell})^{-1} (\hat{x}_{ik} - \hat{x}_{j\ell}) \tag{38}$$

can be regarded as a distance measure between $\hat{x}_{ik}$ and $\hat{x}_{j\ell}$. A small value of $d_{k\ell}$ indicates a high probability of both being the estimates of the same parameter.

To deal with incomplete associations caused by missed detections, we add dummy estimates $\hat{x}_{i0}$ and $\hat{x}_{j0}$ to the sets $\hat{x}_i$ and $\hat{x}_j$, respectively [24]. The distance involving a dummy estimate is defined as

$$d_{k0} = d_{0\ell} = F^{-1}_{\chi^2_n}(1 - \alpha) \tag{39}$$

for a small value (say, 0.01) of α.

To associate the estimates in set $\hat{x}_i$ with those in set $\hat{x}_j$, we solve a generalized 2-D assignment problem

$$\min_{\rho_{k\ell}} \sum_{k=0}^{N_i} \sum_{\ell=0}^{N_j} \rho_{k\ell} d_{k\ell} \tag{40}$$

subject to

$$\sum_{\ell=0}^{N_j} \rho_{k\ell} = 1 \quad \forall\ k = 1, 2, \ldots, N_i \tag{41}$$

$$\sum_{k=0}^{N_i} \rho_{k\ell} = 1 \quad \forall\ \ell = 1, 2, \ldots, N_j \tag{42}$$

$$\rho_{k\ell} \in \{0, 1\} \quad k = 0, 1, \ldots, N_i;\ \ell = 0, 1, \ldots, N_j \tag{43}$$

The modified auction algorithm [24] can be applied to the above problem.

The association results of $\hat{x}_{ik}$ are determined as follows.

If

$$\rho_{k0} = 1 \tag{44}$$

then $\hat{x}_{ik}$ is assigned to the dummy estimate $\hat{x}_{j0}$, that is, the probability that no estimate in $\hat{x}_j$ comes from the same parameter as $\hat{x}_{ik}$ is 0.99 for $\alpha = 0.01$. In this case, $\hat{x}_{ik}$ is not associated.

If

$$\rho_{k\ell} = 1 \tag{45}$$

then $\hat{x}_{ik}$ is associated with $\hat{x}_{j\ell}$.

The association results of $\hat{x}_{j\ell}$ are determined in a similar way.

2.6. The Alternating Direction Method of Multipliers (ADMM) Algorithm

Consider the following equality-constrained optimization problem

$$\min_{z,y} \{f(z) + g(y)\} \tag{46}$$

subject to

$$Az + By = c \tag{47}$$

with variables $z \in \mathbf{R}^p$ and $y \in \mathbf{R}^q$, where $A \in \mathbf{R}^{m \times p}$, $B \in \mathbf{R}^{m \times q}$ and $c \in \mathbf{R}^m$ are given.

The augmented Lagrangian of (46) is defined as

$$L_\rho(z, y, \lambda) = f(z) + g(y) + \lambda^T (Az + By - c) + \frac{\rho}{2} \|Az + By - c\|_2^2 \tag{48}$$

where λ is the dual variable or Lagrange multiplier and $\rho > 0$ is the penalty parameter.

The ADMM algorithm [5] solves (46) by iterating the following 3 steps

$$z^{k+1} \triangleq \arg\min_z L_\rho(z, y^k, \lambda^k) \quad z\text{-minimization} \tag{49}$$

$$y^{k+1} \triangleq \arg\min_y L_\rho(z^{k+1}, y, \lambda^k) \quad y\text{-minimization} \tag{50}$$

$$\lambda^{k+1} \triangleq \lambda^k + \rho(Az^{k+1} + By^{k+1} - c) \quad \text{dual update} \tag{51}$$

where ρ is used as the step size for the dual update and the superscript is the iteration counter.

In the ADMM, the variables z and y are updated in an alternating or sequential fashion instead of being minimized jointly, which accounts for the term *alternating direction*. Separating the minimization over z and y into two steps is precisely what allows for decomposition when f (or g) is separable with respect to a partition of the variable z (or y) into subvectors.

2.7. Distributed Nonlinear Least Squares Algorithm

This subsection presents a distributed solution to the problem P1 in Table I. We are interested in localizing a single target using the network $\mathcal{G}$ without missed detections or false alarms. Suppose each node v_i has a scalar measurement a_i from the target, we need to solve the unconstrained optimization problem

$$\min_x \sum_{i=1}^{S} (h(x) - a_i)^2 \tag{52}$$

TABLE II
Averaging consensus based distributed ADMM algorithm.

1: Node v_i initializes x_i^1 and $\lambda_i^1 = 0$
2: Compute $\bar{x}^1 = \frac{1}{S}\sum_{i=1}^{S} x_i^1$ via a distributed averaging consensus algorithm
3: **for** $k = 1,2,\ldots$ **do** until convergence
4: **for all** v_i **do**
5: Compute x_i^{k+1} via (71)
6: Compute $\bar{x}^{k+1} = \frac{1}{S}\sum_{i=1}^{S} x_i^{k+1}$ via a distributed averaging consensus algorithm
7: Compute λ_i^{k+1} via (72)
8: **end for**
9: **end for**

where $x \in \mathbf{R}^2$ is the parameter to be estimated (or the variable for the minimization), $h(\cdot)$ is a nonlinear function of x (for instance, $h(x)$ is an arctan function in a bearing-only localization problem) and S is the number of sensors.

Consider the constrained optimization problem, which is equivalent to (52)

$$\min_{x_1,x_2,\ldots,x_S} \sum_{i=1}^{S} (h(x_i) - a_i)^2 \tag{53}$$

subject to

$$x_1 = x_2 = \ldots = x_S = w \tag{54}$$

We can put (54) in the form of (47) by setting

$$z = [x_1^T \quad x_2^T \ldots x_S^T]^T \tag{55}$$

$$y = w \tag{56}$$

$$f(z) = \sum_{i=1}^{S} (h(x_i) - a_i)^2 \tag{57}$$

$$g(y) = 0 \tag{58}$$

$$A = I_{2S} \tag{59}$$

$$B = [-I_2 \quad -I_2 \cdots -I_2]^T \in \mathbf{R}^{2S\times 2} \tag{60}$$

$$c = 0 \tag{61}$$

Therefore, the augmented Lagrangian is

$$L_\rho(x_1,x_2,\ldots,x_S,w,\lambda) = \sum_{i=1}^{S} \left[(h(x_i) - a_i)^2 + \lambda_i^T (x_i - w) + \frac{\rho}{2}\|x_i - w\|_2^2 \right] \tag{62}$$

where

$$\lambda = [\lambda_1^T \ \lambda_2^T \ldots \lambda_S^T]^T \tag{63}$$

The z-minimization step (49) is

$$(x_1^{k+1},x_2^{k+1},\ldots,x_S^{k+1}) = \arg\min_{x_1,x_2,\ldots,x_S} L_\rho(x_1,x_2,\ldots,x_S,w^k,\lambda^k) \tag{64}$$

which can be carried out in a distributed fashion as

$$x_i^{k+1} = \arg\min_{x_i} (h(x_i) - a_i)^2 + \lambda_i^{kT}(x_i - w^k) + \frac{\rho}{2}\|x_i - w^k\|_2^2 \quad i = 1,2,\ldots,S \tag{65}$$

The y-minimization step (50) is

$$\begin{aligned} w^{k+1} &= \arg\min_{w} L_\rho(x_1^{k+1},x_2^{k+1},\ldots,x_S^{k+1},w,\lambda^k) \\ &= \arg\min_{w} \sum_{i=1}^{S} \left[\lambda_i^{kT}(x_i^{k+1} - w) + \frac{\rho}{2}\|x_i^{k+1} - w\|_2^2 \right] \\ &= \frac{1}{S}\sum_{i=1}^{S} x_i^{k+1} + \frac{1}{S\rho}\sum_{i=1}^{S} \lambda_i^k \end{aligned} \tag{66}$$

The dual update step (51) is

$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - w^{k+1}) \quad i = 1,2,\ldots,S \tag{67}$$

If we carry out the summation of (67) over i and substitute w^{k+1} from (66), then

$$\sum_{i=1}^{S} \lambda_i^{k+1} = \sum_{i=1}^{S} \lambda_i^k + \rho\sum_{i=1}^{S} x_i^{k+1} - S\rho w^{k+1} = 0 \quad k \neq 0 \tag{68}$$

which means that the dual variables have average value zero after the first iteration. If the dual variables are initialized such that

$$\sum_{i=1}^{S} \lambda_i^1 = 0 \tag{69}$$

then, the y-minimization step simplifies to

$$w^{k+1} = \frac{1}{S}\sum_{i=1}^{S} x_i^{k+1} \triangleq \bar{x}^{k+1} \tag{70}$$

The simplified ADMM steps, in a distributed form, become

$$x_i^{k+1} \triangleq \arg\min_{x_i} [h(x_i) - a_i]^2 + \lambda_i^{kT}(x_i - \bar{x}^k) + \frac{\rho}{2}\|x_i - \bar{x}^k\|_2^2 \quad i = 1,2,\ldots,S \tag{71}$$

$$\lambda_i^{k+1} \triangleq \lambda_i^k + \rho(x_i^{k+1} - \bar{x}^{k+1}) \quad i = 1,2,\ldots,S \tag{72}$$

Based on the above ADMM steps, we obtain an averaging consensus based distributed algorithm as shown in Table II. Each node v_i stores and updates two vectors x_i and λ_i. At iteration $k = 1$, each node initializes a local parameter estimate x_i^1 and obtains $\bar{x}^1$ via a distributed averaging consensus algorithm as discussed in Section 2.2. The dual variables $\lambda_i^1 = 0$ are also initialized. During the kth iteration, each node updates its local parameter estimate x_i^{k+1} using (71). Next, each node reaches the consensus on $\bar{x}^{k+1}$, and subsequently, updates its local dual variable λ_i^{k+1} using (72), which concludes the kth iteration.

Reformulations of (52) other than (53) include [26] and [22], which result in a bridge-sensor based distributed ADMM and a coloring-scheme based distributed ADMM, respectively. However, either prior assignment of bridge sensors [26] or colors [22] is required for the respective algorithm to function properly. However, in these versions it is difficult to make a new assignment in case of node or link failures. Whereas, the averaging consensus based distributed ADMM algorithm does not require any feature assignment to individual nodes since it relies solely on information diffusion across the network.

3. PROBLEM STATEMENT AND FORMULATION

3.1. Problem Statement

Consider a scenario where there are N stationary targets located in $\mathbb{R}^2$. The target locations are denoted as

$$\mathbf{T} = (\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_N) = \left(\begin{bmatrix} T_{x_1} \\ T_{y_1} \end{bmatrix}, \begin{bmatrix} T_{x_2} \\ T_{y_2} \end{bmatrix}, \ldots, \begin{bmatrix} T_{x_N} \\ T_{y_N} \end{bmatrix} \right) \quad (73)$$

The number of targets and their locations are unknown quantities of interest, to be estimated. A wireless sensor network consisting of S stationary nodes is deployed at known locations

$$\mathbf{S} = (\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_S) = \left(\begin{bmatrix} S_{x_1} \\ S_{y_1} \end{bmatrix}, \begin{bmatrix} S_{x_2} \\ S_{y_2} \end{bmatrix}, \ldots, \begin{bmatrix} S_{x_S} \\ S_{y_S} \end{bmatrix} \right) \quad (74)$$

to perform this estimation task. There is one transient event occurring at each target location. Each node is able to observe these transient events by detecting the acoustic signals arising from them and measure the bearings to the targets and the TOAs of the received acoustic signals. The acoustic signal emission times are denoted as

$$\mathbf{t}^{\mathrm{e}} = (t_1^{\mathrm{e}}, t_2^{\mathrm{e}}, \ldots, t_N^{\mathrm{e}}) \quad (75)$$

For notational simplicity, let us denote

$$\mathbf{\Phi} = [\phi_1^T \quad \phi_2^T \ldots \phi_N^T]^T \quad (76)$$

where

$$\phi_i = [T_{x_i} \quad T_{y_i} \quad t_i^{\mathrm{e}}]^T \quad (77)$$

denotes the unknown 3-dimensional parameter of ith target.

If the transient events are separated significantly in time, the measurements from the same event will be close in time and the measurements from different events will also be separated significantly in time, and then the target locations can be estimated one at a time using the algorithm presented in 2.7. Therefore, we assume a more challenging situation that the transient events are close in time. In this case, the data association between the measurements and the targets has to be addressed before the network can fuse the measurements from a common origin to estimate the corresponding target location.

It is assumed that all measurements fall within a short time window W. Let m_ℓ denote the number of measurements (one measurement is defined as a vector consisting of both a bearing and a TOA due to one acoustic signal in this context) obtained by the ℓth sensor within the time window W. The jth measurement received by the ℓth sensor, if it is from the ith target at t_i^{e}, is

$$\mathbf{z}_{\ell j} = \mathbf{h}_\ell(\phi_i) + \mathbf{w}_{\ell j} \quad i = 1, \ldots, N; \quad \ell = 1, \ldots, S; \quad j = 1, \ldots, m_\ell \quad (78)$$

where $\mathbf{w}_{\ell j}$ is a zero mean white Gaussian measurement noise with a known diagonal covariance matrix R_ℓ and

$$\mathbf{h}_\ell(\phi_i) = \begin{bmatrix} \theta_{\ell i} \\ t_{\ell i} \end{bmatrix} = \begin{bmatrix} \arctan\left[\dfrac{T_{y_i} - S_{y_\ell}}{T_{x_i} - S_{x_\ell}} \right] \\ t_i^{\mathrm{e}} + \dfrac{\sqrt{(T_{x_i} - S_{x_\ell})^2 + (T_{y_i} - S_{y_\ell})^2}}{c} \end{bmatrix} \quad (79)$$

where t_i^{e} is the unknown emission time of the acoustic signal from ith target and c is the known speed of sound.

To incorporate false alarms, we denote a clutter target (with index 0) as ϕ_0. A false measurement detected by the ℓth sensor consists of a bearing θ_0, which is uniformly distributed in the field of view of the ℓth sensor, and its arrival time t_0, which is uniformly distributed in the interval $[0, W]$. The number of false alarms from each sensor is assumed to be a Poisson random variable with mean

$$N_{\mathrm{fa}} = \lambda_{\mathrm{fa}} \Phi W \quad (80)$$

where Φ is the range of the field of view and is assumed to be the same for each sensor and λ_{fa} can be interpreted as the spatial-temporal density.

The likelihood function [2] of the target parameter[4] (location and emission time) based on the measurement $\mathbf{z}_{\ell j}$ is

$$\Lambda(\phi_0; \mathbf{z}_{\ell j}) \triangleq p(\mathbf{z}_{\ell j} \mid \phi_0) = p(\theta_0) p(t_0) = \frac{1}{\Phi W} \quad (81)$$

$$\Lambda(\phi_i; \mathbf{z}_{\ell j}) \triangleq p(\mathbf{z}_{\ell j} \mid \phi_i) = |2\pi R_\ell|^{-1/2} \cdot \exp\{-\tfrac{1}{2}[\mathbf{z}_{\ell j} - \mathbf{h}_\ell(\phi_i)]' R_\ell^{-1} [\mathbf{z}_{\ell j} - \mathbf{h}_\ell(\phi_i)]\} \quad i \neq 0 \quad (82)$$

where (81) is the probability density function (pdf) of a clutter-origin measurement (a false alarm), and (82) is the pdf of a real measurement from a true target with unknown ϕ_i.

The problem is to estimate N and $\mathbf{\Phi} = \{\phi_i, \ i = 1, \ldots, N\}$ (therefore knowing $\mathbf{T} = \{\mathrm{T}_i, \ i = 1, \ldots, N\}$) given the complete set of observations $\mathbf{Z} = \{\mathbf{z}_{\ell j}, \ \ell = 1, \ldots, S; \ j = 1, \ldots, m_\ell\}$ in the presence of missed detections and false alarms and without the knowledge of the true data association.

[4] If the source is clutter, it has no emission time, only an arrival time.

3.2. Poisson Point Process Measurement Modeling

Assume the number of targets, N, is given. The number of measurements m_ℓ and $\{\mathbf{z}_{\ell j}, j = 1,2,\ldots,m_\ell\}$ obtained at the ℓth sensor is jointly modeled as a realization of a Poisson Point Process (PPP) [11]. The measurement set at the ℓth sensor is denoted as

$$\psi_\ell = \{m_\ell, \mathbf{z}_{\ell 1}, \mathbf{z}_{\ell 2}, \ldots, \mathbf{z}_{\ell m_\ell}\} \tag{83}$$

In this case, the points $\mathbf{z}_{\ell j}$ occur in the space $\mathbb{S} = \{(\theta,t) : \theta \in \Phi, t \in [0,W]\}$ and their order is irrelevant. The PPP is fully parameterized by its spatial intensity function

$$\mu_\ell(\mathbf{z}) = \sum_{i=0}^{N} p_i^{\mathrm{d}} g_{\ell i}(\mathbf{z}) \tag{84}$$

where p_i^{d} is the probability of detection for the real target i ($i \neq 0$) and is assumed to be the same at each sensor and with abuse of notation

$$p_0^{\mathrm{d}} = N_{\mathrm{fa}} \tag{85}$$

is the expected number of false alarms at each sensor; the density $g_{\ell i}(\mathbf{z})$ is the conditional[5] pdf of a measurement $\mathbf{z}$ obtained by the ℓth sensor given that it is associated with the ith target and is given by

$$g_{\ell i}(\mathbf{z}) = \frac{1}{\Phi W} \quad i = 0 \tag{86}$$

$$g_{\ell i}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{h}_\ell(\phi_i), R_\ell) \quad i = 1,\ldots,N \tag{87}$$

For notational simplicity, we denote

$$\mathbf{p}^{\mathrm{d}} = [p_0^{\mathrm{d}} \quad p_1^{\mathrm{d}} \ldots p_N^{\mathrm{d}}]^T \tag{88}$$

which is assumed to be unknown and therefore the set of parameters to be estimated is expanded to $\boldsymbol{\theta} = [\boldsymbol{\Phi}^T \ \ \mathbf{p}^{\mathrm{d}T}]^T$ for a given N.

The number of points in the PPP is a Poisson random variable with mean $\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z}$, that is, the probability mass function (pmf) of m_ℓ is

$$\begin{aligned} p(m_\ell) &= \frac{(\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z})^{m_\ell}}{m_\ell!} \exp\left\{ - \int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z} \right\} \\ &= \frac{\left(\sum_{i=0}^{N} p_i^{\mathrm{d}}\right)^{m_\ell}}{m_\ell!} \exp\left(- \sum_{i=0}^{N} p_i^{\mathrm{d}} \right) \end{aligned} \tag{89}$$

The m_ℓ points are defined as independent and identically distributed (i.i.d.) samples of a random variable with probability density function

$$p(\mathbf{z}) = \frac{\mu_\ell(\mathbf{z})}{\int_{\mathbb{S}} \mu_\ell(\mathbf{z}) d\mathbf{z}} = \frac{\sum_{i=0}^{N} p_i^{\mathrm{d}} g_{\ell i}(\mathbf{z})}{\sum_{i=0}^{N} p_i^{\mathrm{d}}} \tag{90}$$

The joint pmf-pdf of ψ_ℓ from (83) is

$$p(\psi_\ell \mid \boldsymbol{\theta}) = \exp\left(- \sum_{i=0}^{N} p_i^{\mathrm{d}} \right) \prod_{j=1}^{m_\ell} \mu_\ell(\mathbf{z}_{\ell j} \mid \boldsymbol{\theta}) \tag{91}$$

where the conditioning (dependency) on $\boldsymbol{\theta}$ will be explicitly indicated hereafter. The factorial term $m_\ell!$ in (89) is canceled out because there are $m_\ell!$ permutations of an ordered list of measurements. Let $\boldsymbol{\Psi}$ denote the set of all measurement sets (from the S sensors), i.e.,

$$\boldsymbol{\Psi} = \{\psi_1, \psi_2, \ldots, \psi_S\} \tag{92}$$

The conditional independence of the S measurement sets yields

$$p(\boldsymbol{\Psi} \mid \boldsymbol{\theta}) = \prod_{\ell=1}^{S} p(\psi_\ell \mid \boldsymbol{\theta}) \tag{93}$$

Therefore, we can find the maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ by maximizing (93).

3.3. Data Association Modeling

Since the intensity function (84) is a mixture of uniform or Gaussian pdf and the association is unknown, we model the latent association variables as conditionally independent random variables

$$\kappa_{\ell j} \in \{0,1,2,\ldots,N\} \tag{94}$$

that identify which component generated $\mathbf{z}_{\ell j}$. Here $\kappa_{\ell j} = 0$ indicates that the measurement is generated by the clutter. The set of latent variables for the ℓth sensor is denoted as

$$\kappa_\ell = \{\kappa_{\ell 1}, \ldots, \kappa_{\ell m_\ell}\} \tag{95}$$

such that the complete set of latent variables for all sensors is

$$\boldsymbol{\kappa} = \{\kappa_1, \ldots, \kappa_S\} \tag{96}$$

The latent association variables may be regarded as "marks" associated with each of the points in the PPP. If we define a mark space

$$\mathbb{M} \triangleq \{0,1,2,\ldots,N\} \tag{97}$$

then the marked measurement set at the ℓth sensor denoted by

$$\psi_\ell^{\mathrm{M}} = \{m_\ell, (\mathbf{z}_{\ell 1}, \kappa_{\ell 1}), \ldots, (\mathbf{z}_{\ell m_\ell}, \kappa_{\ell m_\ell})\} \tag{98}$$

represents a realization of the marked[6] PPP for the ℓth sensor on the product space $\mathbb{S} \times \mathbb{M}$. It can be shown that the intensity function of ψ_ℓ^{M} is

$$\mu_\ell^{\mathrm{M}}(\mathbf{z}, \kappa \mid \boldsymbol{\theta}) = p_\kappa^{\mathrm{d}} g_{\ell\kappa}(\mathbf{z}) \tag{99}$$

The joint probability density function of ψ_ℓ^{M} is, similarly to (91), given by

$$\begin{aligned} p(\psi_\ell^{\mathrm{M}} \mid \boldsymbol{\theta}) &= \exp\left(- \sum_{\kappa=0}^{N} \int_{\mathbb{S}} \mu_\ell^{\mathrm{M}}(\mathbf{z}, \kappa \mid \boldsymbol{\theta}) d\mathbf{z} \right) \\ &\quad \cdot \prod_{j=1}^{m_\ell} \mu_\ell^{\mathrm{M}}(\mathbf{z}_{\ell j}, \kappa_{\ell j} \mid \boldsymbol{\theta}) \end{aligned} \tag{100}$$

[5] $g_{\ell i}(\mathbf{z} \mid \phi_i)$ will be used when the conditioning needs to be explicitly indicated.

[6] The superscript of ψ_ℓ^{M} indicates that the associations are known, i.e., "marked."

TABLE III
Centralized EM algorithm.

1: Initializes $\boldsymbol{\theta}^{(0)}$
2: **for** $n = 1,2,\ldots$ **do** until convergence
3: **E step** Evaluate

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(n-1)}) = \sum_{\boldsymbol{\kappa}} p(\boldsymbol{\kappa} \mid \mathbf{Z}, \boldsymbol{\theta}^{(n-1)}) \ln p(\boldsymbol{\Psi}^{\mathrm{M}} \mid \boldsymbol{\theta}) \tag{106}$$

4: **M step** Evaluate $\boldsymbol{\theta}^{(n)}$ as

$$\boldsymbol{\theta}^{(n)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(n-1)}) \tag{107}$$

5: **end for**

Let us denote the marked measurement sets from all sensors as

$$\boldsymbol{\Psi}^{\mathrm{M}} = \{\psi_1^{\mathrm{M}}, \psi_2^{\mathrm{M}}, \ldots, \psi_S^{\mathrm{M}}\} \tag{101}$$

The conditional independence of these S marked measurement sets yields the pmf-pdf for $\boldsymbol{\Psi}^{\mathrm{M}}$ as

$$p(\boldsymbol{\Psi}^{\mathrm{M}} \mid \boldsymbol{\theta}) = \exp\left(-S\sum_{i=0}^{N} p_i^{\mathrm{d}}\right) \prod_{\ell=1}^{S}\prod_{j=1}^{m_\ell} p_{\kappa_{\ell j}}^{\mathrm{d}} g_{\ell\kappa_{\ell j}}(\mathbf{z}_{\ell j} \mid \boldsymbol{\theta}) \tag{102}$$

where we have used the fact

$$\sum_{\kappa=0}^{N}\left(\int_{\mathbb{S}} p_\kappa^{\mathrm{d}} g_{\ell\kappa}(\mathbf{z} \mid \mathbf{T}, \mathbf{t}^{\mathrm{e}}) d\mathbf{z}\right) = \sum_{i=0}^{N} p_i^{\mathrm{d}} \tag{103}$$

Dividing (102) by (93) leads to the density of the marks conditioned on the observed measurements and the unknown parameters

$$p(\boldsymbol{\kappa} \mid \mathbf{Z}, \boldsymbol{\theta}) = \prod_{\ell=1}^{S}\prod_{j=1}^{m_\ell} p_\ell(\kappa_{\ell j} \mid \mathbf{z}_{\ell j}, \boldsymbol{\theta}) \tag{104}$$

where

$$p_\ell(\kappa_{\ell j} \mid \mathbf{z}_{\ell j}, \boldsymbol{\theta}) = \frac{p_{\kappa_{\ell j}}^{\mathrm{d}} g_{\ell\kappa_{\ell j}}(\mathbf{z}_{\ell j} \mid \boldsymbol{\theta})}{\mu_\ell(\mathbf{z}_{\ell j} \mid \boldsymbol{\theta})} \tag{105}$$

4. CENTRALIZED ALGORITHM

4.1. Centralized EM Algorithm

Given the joint distribution $p(\boldsymbol{\Psi}^{\mathrm{M}} \mid \boldsymbol{\theta})$ over observed $\boldsymbol{\Psi}$ and latent variables $\boldsymbol{\kappa}$, governed by the parameter $\boldsymbol{\theta}$, the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ from the likelihood function $p(\boldsymbol{\Psi} \mid \boldsymbol{\theta})$ can be found by the standard (named as centralized hereafter) EM algorithm [12] as shown in Table III.

Evaluation of the E step decomposes into two terms

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(n-1)}) = Q_p + Q_\phi \tag{108}$$

where

$$Q_p = Q(\mathbf{p}^{\mathrm{d}} \mid \boldsymbol{\theta}^{(n-1)}) = -S\sum_{i=0}^{N} p_i^{\mathrm{d}} + \sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell}\sum_{i=0}^{N} \ln(p_i^{\mathrm{d}}) \alpha_{\ell ji}^{(n-1)} \tag{109}$$

$$Q_\phi = Q(\boldsymbol{\Phi} \mid \boldsymbol{\theta}^{(n-1)}) = \sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell}\sum_{i=0}^{N} \ln(g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_i)) \alpha_{\ell ji}^{(n-1)} \tag{110}$$

where

$$\alpha_{\ell ji}^{(n-1)} = p_\ell(\kappa_{\ell j} = i \mid \mathbf{z}_{\ell j}, \boldsymbol{\theta}^{(n-1)}) = \frac{p_i^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_i^{(n-1)})}{\sum_{i=0}^{N} p_i^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_i^{(n-1)})} \tag{111}$$

The M step involves two separate maximizations with respect to $\mathbf{p}^{\mathrm{d}}$ and $\boldsymbol{\Phi}$. From the Karush-Kuhn-Tucker (KKT) conditions [20], we have

$$p_i^{\mathrm{d}(n)} = \begin{cases} \dfrac{1}{S}\sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell} \alpha_{\ell j0}^{(n-1)} & \text{if } i = 0 \\ \min\left\{1, \dfrac{1}{S}\sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell} \alpha_{\ell ji}^{(n-1)}\right\} & \text{if } i \neq 0 \end{cases} \tag{112}$$

Since Q_ϕ in (110) can be further decomposed into $N+1$ terms, the parameters of each target can be estimated independently as

$$\boldsymbol{\phi}_i^{(n)} = \arg\max_{\boldsymbol{\phi}_i} \sum_{\ell=1}^{S}\sum_{j=1}^{m_\ell} \ln(g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_i)) \alpha_{\ell ji}^{(n-1)} \tag{113}$$

5. DISTRIBUTED ALGORITHM

Note that it is possible to have a distributed implementation of the centralized EM algorithm if (i) every node has the consensus on the initialization and (ii) every node has the consensus on the parameter estimates at the end of each M step. The second condition can readily be satisfied if an averaging consensus based distributed ADMM is applied to solve (113), which is a nonlinear least squares problem, and a distributed averaging consensus algorithm is applied to obtain (112). However, it is not trivial to have the same initialization for $\boldsymbol{\theta} = [\boldsymbol{\Phi}^T \ \mathbf{p}^{\mathrm{d}T}]^T$ among all sensors, especially for the component $\boldsymbol{\Phi}$. Simulation results show that it is good enough to initialize each p_i^{d} to be 1. Whereas, equal initialization at some pre-fixed values (for instance, zero vectors) for $\boldsymbol{\Phi}$ could result in the convergence of the EM algorithm to estimates that are very different from the desired MLE.

There are two possible initialization approaches in a single target localization scenario. Assume that the data association is known and no missed detections or false alarms occur, we want to localize a single target using the algorithm in Table II. The first approach is to initialize the target location at each node using only its bearing measurement. The average distance from the wireless sensor network to the target is assumed to be D. Given a range R (probably unknown in a real scenario), each node initializes the target location along the measured line of sight in the direction towards the target randomly with a distance (between the initialized target location and the node itself) being uniformly distributed

TABLE IV
RMSE using different initializations for distributed localization of a single target.

	Centralized	Intersection	Random $R = 30$	Random $R = 60$
RMSE (m)	1.7036	1.7733	3.2988	5.2915

in $[D\text{-}0.5R, D\text{+}0.5R]$. The second approach is to obtain the LOS information (bearing and sensor location) from one of its neighbors and use the intersection of two lines of sight as the initial target location estimate.

Table IV lists the root mean square error (RMSE) of the target location (averaged over 100 Monte-Carlo runs) using different initialization approaches to localize a single target at the location (8.7 m, 99.6 m) in a scenario given in Section 6. It shows that the performance of the distributed algorithm with LOS intersection initialization is almost as good as the centralized algorithm, which assumes all bearing measurements available at a fusion center and also uses intersection initialization. With a random initialization based on some knowledge, which is likely unavailable, the distributed algorithm converges to local minimum point of (53) with $h(x)$ being an arctan function.

For a multiple target localization scenario with unknown data association, the random initialization approach will be worse. Therefore, in a similar way as what we did at the fusion center in a centralized fusion algorithm, each node obtains an initial position estimate for each target that is very close to the final MLE, by associating its local measurements with those from its neighbors using the sequential m-best 2-D assignment algorithm [1]. Another important reason that we choose the sequential m-best 2-D algorithm over the random initialization approach is that the position estimates obtained are completely observable with corresponding covariance matrices, which allows the use of the association method described in Section 2.5 to reach the consensus on the initialization.

If the probability of detection is low or the false alarm rate is high, then it is possible that the initial estimated numbers of targets at various nodes are different. Some nodes could have estimated more targets than there actually are due to false alarms, whereas other nodes could have estimated less due to missed detections. Section 5.1 discusses how to reach the consensus on the number of candidate targets[7] among all nodes.

Section 5.2 assumes that each node has an initial set consisting of the same number of target estimates which correspond to the same group of candidate targets, and discusses how to reach the consensus on on target-estimate association, that is, for a given ordered set of candidate targets, each node should know the association between its estimates and the targets. Note that the initial estimated value of parameter $\mathbf{\Phi}$ could still be different from node to node. However, the consensus on target-estimate association requires all the nodes have exactly the same set of target estimates.

In Section 5.3, we develop a distributed algorithm that assumes all the nodes have the same set of target estimates. The consensus on target-estimate association is required for convergence of the algorithm.

5.1. Consensus on the Number of Targets

If the probability of detection is low or the false alarm rate is high, then it is possible that the initial estimated numbers of targets at various nodes are different. Some nodes could have estimated more targets than there should be, whereas other nodes could have estimated less. In this subsection, we extend the problem solved in Section 2.3 to the case when missed detections and/or false alarms exist and develop a distributed set averaging consensus algorithm to expand some or all sets so that we end up with sets of estimates for the same number of candidate (real or false) targets. Each sensor gradually modifies its own set by performing the association test presented in Section 2.5 with the sets of its neighbors.

Let us denote the initial set of estimates with corresponding covariance matrices at node υ_i as

$$\hat{\mathbf{\Phi}}_i = \{\hat{\phi}_{i1}, \hat{\phi}_{i2}, \ldots, \hat{\phi}_{iN_i}\} \tag{114}$$

$$\mathcal{Q}_i = \{Q_{i1}, Q_{i2}, \ldots, Q_{iN_i}\} \tag{115}$$

where each $\hat{\phi}_{ik}$ corresponds to one candidate target and the number of candidate targets N_i is probably distinct for different nodes υ_i. Assume that there are N_c candidate targets with parameters

$$\mathbf{\Phi}_c = \{\phi_1, \phi_2, \ldots, \phi_{N_c}\} \tag{116}$$

of which only N parameters correspond to real targets and the remaining $N_c - N$ parameters correspond to false targets. The number N_c will only be known at the end of the algorithm. For any target parameter in $\mathbf{\Phi}_c$, there is at most one estimate $\hat{\phi}_{ik}$ at node υ_i.

Figure 1 illustrates the concept of candidate target. In this example, both sensors have detected true targets at coordinates (10 m, 10 m) and (20 m, 20 m), therefore each of these is a candidate target. Sensors 1 and 2 each also have an additional estimated target around (31 m, 31 m) and (29 m, 11 m), respectively. In this case, we assume that these two estimates fail to be associated. Therefore, two additional targets, which are assumed to be at coordinates (30 m, 30 m) and (30 m, 10 m), are also candidate targets.

Referring back to the same context in Section 2.5, the sets $\mathbf{\Phi}_c$ and $\hat{\mathbf{\Phi}}_i$ defined in (116) and (114) play the same roles as X and $\hat{x}_i$ in (33) and (34), respectively. The independent estimation error assumption is valid only when two estimates have no common source of error [3]. In the case that two neighboring nodes each have

[7]The concept of candidate target is discussed in detail in Section 5.1.

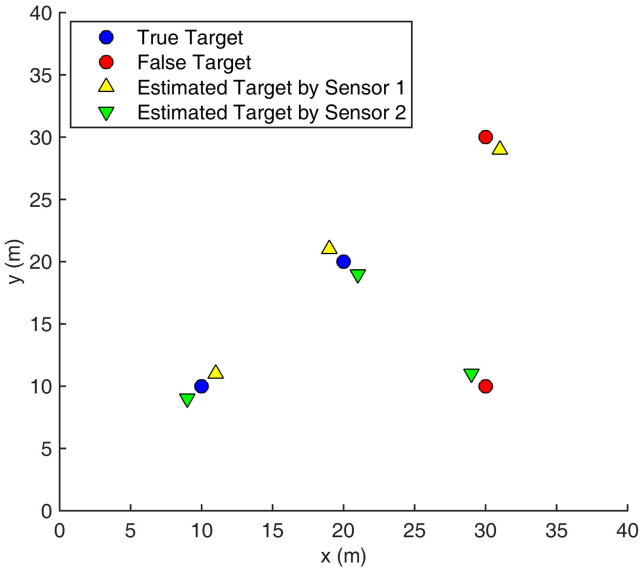


Fig. 1. An illustrative example: each sensor has three estimates, there are four candidate targets.

one estimate for the same target, it is quite likely that these two estimates are obtained using some common measurements and therefore they have correlated errors. Since it is difficult to calculate the cross covariance, we will use (26) as an approximation. This approximation only applies to true target location estimates that are supposed to be associated, and will not affect the decisions involving estimates that belong to false targets. In our approach, the independence assumption (of the errors in the estimated target locations at different sensors) is used only to build the consensus. We do not "fuse" the corresponding covariances, which pertain to errors that are dependent; fusing them under independence assumption would indeed be optimistic and unreasonable.

One iteration of the distributed set averaging consensus algorithm is described next. At iteration t, node v_i expands $\hat{\mathbf{\Phi}}_i(t)$ sequentially with each neighboring node $v_j \in \mathcal{N}_i$.

Firstly, for a given significance level α, the following generalized 2-D assignment problem

$$\min_{\rho_{k\ell}} \sum_{k=0}^{N_i} \sum_{\ell=0}^{N_j} \rho_{k\ell} d_{k\ell}(t) \tag{117}$$

subject to

$$\sum_{\ell=0}^{N_j} \rho_{k\ell} = 1 \text{ for all } k = 1,2,\ldots,N_i \tag{118}$$

$$\sum_{k=0}^{N_i} \rho_{k\ell} = 1 \text{ for all } \ell = 1,2,\ldots,N_j \tag{119}$$

$$\rho_{k\ell} \in \{0,1\} \text{ for all } k = 0,1,\ldots,N_i \text{ and } \ell = 0,1,\ldots,N_j \tag{120}$$

where, similarly as in (38) and (39), with the addition of a dummy estimate $\hat{\phi}_{i0}$ at each node v_i, the distance between two estimates are defined as

$$d_{k\ell}(t) = \begin{cases} (\hat{\phi}_{ik}(t) - \hat{\phi}_{j\ell}(t))^T (P_{ik}(t) + P_{j\ell}(t))^{-1} \\ \cdot (\hat{\phi}_{ik}(t) - \hat{\phi}_{j\ell}(t)) \quad \text{if } k > 0 \text{ and } \ell > 0 \\ F^{-1}_{\chi^2_n}(1-\alpha) \quad \text{if } k = 0 \text{ or } \ell = 0 \end{cases} \tag{121}$$

Next, $\hat{\mathbf{\Phi}}_i(t)$ could be expanded based on the solution $\rho_{k\ell}$ to the assignment problem. If

$$\rho_{0\ell} = 1 \tag{122}$$

which means that the estimate $\hat{\phi}_{j\ell}(t)$ is not associated with any estimate at node v_i, then $\hat{\mathbf{\Phi}}_i(t)$ is expanded to $\hat{\mathbf{\Phi}}_i(t) \cup \{\hat{\phi}_{j\ell}(t)\}$. If there are multiple estimates that are not associated, then they are all used to expand $\hat{\mathbf{\Phi}}_i(t)$.

The algorithm terminates when every node set has the same number of estimates and no set can be expanded further.

Note that if a sensor does not have a position estimate for target i, it will "copy" a position estimate for target i from one of its neighbors. If a sensor has a position estimate for a false target, then all its neighbors need to "copy" this estimate so that every sensor has a position estimate for the same false target. Since the total number of target estimates across all the sensors is a finite number $\sum_{i=1}^{S} N_i$, where S is the number of sensors, and each iteration expands at least one set, the algorithm will be terminated in a finite number of iterations.

5.2. Consensus on the Target-Estimate Association

Suppose that the initial sets of target estimates, either obtained directly via the assignment algorithm across all nodes or by means of the method described in Section 5.1, have the same number of target estimates.

The local variable of $\boldsymbol{\theta}_\ell$ has components $\mathbf{\Phi}_\ell$ and $\mathbf{p}^{\mathrm{d}}_\ell$. We initialize $\mathbf{p}^{\mathrm{d}}_\ell$ as a vector of ones. The component $\mathbf{\Phi}_\ell = [\phi^T_{\ell 1} \;\; \phi^T_{\ell 2} \ldots \phi^T_{\ell N}]^T$ will be initialized using the set obtained via the sequential assignment algorithm denoted by

$$\Phi_\ell = \{\varphi_{\ell 1}, \varphi_{\ell 2}, \ldots, \varphi_{\ell N}\} \tag{123}$$

There are $N!$ ways of initialization for node ℓ. We want to find a permutation for each set Φ_ℓ

$$\pi_\ell(\Phi_\ell) = \{\varphi_{\ell\pi_\ell(1)}, \varphi_{\ell\pi_\ell(2)}, \ldots, \varphi_{\ell\pi_\ell(N)}\} \tag{124}$$

such that for any $k = 1,2\ldots,N$, the set of estimates $\{\varphi_{\ell\pi_\ell(k)}, \; \ell = 1,2,\ldots,S\}$, one from each sensor, corresponds to the same target. For this purpose, we can apply the algorithm in Section 2.3 on the sets Φ_ℓ, $\ell = 1,\ldots,S$, and when the algorithm terminates, we have all the sets equal.

We initialize $\mathbf{\Phi}_\ell$ as

$$\mathbf{\Phi}_\ell = [\varphi^T_{\ell\pi_\ell(1)} \quad \varphi^T_{\ell\pi_\ell(2)} \ldots \varphi^T_{\ell\pi_\ell(N)}]^T \tag{125}$$

where, letting $i = \pi_\ell(k)$

$$\varphi_{\ell i} = [T_{x_i} \quad T_{y_i} \quad t_i^{\mathrm{e}}]^T \tag{126}$$

is ordered such that

$$T_{x_i} \le T_{x_j},\ \forall\ i \le j \tag{127}$$

$$T_{y_i} \le T_{y_j},\ \forall\ i \le j \text{ and } T_{x_i} = T_{x_j} \tag{128}$$

The use of the ordering rules (127) and (128) to label targets makes sense only when the sets of the estimates from all nodes are the same.

5.3. The EM and AC Based Distributed ADMM Algorithm

The centralized EM algorithm provides a method to solve the following optimization problem

$$\min_{\boldsymbol{\theta}} \{-\ln p(\boldsymbol{\Psi} \mid \boldsymbol{\theta})\} \tag{129}$$

where

$$\ln p(\boldsymbol{\Psi} \mid \boldsymbol{\theta}) = \sum_{\ell=1}^{S} \ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}) \tag{130}$$

To develop a distributed algorithm to solve the above problem, we consider an equivalent formulation with equality constraints between local variables $\boldsymbol{\theta}_\ell$ and a global variable $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_S} \sum_{\ell=1}^{S} -\ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}_\ell) \tag{131}$$

subject to

$$\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \ldots = \boldsymbol{\theta}_S = \boldsymbol{\theta} \tag{132}$$

The augmented Lagrangian is

$$L_\rho(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_S, \boldsymbol{\theta}, \lambda) = \sum_{\ell=1}^{S} \Big[-\ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}_\ell) + \lambda_\ell^T (\boldsymbol{\theta}_\ell - \boldsymbol{\theta}) + \frac{\rho}{2} \|\boldsymbol{\theta}_\ell - \boldsymbol{\theta}\|_2^2 \Big] \tag{133}$$

Following the similar derivations as presented in Section 2.7, we can obtain the ADMM steps, which are in a distributed form, as

$$\boldsymbol{\theta}_\ell^{k+1} = \arg\min_{z} \sum_{\ell=1}^{S} \Big[-\ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}_\ell) + \lambda_\ell^{kT} (\boldsymbol{\theta}_\ell - \boldsymbol{\theta}^k) + \frac{\rho}{2} \|\boldsymbol{\theta}_\ell - \boldsymbol{\theta}^k\|_2^2 \Big] \tag{134}$$

$$\boldsymbol{\theta}^{k+1} = \frac{1}{S} \sum_{\ell=1}^{S} \boldsymbol{\theta}_\ell^{k+1} \tag{135}$$

$$\lambda_\ell^{k+1} = \lambda_\ell^k + \rho(\boldsymbol{\theta}_\ell^{k+1} - \boldsymbol{\theta}^{k+1}) \tag{136}$$

TABLE V
EM and averaging consensus based distributed ADMM algorithm.

1: Node v_ℓ initializes $\boldsymbol{\theta}_\ell^1$ by a sequential m-best 2-D assignment algorithm and $\lambda_\ell^1 = 0$
2: Compute $\boldsymbol{\theta}^1 = \frac{1}{S} \sum_{\ell=1}^{S} \boldsymbol{\theta}_\ell^1$ by a distributed averaging consensus algorithm
3: **for** $k = 1, 2, \ldots$ **do** until convergence
4: **for all** v_ℓ **do**
5: Compute $\boldsymbol{\theta}_\ell^{k+1}$ via (134) by a local EM algorithm
6: Compute $\boldsymbol{\theta}^{k+1} = \frac{1}{S} \sum_{\ell=1}^{S} \boldsymbol{\theta}_\ell^{k+1}$ by a distributed averaging consensus algorithm
7: Compute λ_ℓ^{k+1} via (136)
8: **end for**
9: **end for**

Based on the above ADMM steps, we obtain an EM and averaging consensus based distributed algorithm as summarized in Table V. Each node v_ℓ stores and updates two vectors $\boldsymbol{\theta}_\ell$ and λ_ℓ. At iteration k=1, each node initializes a local parameter estimate $\boldsymbol{\theta}_\ell^1$ and reaches the consensus on the global variable $\boldsymbol{\theta}^1$ via a distributed averaging consensus algorithm. The local dual variable is initialized as $\lambda_\ell^1 = 0$. During the kth iteration, each node updates the local variable $\boldsymbol{\theta}_\ell^{k+1}$ via (134), which is solved by the local EM algorithm as in Table VI because of the term $\ln p(\boldsymbol{\psi}_\ell \mid \boldsymbol{\theta}_\ell)$. Next, each node obtains $\boldsymbol{\theta}^k$ via a distributed averaging consensus algorithm, and subsequently, updates its local dual variable λ_i^{k+1} using (136), which concludes the kth iteration.

In the local EM algorithm, the dual variable λ_ℓ is partitioned as

$$\lambda_\ell = \begin{bmatrix} \lambda_{\phi\ell} \\ \lambda_{p\ell} \end{bmatrix} \tag{143}$$

with respect to the components $\boldsymbol{\Phi}_\ell$ and $\mathbf{p}_\ell^{\mathrm{d}}$ of $\boldsymbol{\theta}_\ell$.

5.4. Determination of the Number of Real Targets

The Bayesian information criterion (BIC) was used in our previous paper [13] for the Poisson measurement generation model assumption because this assumption leads to a cardinality selection problem formulation, which is similar to the K-means clustering problem and BIC is one of the widely used and trusted approaches [25] to determine the number of clusters (the number of targets in the present paper).

In the present paper, we assume a Bernoulli measurement generation model, which is more realistic than the Poisson model in the multiple transient emitter localization problem. Therefore, we used the likelihood function (binomial, in view of the Bernoulli model) based thresholding approach to determine the number of targets.

In the distributed algorithm, the estimated probability of detection $\hat{p}_i^{\mathrm{d}}$ will converge to the value in (112) for $n = 10$. Assume that the true probability of detection is high (say, above 0.9) and the number of nodes is large, we expect that most of the nodes have a measurement associated with a particular target. Therefore, for a real

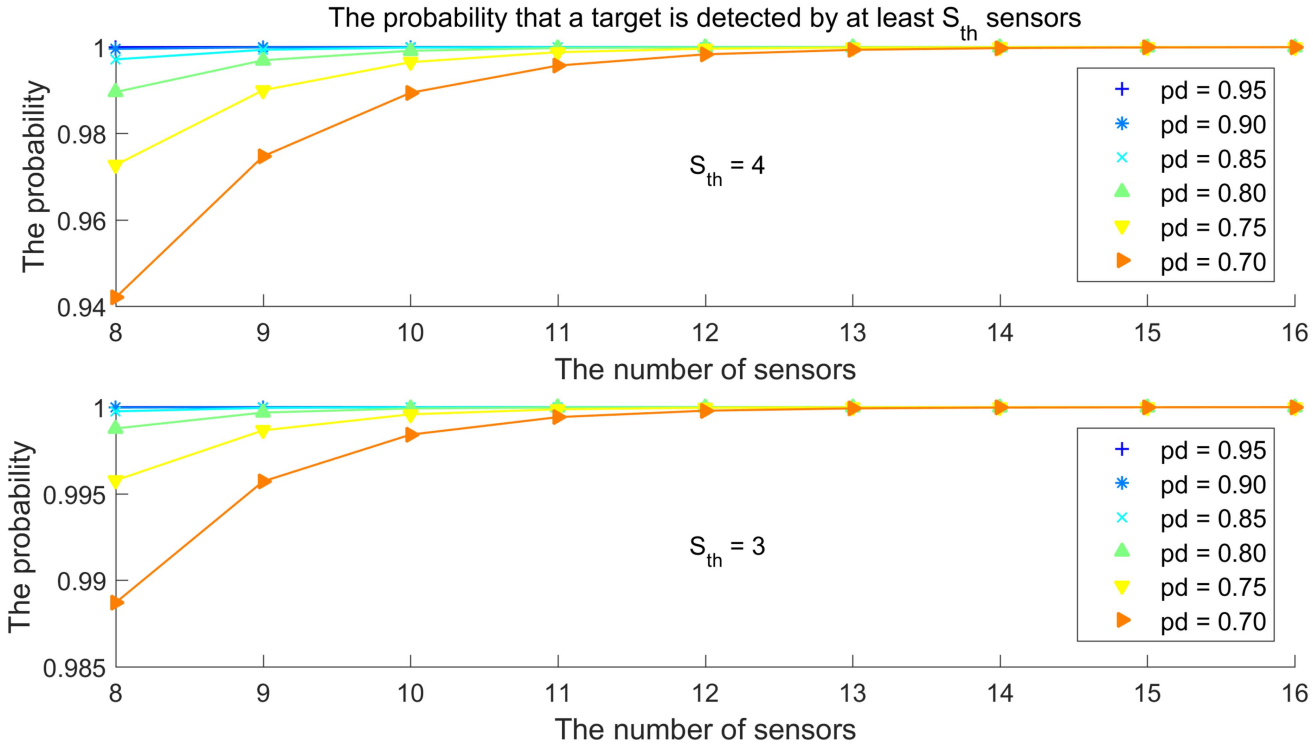


Fig. 2. The probability that a target is detected by at least S_{th} (3 or 4) sensors for varied values for the number of sensors and the probability of detection (pd).

TABLE VI
Local EM algorithm at node v_ℓ to find $\boldsymbol{\theta}_\ell^{k+1}$.

1: Initialization

$$\boldsymbol{\theta}_\ell^{(0)} = \boldsymbol{\theta}_\ell^{k} \tag{137}$$

2: **for** $n = 1,2,\ldots$ **do** until convergence
3: **E step**

$$\alpha_{\ell ji}^{(n-1)} = p_\ell(\kappa_{\ell j} = i \mid \mathbf{z}_{\ell j}, \boldsymbol{\theta}_\ell^{(n-1)}) = \frac{p_{i\ell}^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_{i\ell}^{(n-1)})}{\sum_{i=0}^{N} p_{i\ell}^{\mathrm{d}(n-1)} g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_{i\ell}^{(n-1)})} \tag{138}$$

$$Q(\boldsymbol{\theta}_\ell \mid \boldsymbol{\theta}_\ell^{(n-1)}) = \sum_{\kappa_\ell} p(\kappa_\ell \mid \psi_\ell, \boldsymbol{\theta}_\ell^{(n-1)}) \ln p(\psi_\ell^{\mathrm{M}} \mid \boldsymbol{\theta}_\ell) = Q(\mathbf{p}_\ell^{\mathrm{d}}) + Q(\boldsymbol{\Phi}_\ell)$$

$$Q(\mathbf{p}_\ell^{\mathrm{d}}) = -\sum_{i=0}^{N} p_{i\ell}^{\mathrm{d}} + \sum_{j=1}^{m_\ell} \sum_{i=0}^{N} \ln(p_{i\ell}^{\mathrm{d}}) \alpha_{\ell ji}^{(n-1)} \tag{139}$$

$$Q(\boldsymbol{\Phi}_\ell) = \sum_{j=1}^{m_\ell} \sum_{i=0}^{N} \ln(g_{\ell i}(\mathbf{z}_{\ell j} \mid \boldsymbol{\phi}_{i\ell})) \alpha_{\ell ji}^{(n-1)} \tag{140}$$

4: **M step**

$$\mathbf{p}_\ell^{\mathrm{d}(n)} = \arg\min_{\mathbf{p}_\ell^{\mathrm{d}}} -Q(\mathbf{p}_\ell^{\mathrm{d}}) + \lambda_{p\ell}^{kT}(\mathbf{p}_\ell^{\mathrm{d}} - \mathbf{p}^{\mathrm{d}k}) + \frac{\rho}{2}\|\mathbf{p}_\ell^{\mathrm{d}} - \mathbf{p}^{\mathrm{d}k}\|_2^2 \tag{141}$$

$$\boldsymbol{\Phi}_\ell^{(n)} = \arg\min_{\boldsymbol{\Phi}_\ell} -Q(\boldsymbol{\Phi}_\ell) + \lambda_{\phi\ell}^{kT}(\boldsymbol{\Phi}_\ell - \boldsymbol{\Phi}^k) + \frac{\rho}{2}\|\boldsymbol{\Phi}_\ell - \boldsymbol{\Phi}^k\|_2^2 \tag{142}$$

5: **end for**

target estimate, $\hat{p}_i^{\mathrm{d}}$ is likely to end up with a value close to 1. For a false target estimate, $\hat{p}_i^{\mathrm{d}}$ is likely to end up with a value close to 0, since only a few nodes have a measurement associated with a false target (which is the "same" across sensors, i.e., approximately at the same location). Based on this difference between real targets

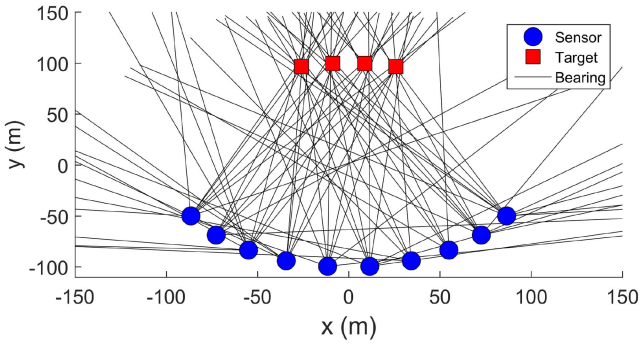


Fig. 3. A scenario with 10 targets and 4 sensors.

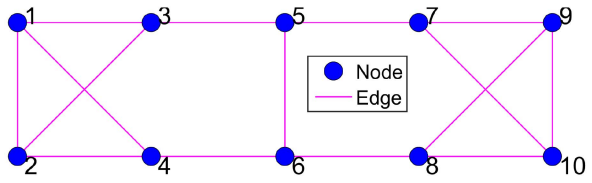


Fig. 4. The graph model of the wireless sensor network in Figure 3.

and false targets, it is reasonable to assume that there is a threshold value of $\hat{p}_i^{\mathrm{d}}$ that can be used to classify targets into either real or false.

If the number of sensors is known and the probability of detection is also known, then one can calculate the probability that a target is detected by at least S_{th} sensors. Figure 2 plots this probability for a range of values for the number of sensors and the probability of detection. Since even at p^{d}=0.7, the probability that a target is detected by at least 3 sensors is greater than 0.995 in most cases. We use the threshold value S_{th}=3. The corresponding threshold value of $\hat{p}_i^{\mathrm{d}}$ is

$$p_{\mathrm{th}}^{\mathrm{d}} = 0.3 \tag{144}$$

when $S = 10$ as in the simulation study. Therefore, we classify the targets with $\hat{p}_i^{\mathrm{d}}$ greater than 0.3 as real targets and otherwise the targets are deemed as false.

6. SIMULATION RESULTS

6.1. Scenario

Assume there are four targets ($N = 4$). The emission times of the acoustic events at the target locations are 0.4 s, 0.3 s, 0.1 s and 0.2 s, respectively. The speed of the acoustic signal is assumed to be 342 m/s. The measurement noise covariance matrix is

$$R_\ell = \begin{bmatrix} 7.6 \times 10^{-5} & 0 \\ 0 & 1 \times 10^{-4} \end{bmatrix} \tag{145}$$

i.e., the bearing standard deviation is $\sqrt{76}$ mrad = 0.5° and the TOA measurement standard deviation amounts to 10 ms, assumed to be the same for all sensors. The probability of detection for the targets is assumed to be 0.9 at all sensors. The time window W is chosen to be 1 s and the field of view of each sensor is from 0 to π. The density of the false alarms is set to be 1.27 $\mathrm{s}^{-1}\mathrm{radian}^{-1}$ such that the expected number of false alarms (N_{fa}) at each sensor is 4, which is equal to the number of real targets. Figure 3 shows one example using a wireless sensor network with 10 sensors numbered from left to right in an ascending order, which is represented by the graph model shown in Figure 4, to localize these 4 targets. Each node has three neighbors.

TABLE VII
CRLB and MSE with and without TOA measurements.

	Bearing	Bearing and TOA
CRLB (m^2)	2.6655	2.6464
MSE (m^2)	2.6396	2.6290

In the simulation, the targets and the sensors are located such that the angle between two LOS from two neighboring targets to any sensor is 5°, which is 10 times the standard deviation of LOS measurement noise, i.e. there are no unresolved measurements.

6.2. The significance of TOA measurements

The TOA measurements play an important role in the data association. The ghosting effect using bearing-only measurements is no longer present due to the additional estimation of a common signal emission time for the measurements associated with a single target. Here, we look at the improved estimation accuracy provided by the TOA measurements on top of the bearing-only measurements.

Assume that the data association is known and no missed detection or false alarms occurs, we want to localize a single target at the location (8.7 m, 99.6 m) with all measurements available at a fusion center. Table VII shows the Cramér-Rao lower bound (CRLB) and MSE of the target location using bearing-only measurements and bearing with TOA measurements. It shows that the improvement of the location estimation due to the additional TOA information is insignificant.

This implies that the TOA information should be only used in the sequential m-best assignment algorithm to obtain initial target estimates. Within the local EM algorithm, we can use only bearing measurements to reduce computational workload without significantly degrading the estimation accuracy.

6.3. Performance Metrics

In the following sections, we evaluate our distributed algorithm by two real-valued metrics for each Monte-Carlo run instead of averaging over all Monte-Carlo runs. These two metrics are the cardinality error for the number of targets and the root mean square (RMS) position error averaged over all targets. The latter is obtained by globally associating each location estimate to the nearest targets.

1) The cardinality error for the number of targets:

Given the true number of targets N_{t} and the estimated number of targets $\hat{N}$, the cardinality error is defined as

$$\tilde{N} = N_{\mathrm{t}} - \hat{N} \tag{146}$$

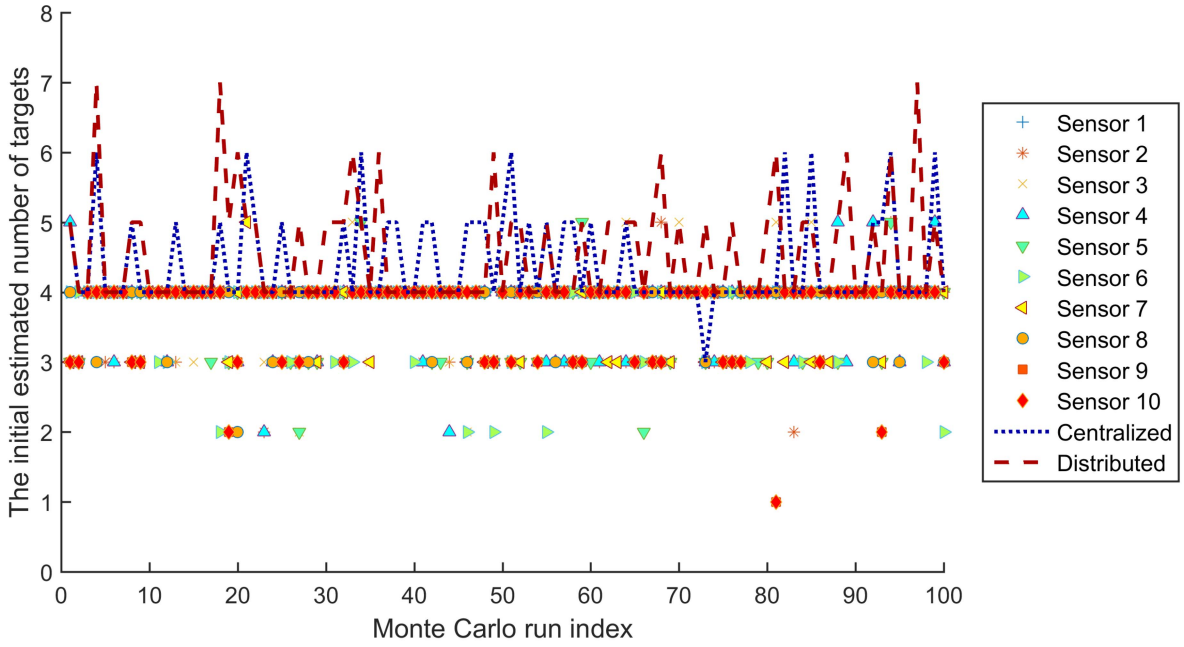


Fig. 5. The initially estimated number (the truth is 4) of targets by individual sensors, the centralized EM algorithm and the EM and AC based distributed ADMM algorithm.

2) The RMS position error:

Given the set of true positions of N_t targets

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_{N_t}, y_{N_t})\} \tag{147}$$

and the set of estimated positions of $\hat{N}$ targets

$$\{(\hat{x}_1, \hat{y}_1), (\hat{x}_2, \hat{y}_2), \ldots, (\hat{x}_{\hat{N}}, \hat{y}_{\hat{N}})\} \tag{148}$$

there are three cases. Let Π_N denote all permutations of the set $\{1, 2, \ldots, N\}$.

Case 1: $N_t = \hat{N}$. The RMS position error is defined as

$$\mathrm{RMS}_p = \min_{\pi \in \Pi_{N_t}} \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} [(x_i - \hat{x}_{\pi(i)})^2 + (y_i - \hat{y}_{\pi(i)})^2]} \tag{149}$$

Case 2: $N_t < \hat{N}$. The RMS position error is defined as

$$\mathrm{RMS}_p = \min_{\pi \in \Pi_{\hat{N}}} \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} [(x_i - \hat{x}_{\pi(i)})^2 + (y_i - \hat{y}_{\pi(i)})^2]} \tag{150}$$

Case 3: $N_t > \hat{N}$. The RMS position error is defined as

$$\mathrm{RMS}_p = \min_{\pi \in \Pi_{N_t}} \sqrt{\frac{1}{\hat{N}} \sum_{i=1}^{\hat{N}} [(\hat{x}_i - x_{\pi(i)})^2 + (\hat{y}_i - y_{\pi(i)})^2]} \tag{151}$$

Note that we need to combine these two real-valued metrics (146 and one of 149–151) in order to have a complete evaluation of the algorithm performance.

6.4. Performance of the EM and AC based distributed ADMM algorithm

For the algorithm evaluation, the target measurements are generated according to a Bernoulli measurement model, specifically, one measurement from each target is generated for each sensor with a probability p_d or nothing with a probability $1 - p_d$. The false alarms are generated for each sensor according to the Poisson model (80) and (81).

Note that the values of the probability of detection, p_d, and the expected number of false alarms, N_{fa}, are required to generate the target measurements. However, the EM and AC based distributed ADMM algorithm do not need to know the values of N_{fa} and p_d. They adapt to these values by "learning them."

We used 100 Monte-Carlo runs to evaluate the performance of our distributed algorithm and make comparisons with a modified version of the centralized algorithm in [13]. Both used the same threshold (144) to determine the number of targets.

Figure 5 shows the number of targets initially estimated by each sensor using the sequential m-best 2-D assignment algorithm on the measurements of its own and its one-hop neighbors. It can be observed that this number is different from sensor to sensor because of the missed detections and false alarms, which is the motivation for the development of the distributed set consensus algorithm described in the Sections V-A and V-B. In the same plot, the centralized algorithm (denoted by "Centralized") obtained the initial estimated number of targets by using the sequential m-best 2-D assignment algorithm on the measurements from all sensors. In contrast, the distributed algorithm obtained the initial estimate (the same for all sensors) of the number of targets via the distributed set consensus algorithm and this estimate is also the estimated number of candidate targets. Since the centralized and distributed algorithms

TABLE VIII
Evolution of each sensor's target location estimates (each row represents a target location estimate) at key stages of the initialization consensus process.

sensor index	initial estimates by SEQ[m(2-D)]	consensus on the number of targets: after 1 iteration	consensus on the number of targets: after 3 iterations	consensus on the estimates: after 1 iteration	consensus on the estimates: after 25 iterations
1	$\begin{bmatrix} -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.89 & 104.59 \\ 7.19 & 94.37 \\ 26.03 & 99.83 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.86 & 95.19 \\ -7.83 & 103.36 \\ 8.28 & 96.08 \\ 25.65 & 98.66 \\ -104.90 & -21.35 \end{bmatrix}$
2	$\begin{bmatrix} -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.45 & 106.09 \\ 7.16 & 94.15 \\ 26.82 & 101.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -5.89 & 104.59 \\ 7.19 & 94.37 \\ 26.03 & 99.83 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.86 & 95.19 \\ -7.83 & 103.36 \\ 8.28 & 96.08 \\ 25.65 & 98.66 \\ -104.90 & -21.35 \end{bmatrix}$
3	$\begin{bmatrix} -6.05 & 105.04 \\ 6.21 & 92.11 \\ 23.71 & 96.02 \end{bmatrix}$	$\begin{bmatrix} -6.05 & 105.04 \\ 6.21 & 92.11 \\ 23.71 & 96.02 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -6.05 & 105.04 \\ 6.21 & 92.11 \\ 23.71 & 96.02 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -26.72 & 94.12 \\ -6.29 & 104.05 \\ 7.04 & 93.26 \\ 25.84 & 99.59 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.85 & 95.22 \\ -7.86 & 103.37 \\ 8.29 & 96.08 \\ 25.64 & 98.64 \\ -104.90 & -21.35 \end{bmatrix}$
4	$\begin{bmatrix} -27.05 & 91.42 \\ -6.63 & 101.14 \\ 8.24 & 97.05 \\ 26.75 & 101.26 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -6.63 & 101.14 \\ 8.24 & 97.05 \\ 26.75 & 101.26 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -27.05 & 91.42 \\ -6.63 & 101.14 \\ 8.24 & 97.05 \\ 26.75 & 101.26 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -26.72 & 94.12 \\ -5.89 & 101.15 \\ 7.78 & 99.08 \\ 26.21 & 99.98 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.85 & 95.22 \\ -7.86 & 103.37 \\ 8.29 & 96.08 \\ 25.64 & 98.64 \\ -104.90 & -21.35 \end{bmatrix}$
5	$\begin{bmatrix} -8.22 & 98.99 \\ 7.64 & 92.64 \\ 26.01 & 100.30 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 102.20 \\ -8.22 & 98.99 \\ 7.64 & 92.64 \\ 26.01 & 100.30 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 102.20 \\ -8.22 & 98.99 \\ 7.64 & 92.64 \\ 26.01 & 100.30 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.83 & 97.86 \\ -6.92 & 98.83 \\ 7.51 & 97.09 \\ 24.78 & 98.37 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.79 & 95.33 \\ -7.98 & 103.42 \\ 8.35 & 96.10 \\ 25.63 & 98.58 \\ -104.90 & -21.35 \end{bmatrix}$
6	$\begin{bmatrix} -25.73 & 102.20 \\ -6.02 & 91.28 \\ 8.56 & 110.96 \\ 24.45 & 96.61 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 102.20 \\ -6.02 & 91.28 \\ 8.56 & 110.96 \\ 24.45 & 96.61 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 102.20 \\ -6.02 & 91.28 \\ 8.56 & 110.96 \\ 24.45 & 96.61 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.11 & 96.95 \\ -7.50 & 98.81 \\ 8.69 & 99.10 \\ 25.70 & 98.53 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.79 & 95.33 \\ -7.98 & 103.42 \\ 8.35 & 96.10 \\ 25.63 & 98.58 \\ -104.90 & -21.35 \end{bmatrix}$
7	$\begin{bmatrix} -24.83 & 95.60 \\ -7.40 & 100.00 \\ 24.94 & 100.57 \end{bmatrix}$	$\begin{bmatrix} -24.83 & 95.60 \\ -7.40 & 100.00 \\ 7.64 & 92.64 \\ 24.94 & 100.57 \end{bmatrix}$	$\begin{bmatrix} -24.83 & 95.60 \\ -7.40 & 100.00 \\ 7.64 & 92.64 \\ 24.94 & 100.57 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.51 & 98.36 \\ -10.26 & 105.19 \\ 8.97 & 94.20 \\ 25.53 & 98.47 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 95.44 \\ -8.10 & 103.47 \\ 8.41 & 96.12 \\ 25.61 & 98.52 \\ -104.90 & -21.35 \end{bmatrix}$
8	$\begin{bmatrix} -21.92 & 91.96 \\ -9.15 & 103.81 \\ 10.31 & 95.76 \\ 25.59 & 95.97 \end{bmatrix}$	$\begin{bmatrix} -21.92 & 91.96 \\ -9.15 & 103.81 \\ 10.31 & 95.76 \\ 25.59 & 95.97 \end{bmatrix}$	$\begin{bmatrix} -21.92 & 91.96 \\ -9.15 & 103.81 \\ 10.31 & 95.76 \\ 25.59 & 95.97 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -24.78 & 97.45 \\ -10.15 & 104.22 \\ 9.87 & 99.56 \\ 25.30 & 96.40 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.73 & 95.44 \\ -8.10 & 103.47 \\ 8.41 & 96.12 \\ 25.61 & 98.52 \\ -104.90 & -21.35 \end{bmatrix}$
9	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 25.58 & 96.51 \end{bmatrix}$	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 10.31 & 95.76 \\ 25.58 & 96.51 \end{bmatrix}$	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 10.31 & 95.76 \\ 25.58 & 96.51 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -24.56 & 95.80 \\ -10.50 & 106.40 \\ 9.64 & 94.98 \\ 25.43 & 97.39 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.72 & 95.47 \\ -8.13 & 103.49 \\ 8.43 & 96.12 \\ 25.61 & 98.50 \\ -104.90 & -21.35 \end{bmatrix}$
10	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 25.58 & 96.51 \end{bmatrix}$	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 10.31 & 95.76 \\ 25.58 & 96.51 \end{bmatrix}$	$\begin{bmatrix} -25.74 & 97.82 \\ -12.72 & 110.89 \\ 10.31 & 95.76 \\ 25.58 & 96.51 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -24.56 & 95.80 \\ -10.50 & 106.40 \\ 9.64 & 94.98 \\ 25.43 & 97.39 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.72 & 95.47 \\ -8.13 & 103.49 \\ 8.43 & 96.12 \\ 25.61 & 98.50 \\ -104.90 & -21.35 \end{bmatrix}$

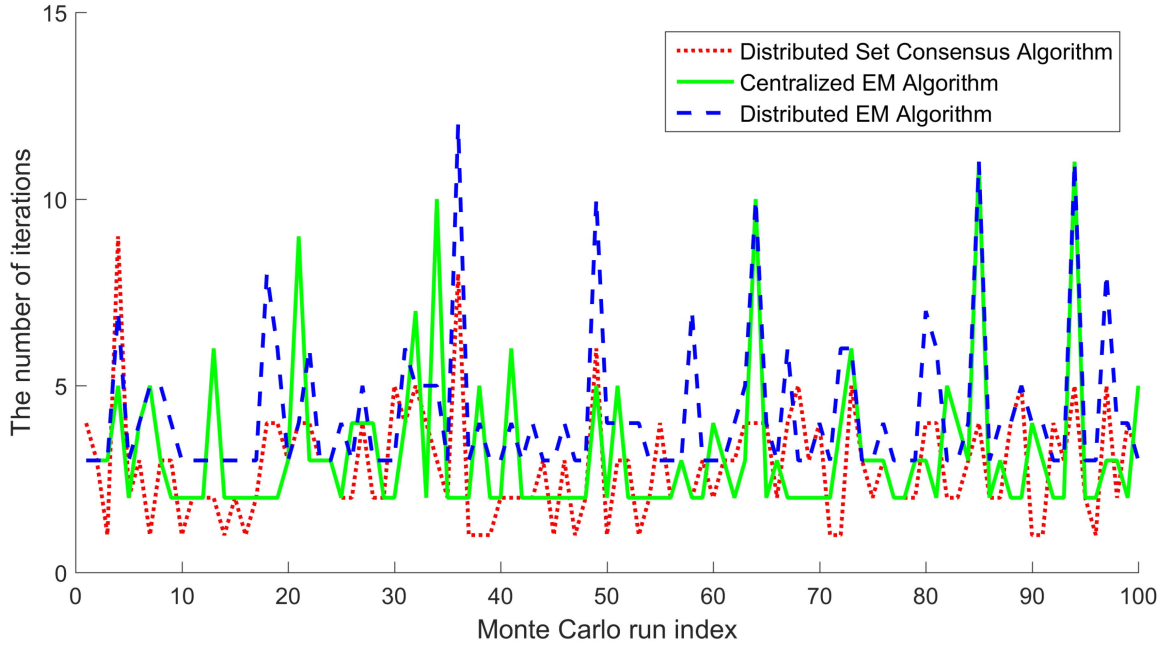


Fig. 6. The number of iterations of the distributed set consensus algorithm, the centralized EM algorithm and the EM and AC based distributed ADMM algorithm.

TABLE IX
Evolution of the target location estimates (same for each sensor) throughout the distributed EM algorithm.

Initial estimates (final from initialization consensus, 60 iterations)	After 2 iterations (converged)	After removing false targets
$\begin{bmatrix} -25.79 & 95.33 \\ -7.98 & 103.42 \\ 8.35 & 96.10 \\ 25.63 & 98.58 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.29 & 97.01 \\ -7.56 & 100.05 \\ 8.95 & 98.45 \\ 25.19 & 98.32 \\ -104.90 & -21.35 \end{bmatrix}$	$\begin{bmatrix} -25.29 & 97.01 \\ -7.56 & 100.05 \\ 8.95 & 98.45 \\ 25.19 & 98.32 \end{bmatrix}$

use different initialization approaches, the initially estimated number of targets is different for the two algorithms. False targets appear in 40 runs, where the estimated number of candidate targets is greater than the true number of targets. Tables VIII and IX illustrate the consensus and distributed EM processes.

Figure 6 shows the number of iterations required for the convergence of the different iterative algorithms presented in this paper. All the algorithms terminate in a few iterations. The EM and AC based distributed algorithm, being itself an iterative algorithm, consists of three steps, two of which are iterative algorithms themselves (steps on Lines 5 and 6 in Table V). By close examination, we found that average number of iterations for these two algorithms is around 3 and 9, respectively. Since local communication only occurs at the AC step (Line 6 in Table V), the average number of communications for each sensor is approximately 50.

Figure 7 plots the number of targets estimated by the centralized and distributed algorithms before and after thresholding. Since the initialization is different for these two EM algorithms, the estimated number of targets is slightly different. In the shooter localization application, the priority is to avoid any missed target and then try to avoid as many false targets as possible. There are two possible sources for the false targets in the final solution. One is that the false alarm rate is high, which can inevitably cause the presence of some false targets. The other is that a target is split into two close targets due to the association test. While the former may cause confusion in the decision making, we may prioritize the targets based on the estimated p^{d} such that the low $\hat{p}^{\mathrm{d}}$ targets have the low priority. The latter may be solved by looking at whether two close targets with low estimated probabilities of detection have their sum close to 1.

The top plot in Figure 8 shows the RMS position error (averaged over all targets) for different cases before we remove the predicted false targets. The "Known Association," which refers to the situation when we know the number of targets and the association between measurements and targets, is meant to serve as a baseline or a lower bound (which is unachievable). In this case, the position estimates can be obtained separately for each target by solving a nonlinear least squares problem, and subsequently the position error can easily be obtained. From the same plot, it can be observed that the distributed algorithm yields the same position error as the centralized algorithm most of the time. While the baseline serves as a lower bound in most cases, it is interesting to note that the performance of the centralized algorithm or the distributed algorithm is better than the baseline in a few situations, which is due to "useful" false measurements.

The bottom plot in Figure 8 shows the RMS position error (averaged over all targets) for different cases

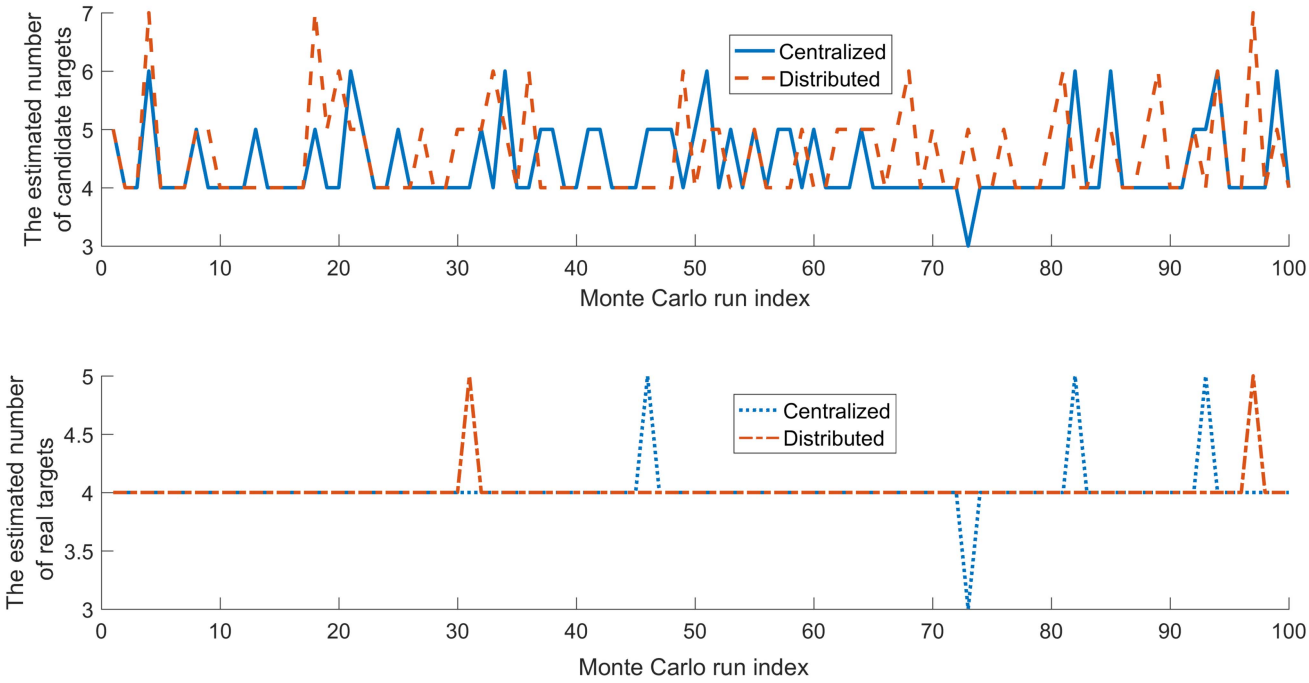


Fig. 7. The number of targets (the truth is 4) estimated by the centralized and distributed algorithms before (top plot) and after (bottom plot) removing false targets using the threshold (144).

Fig. 8. The RMS position error per target evaluated by assuming known target-measurement association, using the centralized EM algorithm and the EM and AC based distributed ADMM algorithm before and after removing the false targets.

after we remove the low $\hat{p}^{\mathrm{d}}$ false targets. This is also a measure of accuracy of the final position estimates provided by the centralized and distributed algorithm. For a clearer comparison, the range of the RMS ratio of the distributed algorithm over the centralized algorithm is also shown as in Table X. While the distributed algorithm can produce a higher error than the centralized algorithm occasionally, in most cases (84%), it yields practically the same localization result as the centralized algorithm. It is interesting to note that the distributed algorithm can be slightly better than the centralized algorithm due to a different initialization.

7. CONCLUSION

This paper considers passive localization of multiple transient emitters using a wireless sensor network and develops a distributed algorithm, which relies solely on local communications between one-hop neighboring

TABLE X
The final (after removing low pd targets) RMS ratio among assuming known association (KA), the centralized algorithm (C) and the distributed algorithm (D).

Interval	(0.9,0.99)	[0.99,1.01]	(1.01,1.1)	[1.1,1.3)	[1.3,1.5)	[1.5,8]
D versus C	3	84	3	6	1	3
C versus KA	12	75	3	4	2	2
D versus KA	15	60	6	9	6	4

sensors. A distributed implementation of the centralized EM based algorithm is not possible unless the consensus on the initial set of estimates can be reached among all sensors. It is shown by simulation that even with the knowledge of data association, we need to carry out the initialization carefully because of the bearing measurements. Random initializations based on individual sets of bearing measurements could converge to a local minimum, therefore it is necessary to use the bearing measurements from neighboring sensors. As in the centralized EM based algorithm, each node uses a sequential m-best 2-D assignment algorithm on measurements from itself and its neighbors to obtain an initial set of target estimates.

Since initially estimated target set can be different from node to node in terms of both the cardinality (this happens when the probability of detection is low or the false alarm rate is high) and the values of estimates (since each sensor uses different measurements for initialization), we developed a distributed set consensus algorithm to reach consensus on the number of candidate targets before each node can reach consensus on the target-estimate association so that a proper initialization is obtained for the EM and AC based distributed ADMM algorithm. Since a Bernoulli measurement generation model is a more realistic assumption as it reflects best the physical process of measurement generation, we presented a likelihood function based thresholding technique to determine the number of targets.

Simulation results show that the EM and AC based distributed ADMM algorithm converges very fast and yields the target location estimates that are almost as good as those of the centralized algorithm. The estimated probability of detection is shown to be able to effectively distinguish real targets from false targets.

REFERENCES

[1] J. Areta, Y. Bar-Shalom, M. Levedahl, and K. Pattipati
Hierarchical track association and fusion for a networked surveillance system.
Journal of Advances in Information Fusion, 1, 2 (Dec. 2006), 140–157.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.
Wiley, 2001.

[3] Y. Bar-Shalom, P. Willett, and X. Tian
Tracking and Data Fusion.
YBS Publishing, 2011.

[4] C. M. Bishop
Pattern Recognition and Machine Learning.
Springer-Verlag, 2006.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein
Distributed optimization and statistical learning via the alternating direction method of multipliers.
Foundations and Trends in Machine Learning, 3, 1 (Jan. 2011), 1–222.

[6] C. Y. Chong
Hierarchical estimation.
In *2nd MIT/ONR Workshop on Distributed Information and Decision Systems Motivated by Naval Command-Control-Communication (C3) Problems*, July 1979.

[7] C. Y. Chong, K. C. Chang and S. Mori
Comparison of optimal distributed estimation and consensus filtering.
In *Proceedings of 19th International Conference on Information Fusion*, Heidelberg, July 2016.

[8] C. Y. Chong, S. Mori and K. C. Chang
Distributed Multitarget Multisensor Tracking.
Multitarget-Multisensor Tracking: Applications and Advances, Volume I, Yaakov Bar-Shalom, Editor, Chapter 8, Artech House, 1990.

[9] C. Y. Chong, S. Mori, E. Tse, and R. P. Wishner
Distributed estimation in distributed sensor networks.
In *Proceedings of the 1982 American Control Conference*, 1982.

[10] C. Y. Chong, E. Tse, and S. Mori
Distributed estimation in networks.
In *Proceedings of the 1983 American Control Conference*, 1983.

[11] D. Daley, and D. Vere-Jones
An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods.
Springer-Verlag, 2003.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin
Maximum likelihood from incomplete data via the em algorithm.
Journal of the Royal Statistical Society, Series B, 39, 1 (Aug. 1977), 1–38.

[13] W. Dou, J. George, L. Kaplan, R. W. Osborne, and Y. Bar-Shalom
Evaluation of fusion algorithms for passive localization of multiple transient emitters.
Journal of Advances in Information Fusion, under review.

[14] H. Durrant-Whyte and M. Stevens
Data fusion in decentralised sensing networks.
In *Proceedings of 4th International Conference on Information Fusion*, 2001.

[15] P. A. Forero, A. Cano, and G. B. Giannakis
Distributed clustering using wireless sensor networks.
IEEE Journal of Selected Topics in Signal Processing, 5, 4 (Aug. 2011), 707–724.

[16] J. George, and L. Kaplan
Shooter localization using a wireless sensor network of soldier-worn gunfire detection systems.
Journal of Advances in Information Fusion, 8, 1 (June 2013), 15–32.

[17] J. George, L. Kaplan, S. Deligeorges, and G. Cakiades
Multi-shooter localization using finite point process.
In *Proceedings of Seventeenth International Conference on Information Fusion*, July 2014.

[18] D. Gu
Distributed EM algorithm for Gaussian mixtures in sensor networks.
IEEE Transactions on Neural Networks, 19, 7 (July 2008), 1154–1166.

[19] V. Indelman, E. Nelson, J. Dong, N. Michael and F. Dellaert
Incremental distributed inference from arbitrary poses and unknown data association: using collaborating robots to establish a common reference.
IEEE Control Systems Magazine, 36, 2(Apr. 2016), 41–74.

[20] H. W. Kuhn, and A. W. Tucker
Nonlinear Programming.
In *Proceedings of Second Berkeley Symposium on Mathematical Statistics and Probability*, 1951, 481–492.

[21] T. W. Martin and K. C. Chang
A distributed data fusion approach for mobile ad hoc networks.
In *Proceedings of 7th International Conference on Information Fusion*, 2005.

[22] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel
D-ADMM: a communication-efficient distributed algorithm for separable optimization.
IEEE Transactions on Signal Processing, 61, 10 (May 2013), 2718–2723.

[23] R. D. Nowak
Distributed EM algorithms for density estimation and clustering in sensor networks.
IEEE Transactions on Signal Processing, 51, 8 (Aug. 2003), 2245–2253.

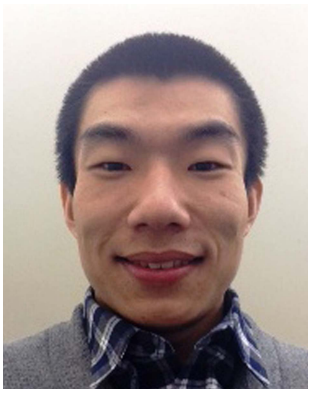
[24] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn
A new relaxation algorithm and passive sensor data association.
IEEE Transactions on Automatic Control, 37, 2 (Feb. 1992), 198–213.

[25] D. Pelleg and A. W. Moore
X-means: extending K-means with efficient estimation of the number of clusters.
In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

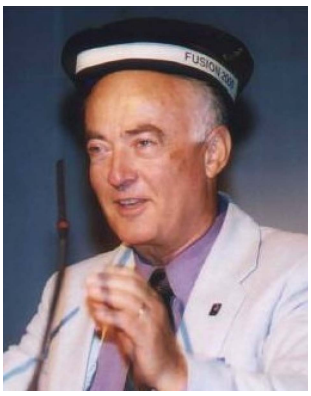
[26] I. D. Schizas, A. Ribeiro, and G. B. Giannakis
Consensus in Ad Hoc WSNs with noisy links—part I: distributed estimation of deterministic signals.
IEEE Transactions on Signal Processing, 56, 1 (Jan. 2008), 350–364.

[27] L. Xiao, and S. Boyd
Fast linear iterations for distributed averaging.
Systems and Control Letters, 53 (2004), 65–78.

[28] L. Xiao, S. Boyd, and S. Lall
A scheme for robust distributed sensor fusion based on average consensus.
In *Fourth International Symposium on Information Processing in Sensor Networks*, Apr. 2005, 63–70.

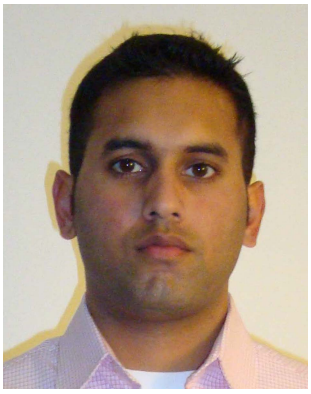


Wenbo Dou is a graduate student in the Department of Electrical and Computer Engineering at the University of Connecticut. He received the Bachelor of Engineering in Electrical and Electronic Engineering with First Class Honors from Nanyang Technological University in 2011. His current research interests include bias estimation, multi-target localization and tracking, data fusion and machine learning.



Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering. From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab. His current research interests are in estimation theory, target tracking and data fusion. He has published over 500 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993; translated into Russian, MGTU Bauman, Moscow, Russia, 2011), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate texts *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), *Tracking and Data Fusion* (YBS Publishing, 2011),and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multi-target tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas. During 1976 and 1977 he served as Associate Editor of the *IEEE Transactions on Automatic Control* and from 1978 to 1981 as Associate Editor of *Automatica*. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–13. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is co-recipient of the M. Barry Carlton Award for the best paper in the *IEEE Transactions on Aerospace and Electronic Systems* in 1995 and 2000 and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008 he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012 the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in Aerospace Engineering based on the citations of his work. He is the recipient of the 2015 ISIF Award for a Lifetime of Excellence in Information Fusion.

Lance M. Kaplan received the B.S. degree with distinction from Duke University, Durham, NC, in 1989 and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1991 and 1994, respectively, all in Electrical Engineering. From 1987–1990, Dr. Kaplan worked as a Technical Assistant at the Georgia Tech Research Institute. He held a National Science Foundation Graduate Fellowship and a USC Dean's Merit Fellowship from 1990–1993, and worked as a Research Assistant in the Signal and Image Processing Institute at the University of Southern California from 1993–1994. Then, he worked on staff in the Reconnaissance Systems Department of the Hughes Aircraft Company from 1994–1996. From 1996–2004, he was a member of the faculty in the Department of Engineering and a senior investigator in the Center of Theoretical Studies of Physical Systems (CTSPS) at Clark Atlanta University (CAU), Atlanta, GA. Currently, he is a researcher in the Networked Sensing and Fusion branch of the U.S Army Research Laboratory (ARL). Dr. Kaplan serves as Editor-In-Chief for the *IEEE Transactions on Aerospace and Electronic Systems* (AES) and as VP of Conferences for the International Society of Information Fusion (ISIF). Previously, he served on the Board of Governors of the IEEE AES Society (2008–2013) and on the Board of Directors of ISIF (2012–2014). He is a three time recipient of the Clark Atlanta University Electrical Engineering Instructional Excellence Award from 1999–2001. He is a Fellow of IEEE and ARL. His current research interests include signal and image processing, information/data fusion, network science and resource management.



Jemin George received his M.S. ('07), and Ph.D. ('10) in Aerospace Engineering from the State University of New York at Buffalo. In 2008, he was a Summer Research Scholar with the U.S. Air Force Research Laboratory's Space Vehicles Directorate and in 2009, he was a National Aeronautics and Space Administration Langley Aerospace Research Summer Scholar. From 2009–2010 he was a Research Fellow with the Department of Mathematics, Technische Universität Darmstadt, Darmstadt, Germany. Since 2010, he has been with the U.S. Army Research Laboratory. His principal research interests include stochastic systems, control theory, nonlinear filtering, information fusion, adaptive networks, distributed sensing and estimation.

Maneuvering Target Tracking Using Continuous Wave Bistatic Sonar with Propagation Delay

RONG YANG
YAAKOV BAR-SHALOM
CLAUDE JAUFFRET
ANNIE-CLAUDE PÉREZ
GEE WAH NG

Acoustic propagation delay has not been investigated for a continuous wave multistatic sonar tracking system except for the recent study conducted by Jauffret et al. [6], which estimates the trajectory of a constant velocity target. The results showed that the estimate bias caused by the propagation delay is not negligible, especially for a bistatic system. This paper develops an interacting multiple model unscented Gauss-Helmert filter with numerical Jacobian (IMM-UGHF-NJ) to track a maneuvering target with propagation delay using a bistatic sonar system. The IMM-UGHF-NJ can overcome the two tracking challenges introduced by the delay, namely, implicit state transition model and lack of analytical expression of the Doppler shifted frequency in the measurement model. Simulation tests have been conducted, and the results show that the IMM-UGHF-NJ can reduce the estimation error significantly, especially for long range or fast moving targets.

Manuscript received December 7, 2016; revised May 17, 2017; released for publication May 22, 2017.

Refereeing of this contribution was handled by Paolo Braca.

Authors' addresses: R. Yang and G. W. Ng, DSO National Laboratories, 12 Science Park Drive, Singapore 118225 (E-mail: yrong@dso.org.sg; ngeewah@dso.org.sg). Y. Bar-Shalom, Department of ECE, University of Connecticut, Storrs, CT 06269, USA (E-mail: ybs@engr.uconn.edu). C. Jauffret and A.-C. Pérez, University of Toulon, IM2NP, UMR7334, CS 60584, 83041 TOULON Cedex, France (E-mail: jauffret@univ-tln.fr; annie-claude.perez@univ-tln.fr).

Y. Bar-Shalom was supported by ARO Grant W911NF-10-1-0369.

1557-6418/18/$17.00

I. INTRODUCTION

Continuous active sonar (CAS), also known as high duty cycle (HDC) sonar, with multistatic setup has attracted the research interest recently. In such a system, the signal is transmitted almost in a full duty cycle. Compared to the commonly used pulse active sonar (PAS) system, which transmits only a short pulse in a cycle, the CAS system has continuous detection capability. Furthermore, a CAS usually uses low intensity signal, so it is less disturbing to underwater fauna.

There are two main types of CAS systems according to the signal waveforms transmitted, namely, frequency modulated (FM) waveforms and continuous constant frequency waveforms (CW). The FM-CAS can provide good target bistatic range information, whereas the CW-CAS has good Doppler shifted frequency measurement (linked to target range rate). The FM-CAS needs to separate indirect path signal from strong direct path signal via methods, such as m-sequence modulation [3] and Dopplergram [10]. The FM-CAS has a frequency bandwidth limitation issue in multistatic system, as broadband waveforms are transmitted repeatedly [5]. The CW-CAS transmits a single fixed frequency waveform, so that it has no frequency bandwidth limitation problem as the FM-CAS. However, due to lack of range information, the observability of a target trajectory in CW-CAS is not as good as FM-CAS, especially for bistatic (a single transmitter-receiver pair) system.

We focus on target tracking using CW-CAS in this paper. A few approaches to this problem have been proposed in literature. A Gaussian mixture probability hypothesis density (GMPHD) filter was developed in [5]. It tracks multiple constant velocity (CV) targets using bearings and Doppler frequencies detected by multistatic CW-CAS. Results show that CV targets can be tracked using more than two transmitter-receiver pairs when target range is not available. This research does not take signal propagation delay into consideration. The effect of propagation delay of CW-CAS has been studied in [6] recently. An exact Doppler frequency model with propagation delay was proposed. and a maximum likelihood (ML) estimator based on this model was developed to perform batch estimation[1] for a CV target. The simulation results showed that the estimation bias induced by the propagation delay is not negligible, especially for a bistatic system.

In this paper, the propagation delay problem raised in [6] is studied further. We extend the target CV trajectory estimation using a batch parameter estimation technique to the dynamic recursive estimation, which can handle not only CV motion but also maneuvering motion. This extension faces two challenges. Firstly, the "target time" t_k and target position (x_k, y_k) in the state

[1]The batch estimation estimates target parameter using a batch of measurements received in a certain duration. The target parameter describes the target motion, for example, target initial position and its velocity (assumed constant for the duration of the batch).

[defined later in (1) and (2)] are highly correlated after propagation delay is introduced. This leads to a state transition equation in an implicit form instead of the commonly used explicit form in [11][12]. Secondly, the Doppler shifted frequency (one of the measurements) does not have an analytical expression in terms of the target state. This is because the Doppler frequency is a function of the bistatic range rate which cannot be described analytically after propagation delay is introduced. Details will be given later in Section II-B. The two challenges mentioned above were overcome in [6] by solving a 2nd order polynomial equation for CV target. However, the approach in [6] cannot be applied to a maneuvering target with coordinated turn (CT) motion, and the new approach in this paper will be shown to handle this.

A dynamic estimation problem uses two basic models, namely, the state transition model and the measurement model. The state transition model describes the evolution of the target state with time, and it is (in most cases) an explicit expression of the state at the current time in terms of the state at the previous time. The measurement model relates the measurement to the state. The two challenges of the dynamic estimation problem considered in this paper are: (i) the implicit state transition model; (ii) the lack of an analytical measurement model. These make this problem impossible to solve using existing filters.

We will develop an interacting multiple model unscented Gauss-Helmert filter with numerical Jacobian (IMM-UGHF-NJ) to cope with the challenges mentioned above. The IMM [2] is a well known hybrid algorithm to handle motion model uncertainty in maneuvering target tracking. The UGHF [11][12][14][15] is a recently developed algorithm for bearings-only tracking (BOT) with implicit state transition model introduced by the acoustic propagation delay. It can be applied to our problem. For the measurement model without analytical form, the NJ (numerical Jacobian) algorithm, which computes the Jacobian numerically, can be utilized [8][13][9]. The Doppler shifted frequency is a function of the bistatic range rate, $\dot{r}$, which has no analytical form due to the unknown time delay. We can compute $\dot{r}$ (derivative of range r) using the NJ.

The structure of the rest of paper is as follows. Section II formulates the problem. Section III presents the IMM-UGHF-NJ. Simulation results and conclusions are in Sections IV and V, respectively.

II. PROBLEM FORMULATION

The problem is illustrated in Fig. 1. At dynamic estimation cycle k, the transmitter emits a CW signal with constant frequency f^{T} at time t_k^{T}, and the receiver receives the Doppler shifted frequency f^{R} at time t_k^{R} via the target reflection at time t_k. We assume the transmitter and receiver are stationary and located at $(x^{\mathrm{T}},y^{\mathrm{T}})$ and $(x^{\mathrm{R}},y^{\mathrm{R}})$, respectively. The target is moving

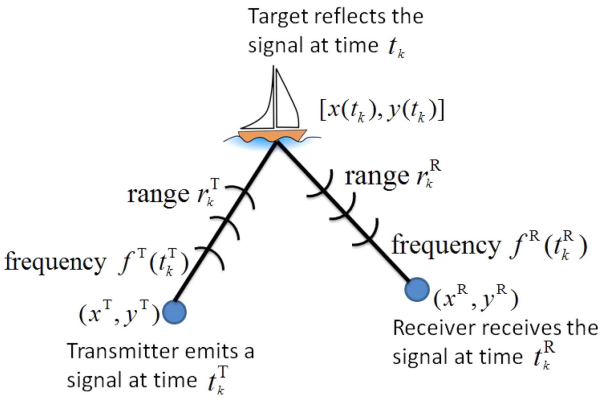


Fig. 1. Signal transmission of CW bistatic sonar.

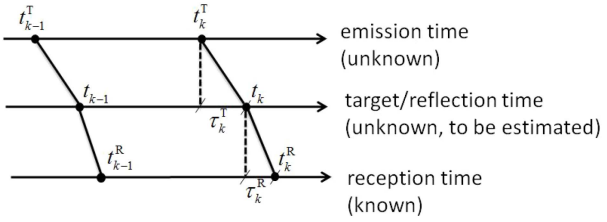


Fig. 2. Time sequences of continuous wave bistatic sonar.

and its location is $[x(t_k),y(t_k)]$ at reflection time t_k. The ranges between the target at t_k to the transmitter and the receiver are r_k^{T} and r_k^{R}, respectively. We also assume that sound propagation is straight with a nearly constant speed among the transmitter, target and receiver.

The target states to be estimated for the CV and CT models at time t_k are

$$\mathbf{x}^{\mathrm{CV}}(t_k) = [x(t_k) \quad y(t_k) \quad \dot{x}(t_k) \quad \dot{y}(t_k) \quad t_k]' \tag{1}$$

$$\mathbf{x}^{\mathrm{CT}}(t_k) = [x(t_k) \quad y(t_k) \quad \dot{x}(t_k) \quad \dot{y}(t_k) \quad \omega(t_k) \quad t_k]' \tag{2}$$

where x, y, $\dot{x}$ and $\dot{y}$ are the target positions and velocities in the x and y coordinates, respectively, ω is the target turn rate, and t_k is the target time (or reflection time) corresponding to the emission time t_k^{T} and the reception time t_k^{R} of the transmitter and receiver, respectively. The measurement vector at time t_k^{R} is

$$\mathbf{z}(t_k^{\mathrm{R}}) = [b(t_k^{\mathrm{R}}) \quad f^{\mathrm{R}}(t_k^{\mathrm{R}})]' \tag{3}$$

where b is the target bearing from the receiver at time t_k^{R} to the target at time t_k, measured clockwise from True North, and f^{R} is the Doppler shifted frequency at the receiver.

A. State transition models

The state transition model describes the evolution of the target state with time. For a generic discrete problem, it is an explicit form given by

$$\mathbf{x}(t_k) = \mathbf{f}[\mathbf{x}(t_{k-1})] + \Gamma\,\mathbf{v}(t_{k-1}) \tag{4}$$

where k is the discrete estimation cycle index, $\mathbf{v}(t_{k-1})$ is the process noise, and Γ is the process noise gain. However, there is no explicit state transition model for

our problem. It can be seen from Fig. 2 that the target time, t_k, is unknown due to the unknown propagation delay τ^{R}. There is an implicit constraint between the known t_k^{R} and unknown t_k given by

$$t_k = t_k^{\mathrm{R}} - \tau_k^{\mathrm{R}} \tag{5}$$

where

$$\tau_k^{\mathrm{R}} = \frac{\sqrt{[x(t_k) - x^{\mathrm{R}}]^2 + [y(t_k) - y^{\mathrm{R}}]^2}}{c^{\mathrm{p}}} \tag{6}$$

and c^{p} is the signal propagation speed in the medium. It can be seen that t_k is on the both sides of the constraint equation (5), since $x(t_k)$ and $y(t_k)$ are functions of t_k. It is difficult to obtain an explicit express of t_k. This leads to use a Gauss-Helmert (GH) state transition model, which describes an implicit constraint systemically [11][12]. The GH model is given by

$$\mathbf{g}[\mathbf{x}(t_k), \mathbf{x}(t_{k-1})] + \Gamma\, \mathbf{v}(t_{k-1}) = \mathbf{0} \tag{7}$$

The GH models for the CV motion[2] and CT motion are given next.

1) Constant velocity Gauss-Helmert model: The GH model for CV motion is given by

$$\mathbf{g}^{\mathrm{CV}}[\mathbf{x}^{\mathrm{CV}}(t_k), \mathbf{x}^{\mathrm{CV}}(t_{k-1})] + \Gamma^{\mathrm{CV}} \mathbf{v}^{\mathrm{CV}}(t_{k-1}) = \mathbf{0}_5 \tag{8}$$

where $\mathbf{0}_5$ is a column vector with 5 elements, $\mathbf{g}^{\mathrm{CV}}[\cdot]$ is the implicit GH state transition function, which combines the CV motion constraints and the delay constraint between $\mathbf{x}(t_k)$ and $\mathbf{x}(t_{k-1})$. It is given by

$$\mathbf{g}^{\mathrm{CV}}(\cdot) = [g_1^{\mathrm{CV}}(\cdot)\ \ g_2^{\mathrm{CV}}(\cdot)\ \ g_3^{\mathrm{CV}}(\cdot)\ \ g_4^{\mathrm{CV}}(\cdot)\ \ g_5^{\mathrm{CV}}(\cdot)]' \tag{9}$$

where

$$g_1^{\mathrm{CV}}(\cdot) = x(t_k) - [x(t_{k-1}) + \dot{x}(t_{k-1})\Delta_k] \tag{10}$$

$$g_2^{\mathrm{CV}}(\cdot) = y(t_k) - [y(t_{k-1}) + \dot{y}(t_{k-1})\Delta_k] \tag{11}$$

$$g_3^{\mathrm{CV}}(\cdot) = \dot{x}(t_k) - \dot{x}(t_{k-1}) \tag{12}$$

$$g_4^{\mathrm{CV}}(\cdot) = \dot{y}(t_k) - \dot{y}(t_{k-1}) \tag{13}$$

$$g_5^{\mathrm{CV}}(\cdot) = t_k - (t_k^{\mathrm{R}} - \tau_k^{\mathrm{R}}) \tag{14}$$

with τ_k^{R} given in (6) and

$$\Delta_k = t_k - t_{k-1} \tag{15}$$

Based on the discrete white noise acceleration (WNA) model [2], the gain matrix Γ^{CV} and the zero-mean white Gaussian process noise $\mathbf{v}^{\mathrm{CV}}$ in (8) compensate for small accelerations and the uncertainty of the sound speed. The noise gain matrix Γ^{CV} is given by

$$\Gamma^{\mathrm{CV}} = \begin{bmatrix} \frac{1}{2}(\Delta_k)^2 & 0 & 0 \\ 0 & \frac{1}{2}(\Delta_k)^2 & 0 \\ \Delta_k & 0 & 0 \\ 0 & \Delta_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{16}$$

The covariance of $\mathbf{v}^{\mathrm{CV}}$ is

$$\mathbf{q}^{\mathrm{CV}} = \mathrm{diag}(\sigma_{\dot{x}}^2 \quad \sigma_{\dot{y}}^2 \quad \sigma_t^2) \tag{17}$$

where $\sigma_{\dot{x}}^2$ and $\sigma_{\dot{y}}^2$ are the variances on small target accelerations in the x and y coordinates respectively, and σ_t^2 is the process noise variance on the target time. The covariance of the error in the model (8) is given by

$$\mathbf{Q}^{\mathrm{CV}}(\Delta_k) = \Gamma^{\mathrm{CV}} \mathbf{q}^{\mathrm{CV}} (\Gamma^{\mathrm{CV}})' \tag{18}$$

2) Coordinated Turn Gauss-Helmert model: The GH state transition model for the CT motion is given by

$$\mathbf{g}^{\mathrm{CT}}[\mathbf{x}^{\mathrm{CT}}(t_k), \mathbf{x}^{\mathrm{CT}}(t_{k-1})] + \Gamma^{\mathrm{CT}} \mathbf{v}^{\mathrm{CT}}(t_{k-1}) = \mathbf{0}_6 \tag{19}$$

where

$$\mathbf{g}^{\mathrm{CT}}(\cdot) = [g_1^{\mathrm{CT}}(\cdot)\ \ g_2^{\mathrm{CT}}(\cdot)\ \ g_3^{\mathrm{CT}}(\cdot)\ \ g_4^{\mathrm{CT}}(\cdot)\ \ g_5^{\mathrm{CT}}(\cdot)\ \ g_6^{\mathrm{CT}}(\cdot)]' \tag{20}$$

with

$$g_1^{\mathrm{CT}}(\cdot) = x(t_k) - \left[x(t_{k-1}) + \frac{\sin[\omega(t_{k-1})\Delta_k]}{\omega(t_{k-1})}\dot{x}(t_{k-1}) - \frac{1 - \cos[\omega(t_{k-1})\Delta_k]}{\omega(t_{k-1})}\dot{y}(t_{k-1})\right] \tag{21}$$

$$g_2^{\mathrm{CT}}(\cdot) = y(t_k) - \left[y(t_{k-1}) + \frac{\sin[\omega(t_{k-1})\Delta_k]}{\omega(t_{k-1})}\dot{y}(t_{k-1}) + \frac{1 - \cos[\omega(t_{k-1})\Delta_k]}{\omega(t_{k-1})}\dot{x}(t_{k-1})\right] \tag{22}$$

$$g_3^{\mathrm{CT}}(\cdot) = \dot{x}(t_k) - \{\cos[\omega(t_{k-1})\Delta_k]\dot{x}(t_{k-1}) - \sin[\omega(t_{k-1})\Delta_k]\dot{y}(t_{k-1})\} \tag{23}$$

$$g_4^{\mathrm{CT}}(\cdot) = \dot{y}(t_k) - \{\sin[\omega(t_{k-1})\Delta_k]\dot{x}(t_{k-1}) + \cos[\omega(t_{k-1})\Delta_k]\dot{y}(t_{k-1})\} \tag{24}$$

$$g_5^{\mathrm{CT}}(\cdot) = \omega(t_k) - \omega(t_{k-1}) \tag{25}$$

$$g_6^{\mathrm{CT}}(\cdot) = t_k - (t_k^{\mathrm{R}} - \tau_k^{\mathrm{R}}) \tag{26}$$

The noise gain matrix Γ^{CT} is given by

$$\Gamma^{\mathrm{CT}} = \begin{bmatrix} \frac{1}{2}(\Delta_k)^2 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(\Delta_k)^2 & 0 & 0 \\ \Delta_k & 0 & 0 & 0 \\ 0 & \Delta_k & 0 & 0 \\ 0 & 0 & \Delta_k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{27}$$

[2]Although an explicit state transition model for the CV motion can be obtained through solving a 2nd order polynomial equation [6], the GH model is a systematical way which is suitable for both CV and CT motions.

$$\mathbf{q}^{\mathrm{CT}} = \mathrm{diag}(\sigma_{\ddot{x}}^2 \quad \sigma_{\ddot{y}}^2 \quad \sigma_{\omega}^2 \quad \sigma_{t}^2) \tag{28}$$

where σ_{ω}^2 is the variance of the Gaussian process noises of ω. The covariance of the error in (19) for the (nearly) CT motion, $\mathbf{Q}^{\mathrm{CT}}(\Delta_k)$, is computed by

$$\mathbf{Q}^{\mathrm{CT}}(\Delta_k) = \Gamma^{\mathrm{CT}}\mathbf{q}^{\mathrm{CT}}(\Gamma^{\mathrm{CT}})' \tag{29}$$

B. Measurement model

The measurement model relates the state at time t_k to the measurement at time t_k^{R}, which is given by

$$\mathbf{z}(t_k^{\mathrm{R}}) = \mathbf{h}[\mathbf{x}(t_k)] + \mathbf{w}(t_k^{\mathrm{R}}) \tag{30}$$

where $\mathbf{w}(t_k^{\mathrm{R}})$ is the measurement noise, and

$$\mathbf{h}(\cdot) = [h_1(\cdot) \;\; h_2(\cdot)]' \tag{31}$$

with

$$h_1(\cdot) = b(t_k^{\mathrm{R}}) = \tan^{-1}\left[\frac{x(t_k) - x^{\mathrm{R}}}{y(t_k) - y^{\mathrm{R}}}\right] \tag{32}$$

$$h_2(\cdot) = f^{\mathrm{R}}(t_k^{\mathrm{R}}) = f^{\mathrm{T}}(t_k^{\mathrm{T}})\left[1 - \frac{\dot{r}(t_k^{\mathrm{R}})}{c^{\mathrm{P}}}\right] \tag{33}$$

The challenge is how to obtain $\dot{r}(t_k^{\mathrm{R}})$ in (33). We know

$$\begin{aligned} r(t_k^{\mathrm{R}}) &= r_k^{\mathrm{T}} + r_k^{\mathrm{R}} \\ &= \sqrt{[x(t_k) - x^{\mathrm{T}}]^2 + [y(t_k) - y^{\mathrm{T}}]^2} \\ &\quad + \sqrt{[x(t_k) - x^{\mathrm{R}}]^2 + [y(t_k) - y^{\mathrm{R}}]^2} \end{aligned} \tag{34}$$

and

$$\begin{aligned} \dot{r}(t_k^{\mathrm{R}}) &= \frac{d[r(t_k^{\mathrm{R}})]}{d(t_k^{\mathrm{R}})} \\ &= \frac{\dot{x}(t_k)[x(t_k) - x^{\mathrm{T}}] + \dot{y}(t_k)[y(t_k) - y^{\mathrm{T}}]}{r_k^{\mathrm{T}}}\frac{dt_k}{d(t_k^{\mathrm{R}})} \\ &\quad + \frac{\dot{x}(t_k)[x(t_k) - x^{\mathrm{R}}] + \dot{y}(t_k)[y(t_k) - y^{\mathrm{R}}]}{r_k^{\mathrm{R}}}\frac{dt_k}{d(t_k^{\mathrm{R}})} \end{aligned} \tag{35}$$

When the signal propagation delay is negligible (for example, for a radar signal), one has $t_k = t_k^{\mathrm{R}}$ and

$$\frac{dt_k}{d(t_k^{\mathrm{R}})} = 1 \tag{36}$$

The analytical form of $\dot{r}(t_k^{\mathrm{R}})$ is then

$$\begin{aligned} \dot{r}(t_k^{\mathrm{R}}) &= \frac{\dot{x}(t_k)[x(t_k) - x^{\mathrm{T}}] + \dot{y}(t_k)[y(t_k) - y^{\mathrm{T}}]}{r_k^{\mathrm{T}}} \\ &\quad + \frac{\dot{x}(t_k)[x(t_k) - x^{\mathrm{R}}] + \dot{y}(t_k)[y(t_k) - y^{\mathrm{R}}]}{r_k^{\mathrm{R}}} \end{aligned} \tag{37}$$

However, the acoustic signal in our problem has significant propagation delay and $t_k \neq t_k^{\mathrm{R}}$. The analytical function

$$t_k = f(t_k^{\mathrm{R}}) \tag{38}$$

is impossible to obtain for a target in CT motion. This causes a major challenge for mapping the state to the measurement. An appropriate filter to cope with this challenge will be developed next.

III. INTERACTING MULTIPLE MODEL UNSCENTED GAUSS-HELMERT FILTER WITH NUMERICAL JACOBIAN

The IMM estimator [2] is the most commonly used hybrid approach to handle model uncertainty in target tracking. This section describes an IMM-UGHF-NJ filter with the implicit CV and CT models described in Section II and lack of analytical expression for the measurement function.

Similarly to the original IMM estimator, the IMM-UGHF-NJ performs the state estimation in four steps: mixing, mode-matched filtering, mode probabilities updating and final state combination:

1) In the mixing step, the m hypotheses (where m is the number of models in the filter) at time $k-1$ expand to m^2 hypotheses using the mixing probabilities based on the mode Markov chain, which is governed by the $m \times m$ mode probability transition matrix Π consisting of the mode transition probabilities, p_{ij}. The m^2 hypotheses are then merged into m hypotheses based on the mixture equations [2].
2) In the mode-matched filtering step, the mixed state estimates are updated by UGHF-NJs (given later) in parallel.
3) The mixing probabilities are obtained, and the updated mode probabilities are computed based on the innovations in the mode-matched UGHF-NJs. The updated mode probabilities together with the mode-conditioned estimated states and covariances are brought to the next step.
4) The final state estimate and its covariance for the current time cycle are computed based on the mixture equations using the latest mode probabilities in the combination step.

Since the states in the CV and CT models described in Section II have different dimensions, the unbiased mixing approach [16] is applied in the IMM filter to increase the CV state from 5 to 6. Before the mixing step, the CV state estimate and its error covariance are augmented with the turn rate information from the CT model.

The IMM-UGHF-NJ differs from the standard IMM in the mode-matched filters, which are UGHF-NJ. The UGHF-NJ handles the implicit GH state transition model and evaluates $f^{\mathrm{R}}(t_k^{\mathrm{R}})$ in the measurement vector (3) numerically. The UGHF-NJ prediction, state-to-measurement mapping and update steps are given in Algorithms 1–3, respectively. In these algorithms, the model superscripts "CV" and "CT" for the states and GH functions are omitted for simplicity.

ALGORITHM 1 *UGHF-NJ prediction*

Generate $(2n_{\mathbf{x}}+1)$ sigma points for $\hat{\mathbf{x}}(\hat{t}_{k-1})$:
$[\{\hat{\mathbf{x}}^i(\hat{t}^i_{k-1})\},\{w^i\}] = \text{SigPt}[\hat{\mathbf{x}}(\hat{t}_{k-1}),\mathbf{P}(\hat{t}_{k-1}),\kappa]$
Predict sigma points using Gauss-Newton algo.:
for all $\hat{\mathbf{x}}^i(\hat{t}^i_{k-1})$, $i \in \{1,\ldots,2n_{\mathbf{x}}+1\}$ **do**
$\mathbf{x}_0 = \hat{\mathbf{x}}^i(\hat{t}^i_{k-1})$
$\breve{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1}) = \text{GaussN}[\mathbf{g}(\mathbf{x}_1,\mathbf{x}_0)]$
end for
Regen sigma points with process noise:

$$\hat{\mathbf{x}}(\hat{t}_k \mid \hat{t}_{k-1}) = \sum_{i=1}^{2n_{\mathbf{x}}+1} w^i \breve{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1})$$

$$\mathbf{P}(\hat{t}_k \mid \hat{t}_{k-1}) = \sum_{i=1}^{2n_{\mathbf{x}}+1} w^i \tilde{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1})(\tilde{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1}))' + \mathbf{Q}(\Delta_k)$$

$$[\{\hat{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1})\},\{w^i\}] = \text{SigPt}[\hat{\mathbf{x}}(\hat{t}_k \mid \hat{t}_{k-1}),\mathbf{P}(\hat{t}_k \mid \hat{t}_{k-1}),\kappa]$$

where
$\tilde{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1}) = \breve{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1}) - \hat{\mathbf{x}}(\hat{t}_k \mid \hat{t}_{k-1})$
κ is a spread scalar of the sigma points.

Algorithm 1 predicts the state $\hat{\mathbf{x}}(\hat{t}_{k-1})$ from time $\hat{t}_{k-1}$ to an unknown target time, t_k, corresponding to the signal reception time t^{R}_k. The relationship between t_k and t^{R}_k is given by the implicit constraint (5). An unscented Gauss-Helmert approach is used for the state prediction with the implicit constraint. Firstly, $2n_{\mathbf{x}}+1$ sigma points of $\hat{\mathbf{x}}(\hat{t}_{k-1})$ are generated using SigPt$(\cdot)$ (given in the Appendix), where $n_{\mathbf{x}}$ is the dimension of the state vector. Secondly, each sigma point is predicted to $\hat{t}^i_k$ using the Gauss-Newton algorithm GaussN$(\cdot)$ (also given in the Appendix) based on the Gauss-Helmert function $\mathbf{g}(\mathbf{x}_1,\mathbf{x}_0)$, where i is the index of the sigma points. The $2n_{\mathbf{x}}+1$ GaussN$(\cdot)$ find $\mathbf{x}_1 = \breve{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1})$ from $\mathbf{x}_0 = \hat{\mathbf{x}}^i(\hat{t}^i_{k-1})$ iteratively. Thirdly, the predicted sigma points are re-generated with considering also the process noise (with the approprate larger prediction covariance).

ALGORITHM 2 *UGHF-NJ mapping the predicted state to measurement*

$[\{t^{\text{R},j}_k\},\{w^j\}] = \text{SigPt}[t^{\text{R}}_k,\sigma_{t^{\text{R}}_k},\kappa]$
for all $\hat{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1})$, $i \in \{1,\ldots,2n_{\mathbf{x}}+1\}$ **do**
$\mathbf{x}_0 = \hat{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1})$
for $j = 1:3$ **do**
$\hat{\mathbf{x}}^{i,j}(\hat{t}^j_k \mid \hat{t}^i_{k-1}) = \text{GaussN}[\mathbf{g}(\mathbf{x}_1,\mathbf{x}_0)|_{t^{\text{R}}_k = t^{\text{R},j}_k}]$
$\hat{r}^{i,j}(t^{\text{R}}_k) \leftarrow \hat{\mathbf{x}}^{i,j}(\hat{t}^j_k \mid \hat{t}^i_{k-1})$
end for
$\hat{\dot{r}}^i(t^{\text{R}}_k) = \text{NJ}[\{t^{\text{R},j}_k\},\{\hat{r}^{i,j}(t^{\text{R}}_k)\},\{w^j\}]$
$\hat{f}^{\text{R},i}(t^{\text{R}}_k) \leftarrow$ using (33)
$\hat{b}^i(t^{\text{R}}_k) \leftarrow$ using (32)
$\mathbf{z}^i(t^{\text{R}}_k) = [\hat{b}^i(t^{\text{R}}_k)\ \ \hat{f}^{\text{R},i}(t^{\text{R}}_k)]'$
end for

$$\hat{\mathbf{z}}(t^{\text{R}}_k) = \sum_{i=1}^{2n_{\mathbf{x}}+1} w^i \hat{\mathbf{z}}^i(t^{\text{R}}_k)$$

Algorithm 2 maps the predicted state to the measurement space. The challenge here is that we cannot obtain the Doppler shifted frequency $f^{\text{R}}(t^{\text{R}}_k)$ in the measurement from the predicted state directly. The range rate $\dot{r}(t^{\text{R}}_k)$ in (33) cannot be derived from the bistatic range $r(t^{\text{R}}_k)$, which has no analytical form in terms of t^{R}_k. We use a numerical approach, called numerical Jacobian (NJ), to obtain $\dot{r}(t^{\text{R}}_k)$ from $r(t^{\text{R}}_k)$. It is known that the slope of the tangent line is the derivative of a nonlinear function at a point of interest. The principle of the NJ$(\cdot)$ (given in the Appendix) is to find the best linear fit to a nonlinear function based on a few weighted points around the point of interest. If we can provide these weighted points around $[t^{\text{R}}_k, r(t^{\text{R}}_k)]$, its derivative $\dot{r}(t^{\text{R}}_k)$ can then be computed using NJ$(\cdot)$. Firstly, we generate the reception time set around t^{R}_k using SigPt$(\cdot)$, i.e.,

$$\{t^{\text{R},j}_k\} = \{t^{\text{R}}_k, t^{\text{R}}_k - \sigma_{t^{\text{R}}_k}, t^{\text{R}}_k + \sigma_{t^{\text{R}}_k}\} \quad j = 1,2,3 \tag{39}$$

where $\sigma_{t^{\text{R}}_k}$ is a very small shift from t^{R}_k. Its weight set is $\{w^j\}$. Secondly, we use GaussN$(\cdot)$ to obtain the predicted state set $\{\hat{\mathbf{x}}^{i,j}(\hat{t}_k \mid \hat{t}_{k-1})\}$ corresponding to the reception time set $\{t^{\text{R},j}_k\}$ for the ith sigma point of the predicted state (obtained from Algorithm 1). The bistatic range can then be computed using (34). The set of bistatic ranges corresponding to $\{t^{\text{R},j}_k\}$ for the ith sigma point of the predicted state is

$$\{\hat{r}^{i,j}(t^{\text{R}}_k)\} = \{\hat{r}^i(t^{\text{R}}_k), \hat{r}^i(t^{\text{R}}_k - \sigma_{t^{\text{R}}_k}), \hat{r}^i(t^{\text{R}}_k + \sigma_{t^{\text{R}}_k})\} \quad j = 1,2,3 \tag{40}$$

Thirdly, we use these two sets, $\{t^{\text{R},j}_k\}$ and $\{\hat{r}^{i,j}(t^{\text{R}}_k)\}$, which form three points around $[t^{\text{R}}_k, \hat{r}^i(t^{\text{R}}_k)]$ to evaluate the range rate $\hat{\dot{r}}^i(t^{\text{R}}_k)$ using NJ$(\cdot)$. Once $\hat{\dot{r}}^i(t^{\text{R}}_k)$ is obtained, $f^{\text{R},i}(t^{\text{R}}_k)$ can be computed using (33), and the predicted measurement $\mathbf{z}^i(t^{\text{R}}_k)$ follows.

ALGORITHM 3 *UGHF-NJ update*

$$\hat{\mathbf{x}}(\hat{t}_k) = \hat{\mathbf{x}}(\hat{t}_k \mid \hat{t}_{k-1}) + \mathbf{K}_k \boldsymbol{\nu}(t^{\text{R}}_k)$$
$$\mathbf{P}(\hat{t}_k) = \mathbf{P}(\hat{t}_k \mid \hat{t}_{k-1}) - \mathbf{K}_k \mathbf{S}(t^{\text{R}}_k)\mathbf{K}'_k$$

where
$\boldsymbol{\nu}(t^{\text{R}}_k) = \mathbf{z}(t^{\text{R}}_k) - \hat{\mathbf{z}}(t^{\text{R}}_k)$
$\mathbf{K}_k = \mathbf{P}_{xz}\mathbf{S}(t^{\text{R}}_k)^{-1}$
$\mathbf{S}(t^{\text{R}}_k) = \mathbf{R} + \mathbf{P}_{zz}$

$$\mathbf{P}_{xz} = \sum_{i=1}^{2n_{\mathbf{x}}+1} w^i \tilde{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1})\tilde{\mathbf{z}}^i(t^{\text{R}}_k)'$$

$$\mathbf{P}_{zz} = \sum_{i=1}^{2n_{\mathbf{x}}+1} w^i [\tilde{\mathbf{z}}^i(t^{\text{R}}_k)\tilde{\mathbf{z}}^i(t^{\text{R}}_k)']$$

$\tilde{\mathbf{z}}^i(t^{\text{R}}_k) = \hat{\mathbf{z}}^i(t^{\text{R}}_k) - \hat{\mathbf{z}}(t^{\text{R}}_k)$
$\tilde{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1}) = \hat{\mathbf{x}}^i(\hat{t}^i_k \mid \hat{t}^i_{k-1}) - \hat{\mathbf{x}}(\hat{t}_k \mid \hat{t}_{k-1})$

Algorithm 3 updates the predicted state based on the measurement $\mathbf{z}(t^{\text{R}}_k)$. This step is the same as in the conventional UKF.

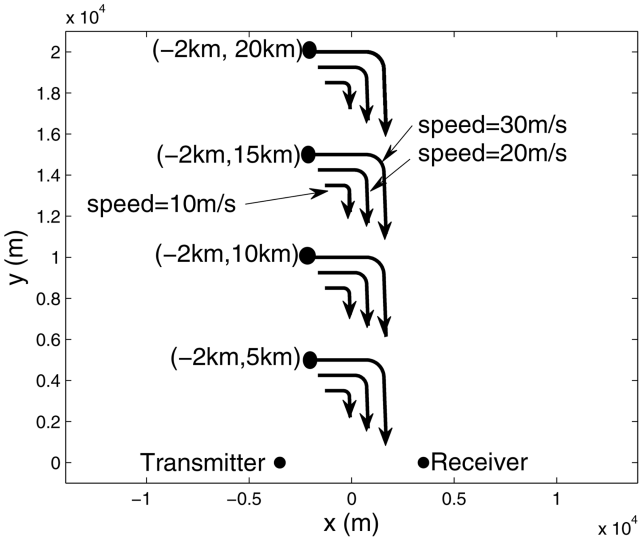


Fig. 3. Test scenarios.

IV. SIMULATION RESULTS

The IMM-UGHF-NJ is tested with simulated data in this section. The simulated scenarios are shown in Fig. 3. Twelve targets move in CV-CT-CV motion with different speeds and ranges. They are categorised into four groups based on the ranges (or distances) to the transmitter and receiver, which are between 0–5 km, 5–10 km, 10–15 km and 15–20 km. Each category has three targets with speeds 10 m/s, 20 m/s and 30 m/s, respectively. All targets have two CV legs linked by a CT arc. The durations of the first CV, CT and the second CV are 90 s, 45 s and 90 s, respectively. The CT arc is a 90° right turn with turn rate 2°/s. The transmitter and receiver are located at $(-3500\text{ m}, 0\text{ m})$ and $(3500\text{ m}, 0\text{ m})$, respectively. The transmitter emits a CW signal with frequency 1000 Hz. The sampling interval of the receiver is $T = 1$ s. The measurement errors of bearing and Doppler shifted frequency at receiver are assumed Gaussians with standard deviations $\sigma_b = 1°$ and $\sigma_f = 0.25$ Hz, respectively. The sound propagation speed in water is $c^{\mathrm{p}} = 1484$ m/s.

The following two algorithms are used in testing:

- IMM-UKF: The mode-matched filters are UKF. They estimate target position and velocity only. The propagation delay is not taken into consideration at all. The Doppler shifted frequency in the measurement model is based on (37) which is commonly used in multistatic radar tracking system. The target times are taken as the signal reception times by the receiver.
- IMM-UGHF-NJ: This is the new algorithm proposed in this paper. The propagation delay is taken into consideration in the state estimation, and the target times attached to the target trajectory are estimated from multiple UGHF-NJs.

One CV model and two CT models (CT-L and CT-H) are used in both IMM estimators. The CT-L and CT-H have low and high turn rate process noises, respectively. This setup can provide a fast turn rate adaptation during model switching [4]. The initial mode probabilities for the three models are 1/3. The probability transition matrix Π^3 is

$$\Pi^3 = \begin{bmatrix} 0.950 & 0.025 & 0.025 \\ 0.025 & 0.950 & 0.025 \\ 0.025 & 0.025 & 0.950 \end{bmatrix} \tag{41}$$

The measurement error covariance $\mathbf{R}$ is

$$\mathbf{R} = \mathrm{diag}[(1°)^2 \ \ (0.25\text{ Hz})^2] \tag{42}$$

In the IMM-UGHF-NJ, the process noise covariances $\mathbf{q}^{\mathrm{CV}}$, $\mathbf{q}^{\mathrm{CT\text{-}L}}$ and $\mathbf{q}^{\mathrm{CT\text{-}H}}$ are, respectively,

$$\mathbf{q}^{\mathrm{CV}} = \mathrm{diag}[(0.1\text{ m/s}^2)^2 \ \ (0.1\text{ m/s}^2)^2 \ \ (0.1\text{s})^2] \tag{43}$$

$$\mathbf{q}^{\mathrm{CT\text{-}L}} = \mathrm{diag}[(0.1\text{ m/s}^2)^2 \ \ (0.1\text{ m/s}^2)^2 \ \ (0.1°/\text{s})^2 \ \ (0.1\text{s})^2] \tag{44}$$

$$\mathbf{q}^{\mathrm{CT\text{-}H}} = \mathrm{diag}[(0.1\text{ m/s}^2)^2 \ \ (0.1\text{ m/s}^2)^2 \ \ (1°/\text{s})^2 \ \ (0.1\text{s})^2] \tag{45}$$

and κ is set to 1 in all SigPt$(\cdot)$ (see the Appendix), and $\sigma_{t_k^{\mathrm{R}}}$ is set to 0.1s in Algorithm 2. The initial state estimates are

$$\hat{\mathbf{x}}^{\mathrm{CV}}(t_0) = [\hat{r}_0 \sin b_0 \ \ \hat{r}_0 \cos b_0 \ \ \hat{\dot{x}}_0 \ \ \hat{\dot{y}}_0 \ \ \hat{t}_0]' \tag{46}$$

$$\hat{\mathbf{x}}^{\mathrm{CT\text{-}L}}(t_0) = [\hat{r}_0 \sin b_0 \ \ \hat{r}_0 \cos b_0 \ \ \hat{\dot{x}}_0 \ \ \hat{\dot{y}}_0 \ \ 0.1°/s \ \ \hat{t}_0]' \tag{47}$$

$$\hat{\mathbf{x}}^{\mathrm{CT\text{-}H}}(t_0) = \hat{\mathbf{x}}^{\mathrm{CT\text{-}L}}(t_0) \tag{48}$$

where

$$\hat{r}_0 \sim \mathcal{N}(r_0^{\mathrm{R}}, \sigma_r^2) \tag{49}$$

$$b_0 = b(t_0^{\mathrm{R}}) \tag{50}$$

$$\hat{\dot{x}}_0 \sim \mathcal{N}(\dot{x}_0, \sigma_{\dot{x}}^2) \tag{51}$$

$$\hat{\dot{y}}_0 \sim \mathcal{N}(\dot{y}_0, \sigma_{\dot{y}}^2) \tag{52}$$

$$\hat{t}_0 = t_0^{\mathrm{R}} - \hat{r}_0 / c^{\mathrm{p}} \tag{53}$$

with r_0^{R} the true value of the range from the target at time t_0 to the receiver at time t_0^{R}, $\sigma_r = 400$ m, and $b(t_0^{\mathrm{R}})$ is the measured bearing at time t_0^{R}, $\dot{x}_0$ and $\dot{y}_0$ are the true target velocities, and $\sigma_{\dot{x}} = \sigma_{\dot{y}} = 4$ m/s. The initial state error covariances for the three models are

$$\mathbf{P}^{\mathrm{CV}}(t_0) = \begin{bmatrix} P_{xx} & P_{xy} & 0 & 0 & 0 \\ P_{yx} & P_{yy} & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x}}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}}^2 & 0 \\ 0 & 0 & 0 & 0 & (\sigma_r / c^{\mathrm{p}})^2 \end{bmatrix} \tag{54}$$

$$\mathbf{P}^{\text{CT-L}}(t_0) = \mathbf{P}^{\text{CT-H}}(t_0)$$

$$= \begin{bmatrix} P_{xx} & P_{xy} & 0 & 0 & 0 & 0 \\ P_{yx} & P_{yy} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (0.02^\circ/s)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (\sigma_r/c^{\text{P}})^2 \end{bmatrix} \tag{55}$$

where

$$P_{xx} = (\hat{r}_0 \sigma_b \cos b_0)^2 + (\sigma_r \sin b_0)^2 \tag{56}$$

$$P_{yy} = (\hat{r}_0 \sigma_b \sin b_0)^2 + (\sigma_r \cos b_0)^2 \tag{57}$$

$$P_{xy} = P_{yx} = (\sigma_r^2 - \hat{r}_0^2 \sigma_b^2) \sin b_0 \cos b_0 \tag{58}$$

The parameters in the IMM-UKF, including the process noise covariances, initial states and initial state error covariances are the same as the IMM-UGHF-NJ, but the elements corresponding to the target time are removed.

The simulation results present the root mean square errors (RMSE) of the estimated target positions and speeds obtained from 100 Monte Carlo runs. The estimated position and speed errors at time $\hat{t}_k$ are computed by

$$\text{pos}^{\text{err}}(\hat{t}_k) = \sqrt{[\hat{x}(\hat{t}_k) - x(\hat{t}_k)]^2 + [\hat{y}(\hat{t}_k) - y(\hat{t}_k)]^2} \tag{59}$$

$$\text{sp}^{\text{err}}(\hat{t}_k) = \sqrt{[\hat{\dot{x}}(\hat{t}_k) - \dot{x}(\hat{t}_k)]^2 + [\hat{\dot{y}}(\hat{t}_k) - \dot{y}(\hat{t}_k)]^2} \tag{60}$$

where $\hat{x}(\hat{t}_k)$, $\hat{y}(\hat{t}_k)$, $\hat{\dot{x}}(\hat{t}_k)$ and $\hat{\dot{y}}(\hat{t}_k)$ are the estimated target positions and velocities in the x and y coordinates respectively, $x(\hat{t}_k)$, $y(\hat{t}_k)$, $\dot{x}(\hat{t}_k)$ and $\dot{y}(\hat{t}_k)$ are the true target positions and velocities in the x and y coordinates respectively, and $\hat{t}_k$ is the estimated target time in estimation cycle k.

Tables I and II show the averages of position and speed RMSE for the two algorithms for the twelve simulated targets from the four categories displayed in Fig. 3. Figs. 4–7 show the position RMSE versus time of the two algorithms for four simulated targets, one from each category, respectively. They are the targets in the range between 0–5 km with speed 30 m/s, range between 5–10 km with speed 10 m/s, range between 10–15 km with speed 20 m/s and range between 15–20 km with speed 30 m/s. It can be seen that the IMM-UGHF-NJ outperforms the IMM-UKF for all targets. The accuracy improvement is target range and speed dependent. A faster and longer range target has more improvement than a slower one at a shorter range. This is because that estimation error of the IMM-UKF depends on the target speed and propagation delay τ_k^{R} (details can be found in Section V-C of [12]). The range from the target to the receiver is proportional to the propagation delay. From the results we can say that the estimation error without considering propagation delay is significant, especially for a long range target or a fast target (such as a speed boat or torpedo).

TABLE I
Averages of position RMSE

Target Range (km)	Target Speed (m/s)	IMM-UKF (m)	IMM-UGHF-NJ (m)	Improv. (m)
0–5	10	412.6	411.9	0.7
	20	397.9	375.4	22.5
	30	319.3	281.8	37.5
5–10	10	406.5	400.8	5.7
	20	387.4	336.0	51.4
	30	438.4	325.2	113.2
10–15	10	413.1	401.5	11.6
	20	489.9	428.2	61.7
	30	523.0	369.7	153.3
15–20	10	436.8	412.7	24.1
	20	481.0	407.5	73.5
	30	614.2	399.5	214.7

TABLE II
Averages of speed RMSE

Target Range (km)	Target Speed (m/s)	IMM-UKF (m/s)	IMM-UGHF-NJ (m/s)	Improv. (m/s)
0–5	10	1.9	1.8	0.1
	20	2.4	2.0	0.4
	30	2.9	2.4	0.5
5–10	10	2.0	1.8	0.2
	20	2.7	1.9	0.8
	30	3.8	2.1	1.7
10–15	10	2.4	1.9	0.5
	20	3.5	1.9	1.6
	30	5.0	2.2	2.8
15–20	10	3.0	2.4	0.6
	20	4.3	2.0	2.3
	30	6.2	2.1	4.2

The maneuvering mode probabilities of the two IMM filters are also investigated. Figs. 8–11 show the sum of the mode probabilities of the two CT models (which represents the target maneuvering probability) versus time for the four targets, respectively. It can be seen that the maneuvering probability for both filters increases when the target is maneuvering. The IMM-UGHF-NJ reacts faster than the IMM-UKF. A delay in the model switching for a long range target (> 5 km) is observed. However, the mode probability does not match the ground truth very well when the target is in CV motion. This is because the turn rate ω in CT models can adapt to a small value when the target is in CV motion.

To evaluate the consistency of the IMM-UGHF-NJ and IMM-UKF, the average normalized estimation error squared (NEES) is evaluated. The average (2D) position

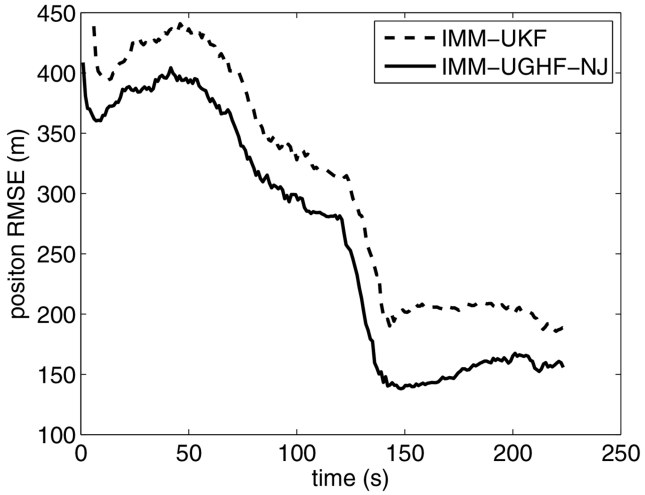


Fig. 4. Position estimate RMSE versus time for the target with speed = 30 m/s and range less than 5 km.

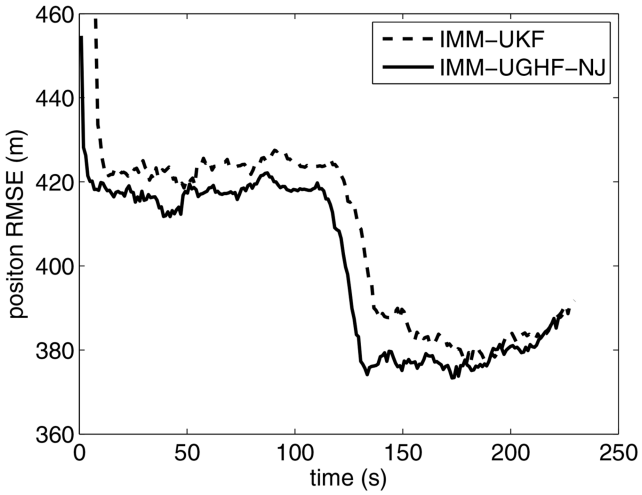


Fig. 5. Position estimate RMSE versus time for the target with speed = 10 m/s and range 5–10 km.

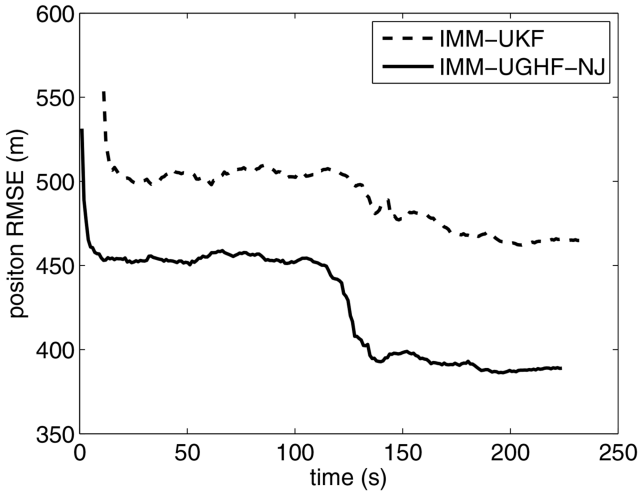


Fig. 6. Position estimate RMSE versus time for the target with speed = 20 m/s and range 10–15 km.

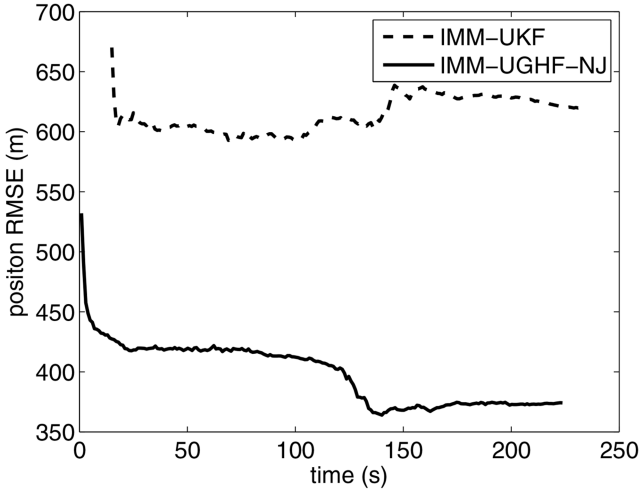


Fig. 7. Position estimate RMSE versus time for the target with speed = 30 m/s and range 15–20 km.

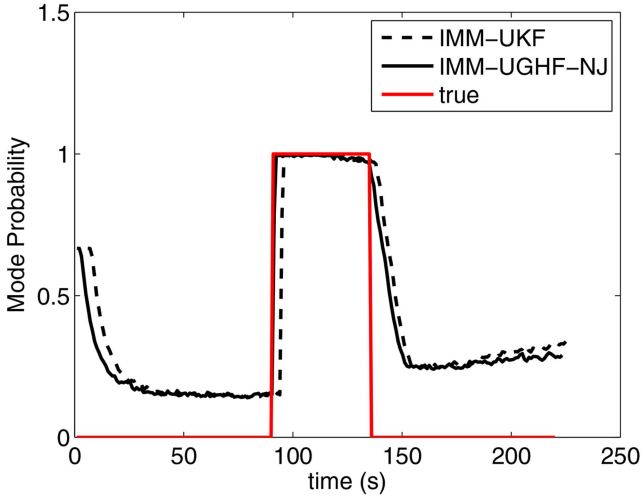


Fig. 8. Maneuvering probability versus time for target with speed = 30 m/s and range less than 5 km.

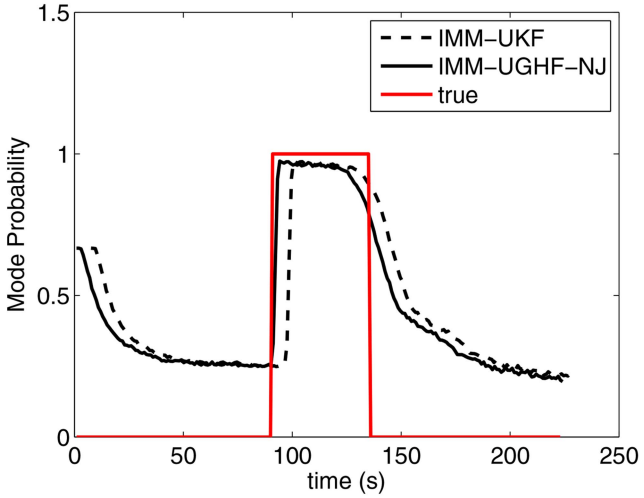


Fig. 9. Maneuvering probability versus time for target with speed = 10 m/s and range 5–10 km.

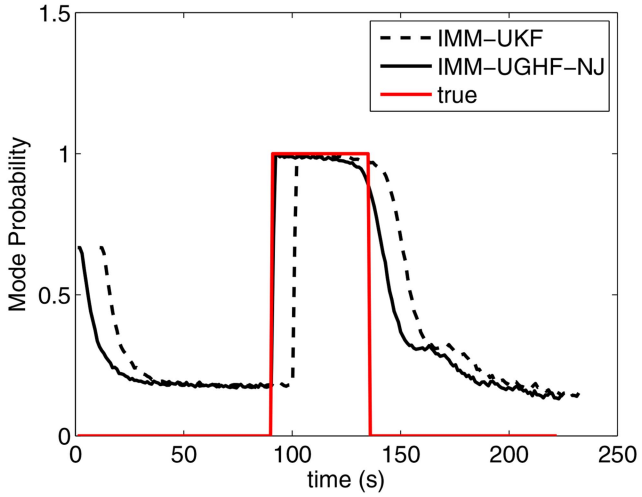


Fig. 10. Maneuvering probability versus time for target with speed = 20 m/s and range 10–15 km.

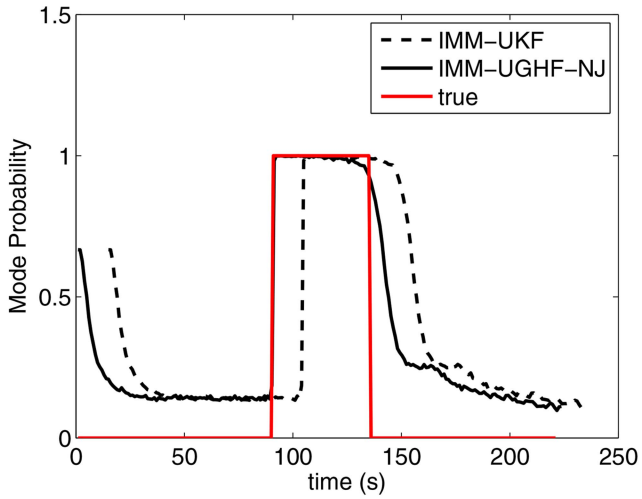


Fig. 11. Maneuvering probability versus time for target with speed = 30 m/s and range 15–20 km.

NEES at time $\hat{t}_k$ for N Monte Carlo runs is [1]

$$\bar{\epsilon}(\hat{t}_k) = \frac{1}{2N}\sum_{i=1}^{N} \tilde{\mathbf{x}}^i_{1:2}(\hat{t}_k)'[\mathbf{P}^i_{1:2,1:2}(\hat{t}_k)]^{-1}\tilde{\mathbf{x}}^i_{1:2}(\hat{t}_k) \qquad (61)$$

where i the run index, $\mathbf{P}^i_{1:2,1:2}(\hat{t}_k)$ is the position estimate error covariance submatrix at the estimated target time $\hat{t}_k$, and

$$\tilde{\mathbf{x}}_{1:2}(\hat{t}_k) = \hat{\mathbf{x}}_{1:2}(\hat{t}_k) - \mathbf{x}_{1:2}(\hat{t}_k) \qquad (62)$$

The two-sided 95% probability region for a 200 degrees of freedom ($N = 100$, dimension of $\mathbf{x}_{1:2} = 2$) chi-square random variable is [162,241.2]. Dividing by 200, the average NEES interval is [0.81,1.21].

Fig. 12 shows the average position NEES versus time of the IMM-UGHF-NJ for the four targets with expected value 1. It can be seen that most of the position NEES are within the interval [0.81,1.21]. There are two exception cases out of the interval. One is at the model switching times which are around 90 s and 135 s.

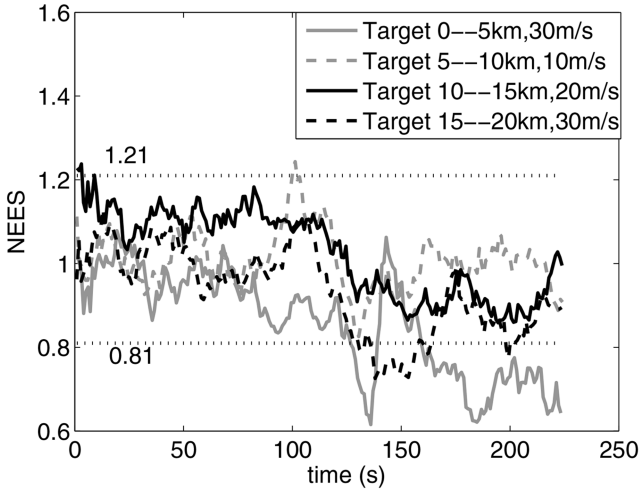


Fig. 12. Four targets position NEES versus time for the IMM-UGHF-NJ.

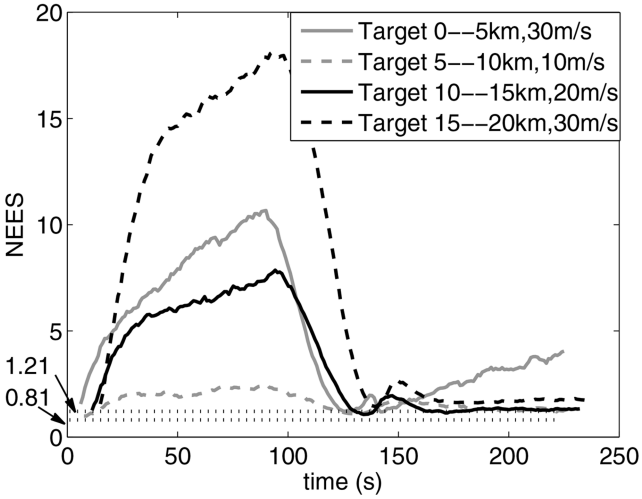


Fig. 13. Four target position NEES versus time for the IMM-UKF.

Another one is at the ending part of the near range target (0–5 km, 30 m/s). When the target is switching between the CV and CT motions, the IMM-UGHF-NJ cannot adapt to the correct model immediately, and this causes short delay in the maneuver start and maneuver end, but these delays are shorter than for the IMM-UKF. For the near range target (0–5 km, 30 m/s), the NEES is below the lower bound 0.81 at the ending part ($t > 160$ s). We can observe from Fig. 8 that the maneuvering probability is greater than 0.24 when $t > 160$ s. It is apparently worse than for the other three targets shown in Figs. 9–11. This is caused by the marginal observability of the CV motion model from the measurements, and results in the maneuvering probability (sum of the probabilities of CT models) not small enough. The error covariance of the combined estimate is too large (pessimistic) when the contribution of the incorrect models (the maneuvering models with probability around 0.25) cannot be overlooked. The small NEES is therefore caused by this large error

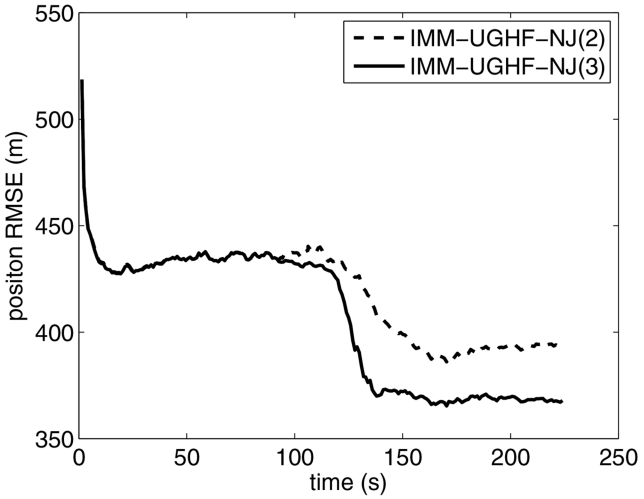


Fig. 14. Position estimate RMSE versus time of the IMM-UGHF-NJ using two-model and three-model for the target with speed = 20 m/s and range 10–15 km.

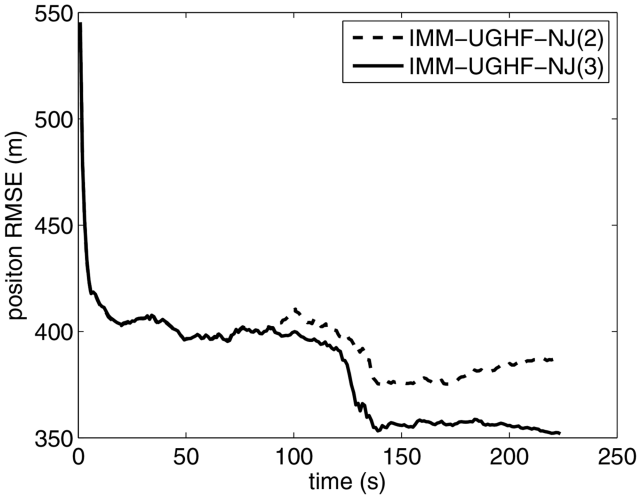


Fig. 15. Position estimate RMSE versus time of the IMM-UGHF-NJ using two-model and three-model for the target with speed = 30 m/s and range 15–20 km.

covariance.

Fig. 13 shows the NEES of the IMM-UKF for the same four targets. All of them are above the upper bound 1.21. Obviously, the IMM-UKF provides biased estimation without considering propagation delay.

We also compare the results of using three models and two models in the IMM-UGHF-NJ. The models and parameters in the three-model configuration have been defined before. The two-model IMM-UGHF-NJ uses one CV model and one CT model. Their initial mode probabilities are 1/2, and the probability transition matrix Π^2 is

$$\Pi^2 = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \tag{63}$$

The process noises the initial states in the two-model estimator are set as the same for the CV and CT in the three-model case, except the process noise variance on turn rate is set as $(0.5^\circ/\text{s})^2$ (intermediate value between those in the CT-L and CT-H models). Figs. 14 and 15 show the position estimate RMSE versus time of using two models and three models for the two targets (10–15 km, 20 m/s and 15–20 km, 30 m/s). It can be seen that there is no difference in the first leg ($t < 90$ s) between two-model and three-model IMM-UGHF-NJs. Once the targets start maneuvering, the three-model IMM-UGHF-NJ outperforms the two-model version. This is due to the model CT-H with high process noise on the turn rate. It allows the turn rate to adapt to the correct value quickly during model switchings. Meanwhile, the CT-L model with slow change in turn rate can balance the CT-H after model switching.

V. CONCLUSIONS

This paper developed the IMM-UGHF-NJ filter to track maneuvering targets using bistatic CW-CAS in the presence of propagation delay. The IMM-UGHF-NJ can overcome the two challenges of this tracking problem, namely, the implicit state transition model and absence of analytical expression of the Doppler shifted frequency in the measurement model. Simulation tests were conducted on targets with different ranges and speeds. Results show that the IMM-UGHF-NJ outperforms the IMM-UKF which does not take the propagation delay into consideration. It is also found that the estimate accuracy improvement of the IMM-UGHF-NJ over the IMM-UKF is more significant for a longer range or a higher speed target. Such a target (for example a speed boat or a torpedo) needs an appropriate filter (IMM-UGHF-NJ) to handle the propagation delay. A statistical study of the results was also conducted through the NEES. The results show that the IMM-UGHF-NJ is a consistent filter in most of the cases, except the situations when the target motion uncertainty cannot be well observed from measurements. The NEES results of the IMM-UKF are far above the upper bound because of its biased estimation due to ignoring the propagation delay.

Although the IMM-UGHF-NJ is developed based on the stationary transmitter and receiver, it can be applied to a moving transmitter and receiver if their positions are known accurately. Further study will be conducted to cope with inaccurate transmitter and receiver positions.

APPENDIX

The three algorithms SigPt(·), GaussN(·) and NJ(·) used in IMM-UGHF-NJ are given next.

a) SigPt(·) generates the sigma points for a random variable $\mathbf{x}$ with covariance $\mathbf{P}_\mathbf{x}$ [7].

$$[\mathbf{x}^i, w^i] = \text{SigPt}(\mathbf{x}, \mathbf{P}_\mathbf{x}, \kappa) \quad i = 1, \ldots, 2n_\mathbf{x} + 1 \tag{64}$$

where

$$\mathbf{x}^1 = \mathbf{x} \tag{65}$$

$$\mathbf{x}^i = \mathbf{x} + \left[\sqrt{(n_\mathbf{x}+\kappa)\mathbf{P}_\mathbf{x}}\right]_{i-1} \quad i = 2,\ldots,n_\mathbf{x}+1 \tag{66}$$

$$\mathbf{x}^i = \mathbf{x} - \left[\sqrt{(n_\mathbf{x}+\kappa)\mathbf{P}_\mathbf{x}}\right]_{i-n_\mathbf{x}-1} \quad i = n_\mathbf{x}+2,\ldots,2n_\mathbf{x}+1 \tag{67}$$

$$w_0 = \frac{\kappa}{n_\mathbf{x}+\kappa} \quad i = 1 \tag{68}$$

$$w_i = \frac{1}{2(n_\mathbf{x}+\kappa)} \quad i = 2,\ldots,2n_\mathbf{x}+1 \tag{69}$$

where $n_\mathbf{x}$ is the dimension of $\mathbf{x}$, $\left[\sqrt{(n_\mathbf{x}+\kappa)\mathbf{P}_\mathbf{x}}\right]_{i^*}$ indicates the i^*th column of the matrix $[\cdot]$, and κ is a scalar that determines the spread of sigma points.

b) GaussN$(\cdot)$ is a Gauss-Newton algorithm to obtain the solution of an implicit equation $\mathbf{g}(\cdot) = \mathbf{0}$ iteratively [11][12] and yields

$$\hat{\mathbf{x}}_1 = \text{GaussN}[\mathbf{g}(\mathbf{x}_1, \mathbf{x}_0)] \tag{70}$$

where $\mathbf{x}_0$ is known. The iteration procedure is

$$\hat{\mathbf{x}}_1^{j+1} = \hat{\mathbf{x}}_1^j - (\mathbf{A}^j)^{-1}\mathbf{g}(\hat{\mathbf{x}}_1^j, \mathbf{x}_0) \tag{71}$$

where j is the iteration index, $\mathbf{A}^j$ is the Jacobian matrix defined by

$$\mathbf{A}^j = \frac{\partial \mathbf{g}[(\hat{\mathbf{x}}_1^j, \mathbf{x}_0)]}{\partial \hat{\mathbf{x}}_1^j} \tag{72}$$

c) NJ$(\cdot)$ calculates the Jacobian (or derivative) $\mathbf{H}$ of a function

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) \tag{73}$$

at a point of interest $\mathbf{x}_0$ numerically [8][13][9]. There is no analytical form for $\mathbf{h}(\cdot)$, but $\mathbf{z}$ can be obtained through numerical method from a given $\mathbf{x}$. The Jacobian is

$$\mathbf{H} = \text{NJ}[\{\mathbf{x}^i\}, \{\mathbf{z}^i\}, \{w^i\}] \tag{74}$$

where $\{\mathbf{x}^i\}$ is the sigma point set around $\mathbf{x}_0$ generated from a very small covariance, $\{\mathbf{z}^i\}$ is its corresponding set after transformation and $\{w^i\}$ is the set of weights. The NJ is implementing through the following steps:

1) Form the sigma point set

$$\bar{\mathbf{X}} = \begin{bmatrix} \mathbf{x}^1 & \mathbf{x}^2 & \cdots & \mathbf{x}^{2n_\mathbf{x}+1} \\ 1 & 1 & \cdots & 1 \end{bmatrix} - \begin{bmatrix} \mathbf{x}^1 \\ 0 \end{bmatrix} \tag{75}$$

$$\mathbf{Z} = \begin{bmatrix} \bar{\mathbf{z}}^1 \\ \bar{\mathbf{z}}^2 \\ \vdots \\ \bar{\mathbf{z}}^l \end{bmatrix} = [\mathbf{z}^1 \quad \mathbf{z}^2 \quad \cdots \quad \mathbf{z}^{2n_\mathbf{x}+1}] \tag{76}$$

where $\mathbf{x}^1 = \mathbf{x}_0$, and l is the dimension of $\mathbf{z}$.

2) Estimate $\mathbf{H}$ using the weighted least squares (WLS) algorithm

$$\mathbf{a}^j = (\bar{\mathbf{X}}\mathbf{W}\bar{\mathbf{X}}')^{-1}\bar{\mathbf{X}}\mathbf{W}(\bar{\mathbf{z}}^j)' \tag{77}$$

$$\hat{\bar{\mathbf{H}}} = [\mathbf{a}^1 \quad \mathbf{a}^2 \quad \cdots \quad \mathbf{a}^l]' \tag{78}$$

$$\hat{\mathbf{H}} = \hat{\bar{\mathbf{H}}}(1:l, 1:n_\mathbf{x}) \tag{79}$$

where $\mathbf{W} = \text{diag}(\{w^i\})$, $j \in \{1,\ldots,l\}$, and $\hat{\mathbf{H}}$ is $\hat{\bar{\mathbf{H}}}$ without the last column.

REFERENCES

[1] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T.
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software,
New York: Wiley, 2001.

[2] Bar-Shalom, Y., Willett, P. K., and Tian, X.
Tracking and Data Fusion: A Handbook of Algorithms,
YBS Publishing, 2011.

[3] DeFerrari, H. A.
The Application of m-Sequences to Bi-static Active Sonar.
Journal of the Acoustical Society of America, 114(4):2399–2400, 2003.

[4] Huang, H. A. J., Bar-Shalom, Y., Yang, R., and Ng, G. W.
Tracking a maneuvering target using two heterogeneous passive sensors on a single stationary platform with IMM estimation.
accepted by Journal of Advances in Information Fusion, Dec. 2016.

[5] Grimmett, D., Wakayama, C.
Multistatic Tracking for Continous Active Sonar using Doppler-Bearing Measurements.
Proc. 16th International Conference on Information Fusion, Istanbul, Turkey, Jul. 2013.

[6] Jauffret, C., Pérez, A.-C., Blanc-Benon, P., and Tanguy, H.
Doppler-only Target Motion Analysis in a High Duty Cycle Sonar System.
Proc. 19th International Conference on Information Fusion, Heidelburg, Germany, Jul. 2016.

[7] Julier, S. J., and Uhlmann, J. K.
A new extension of the Kalman filter to nonlinear systems.
Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls, Apr. 1997.

[8] Xiong, Y. B., Zhong, X. H., and Yang, R.
The linear fitting Kalman filter for nonlinear tracking.
Proceedings of the 5th Asia-Pacific Conference on Synthetic Aperture Radar, Singapore, Sep. 2015.

[9] Xiong, Y. B., and Zhong, X. H.
Linear fitting Kalman filter.
IET Signal Processing, 10(4):404–412, Jun. 2016.

[10] Yang, T. E.
Acoustic Dopplergram for Intruder Defense.
Proceedings of IEEE Oceans 2007, Vancouver, BC, Canada, Sep. 2007.

[11] Yang, R., Bar-Shalom, Y., Huang, J. A. H., and Ng, G. W.
Interacting multiple model unscented Gauss-Helmert filter for bearings-only tracking with state-dependent propagation delay.
Proc. 17th International Conference on Information Fusion, Salamanca, Spain, Jul. 2014.

[12] Yang, R., Bar-Shalom, Y., Huang, J. A. H., and Ng, G. W.
UGHF for acoustic tracking with state-dependent propagation delay.
IEEE Transactions on Aerospace and Electronic Systems, 51(3):1747–1761, Aug. 2015.

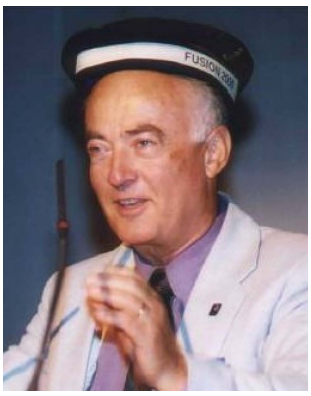
[13] Yang, R., and Bar-Shalom, Y.
Comparison of altitude estimation using 2D and 3D radars over spherical Earth.
Proceedings of IEEE Aerospace Conference 2016, Big Sky, MT, USA, Mar. 2016.

[14] Yang, R., Bar-Shalom, Y., and Ng, G. W.
Bearings-only tracking with fusion from heterogenous passive sensors: ESM/EO and acoustic.
Proc. 18th International Conference on Information Fusion, Washington, DC, Jul. 2015.

[15] Yang, R., Bar-Shalom, Y., and Ng, G. W.
Bearings-Only Tracking with Fusion from Heterogenous Passive Sensors: ESM/EO and Acoustic.
Journal of Advances in Information Fusion, 11(2), Dec. 2016.

[16] Yuan, T., Bar-Shalom, Y., Willett, P., Mozeson, E., Pollak, S., and Hardiman D.
A multiple IMM estimation approach with unbiased mixing for thrusting projectiles.
IEEE Transactions on Aerospace and Electronic Systems, 48(4):3250–3267, Oct. 2012.

Rong Yang received her B.E. degree in information and control from Xi'an Jiao Tong University, China in 1986, M.Sc. degree in electrical engineering from the National University of Singapore in 2000, and Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore in 2012. She is currently a Principal Member of Technical Staff at DSO National Laboratories, Singapore. Her research interests include passive tracking, low observable target tracking, GMTI tracking, hybrid dynamic estimation and data fusion. She was Publicity and Publication Chair of FUSION 2012 and received the FUSION 2014 Best Paper Award (First runner up).



Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering. From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab. His current research interests are in estimation theory, target tracking and data fusion. He has published over 500 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993; translated into Russian, MGTU Bauman, Moscow, Russia, 2011), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate texts *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), *Tracking and Data Fusion* (YBS Publishing, 2011), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multi-target tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas. During 1976 and 1977 he served as Associate Editor of the *IEEE Transactions on Automatic Control* and from 1978 to 1981 as Associate Editor of *Automatica*. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–13. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is co-recipient of the M. Barry Carlton Award for the best paper in the *IEEE Transactions on Aerospace and Electronic Systems* in 1995 and 2000 and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008 he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012 the Connecticut Medal of Technology. In 2015, he received from the International Society for Information Fusion the Lifetime of Excellence in Information Fusion award. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in Aerospace Engineering based on the citations of his work.

Claude Jauffret was born in France on Mar. 29, 1957, and received the diplôme d'Etudes Appronfondies in Applied mathematics from Saint Charles University, Marseille, France, in 1981, the Diplôme d'Ingénieur from Ecole Nationale Supérieure d'Informatique et de Mathématiques Appliqués de Grenoble, Grenoble, France, in 1983, the title of "Docteur de l'Université" in 1993 and the "Habilitation à Diriger des Recherches" from the Université de Toulon et du Var, France.

From Nov. 1983 to Nov. 1988, he worked on passive sonar systems, more precisely on target motion analysis at GERDSM, France. After a sabbatical year at the University of Connecticut (from Nov. 1988 to Dec. 1989) during which he worked of tracking problems in cluttered environment, and he developed researches in tracking, data fusion, and extraction in CERDSM. Since Sept. 1996, he has been at the Université de Toulon where he teaches statistical signal processing. His current researches are about observability, estimation in nonlinear systems as they appear in tracking problems.

Annie-Claude Pignol was born in France on Nov. 10, 1965, received the Diplôme d'Etudes Appronfondies in Optics and Image Processing from the Université de Toulon et du Var, Toulon, France, in 1988 and the title of "Docteur de l'Université" in 1991 from the Université de Toulon et du Var, France.

Since Sept. 1994, she has been at the Université de Toulon where she teaches electronic systems. Her researches were focused signal processing applied to biomedical systems before turning them to Target Motion Analysis.

Gee Wah Ng received his M.Sc. and Ph.D. from University of Manchester Institute of Science and Technology, United Kingdom. He is currently a Distinguished Member of Technical Staff and Programme Director (Information Exploitation) of Information Division at DSO National Laboratories. He has delivered many projects in the decision support areas and has authored three books. He is active in international conferences in the areas of information fusion and intelligent systems. His research interests in data and information fusion include target tracking, computational intelligence, machine learning, self-tuning and sensor networks. He was General Chair of FUSION 2012.

Approaches to Obtain a Large Number of Ranked Solutions to 3-Dimensional Assignment Problems

LINGYI ZHANG
DAVID SIDOTI
SPANDANA VALLABHANENI
KRISHNA R. PATTIPATI
DAVID A. CASTAÑÓN

A generalized 3-dimensional assignment problem is a decision-making process that involves allocating limited resources to a set of tasks over time, where the objective is to optimize a cost function subject to a set of generalized assignment constraints. The 3-dimensional (3-D) assignment problems are known to be NP-hard. In this paper, we propose a novel approach to efficiently solve an m-best 3-D assignment problem with non-unity right-hand side constraints (also referred to simply as 3-D assignment problem), where m may be large (as many as 10^4 solutions), by decomposing it into two sequential phases. In phase I, we partition the original problem space into a series of subproblems via Murty's m-best search space decomposition procedure. Modifications previously proposed in the literature for the 2-dimensional (2-D) assignment problem are applied to optimize the search space decomposition for the 3-D assignment problem. In phase II, we solve each subproblem by using Lagrangian relaxation and solving the 3-D assignment problem as a combination of relaxed 2-D assignment problems and 2-D transportation problems. The 2-D assignment problem is solved by the JVC or auction algorithms, and the 2-D transportation problem is solved by the simplex-based transportation, Transauction or RELAX-IV algorithms. The sequence of relaxed 2-D problems are interchangeable, while adhering to the relaxed constraints. We validate and compare the performance and utility of the proposed algorithms and search space decomposition optimizations via extensive numerical experiments. Overall, the fully optimized algorithm took less than 50 seconds, on average, to obtain 10^4 soutions for a tensor of dimension $30 \times 30 \times 8$.

Manuscript received Sept. 9, 2016; revised February 7, 2017; released for publication June 26, 2017.

Refereeing of this contribution was handled by Stefano Coraluppi.

Authors' addresses: L. Zhang, D. Sidoti, S. Vallabhaneni, and K. R. Pattipati, Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, 06269 USA (E-mail: lingyi.zhang@uconn.edu; krishna.pattipati@uconn.edu). D. A. Castañón, Boston University, Boston, MA, 02215 USA (E-mail: dac@bu.edu).

Research supported by the U.S. Office of Naval Research under contract #N00014-16-1-2036 and by the Department of Defense High Performance Computing Modernization Program under subproject contract #HPCM034125HQU.

1. INTRODUCTION

1.1. Motivation

Assignment problems are applicable to a diverse array of real world problems [1]–[3]. This set of problems takes the form of how best to assign a number of items or objects to some (possibly different) number of machines or people during different time periods. Assignment problems are of a combinatorial nature, each requiring some form of an objective function to indicate the value or utility of individual assignments. A sampling of how diverse and widely applicable such assignment problems are can be seen from the following: multi-target tracking, quadratic assignment problems, traveling salesman problems, or vehicle routing problems. Such problems also occur in academia or the military, where a set of military troops [1] or teachers [2] must be assigned to locations or classrooms that are temporally dependent in value or utility. Assignment problems have even been motivated from a telecommunications standpoint, where a set of satellites must be launched from a set of locations to maximize their coverage [1].

A 2-dimensional (2-D) assignment problem may be viewed as a weighted bipartite graph matching problem, where arcs must link two sets of nodes together such that an objective function is optimized, while satisfying a set of one-to-one constraints. The 3-dimensional (3-D) extension of this problem has been proven to be NP-hard [4]–[6]. In particular, one application that we focus on in this paper is a nuclear fuel assembly (FA) loading pattern optimization. The core of a nuclear reactor is formed by large sets of elongated, rectangular FAs arranged in a cylindrical fashion, as shown in Fig. 1.

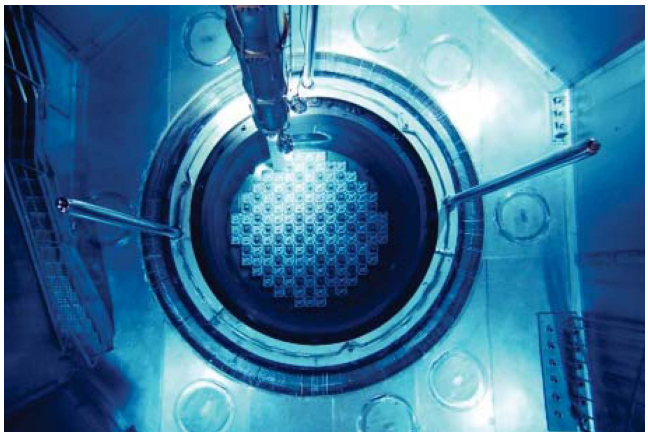


Fig. 1. The core of a nuclear reactor is formed by large sets of fuel assemblies where position, type, and rotation/orientation must be chosen for each one. Illustrated here is a nuclear fuel assembly loading operation at Fangchenggang nuclear power plant in China's Guangxi province [7].

The nuclear fuel assembly loading pattern optimization problem involves choosing: 1) the position of the FA in the nuclear reactor core, 2) the type of FA to put

in the chosen position, and 3) the rotation/orientation of the chosen FA type in the chosen position. Each dimension of the 3-D assignment corresponds to each of the decision variables above. In general, this problem is treated as a multiple objective combinatorial problem, but what separates it from the traditional 3-D assignment problems is the requirement for a dense set of new discrete loading patterns though a dynamically estimated probability distribution (represented by a reward tensor). This conversion to a 3-D assignment problem is a completely new approach for nuclear fuel loading pattern optimization. The reward tensor is dynamically updated based on the "best" solutions taken from the multi-objective Pareto front. "Best" in this case may not necessarily refer to the optimal, but one of a large number of solutions (assignments). By "large," we mean on the order of 10^4 solutions. Evaluation of each loading pattern by reactor-physics-based external code may be very time consuming (≈ 0.1 to 10 minutes, depending on the required accuracy of loading pattern response evaluation), so there exists a need to evaluate only new (unique) loading patterns (assignments).

In such scenarios, an m-best 3-D assignment problem is needed, wherein a large set of solutions is generated in a reasonable amount of time (< 10 minutes for 10^4 solutions), so that the set of assignments may be externally evaluated (each of which, in turn, may take 0.1 to 10 minutes). It may also be a viable approach to obtain a dense set of solutions that are near-optimal and satisfy the decision maker (such as in the case of resource allocation or military troop allocation problems) or customer preferences (as in [2], where they attempt to satisfy both student and tutor requirements or requests). Having a large set of solutions offers a range of options that may be of interest to a decision maker attempting to optimize with respect to multiple, possibly conflicting, objectives.

This paper offers an effective solution approach for finding a large number of m-best solutions to the 3-D assignment problems with non-unity right-hand side constraints with application to many real world challenges. The problem space may be decomposed into multiple partitions based on the optimal assignment, as detailed in [8]. Through a two-phase approach, we offer a method for rapidly generating large numbers of solutions to the 3-D assignment problems.

1.2. Related Research

There exist a number of well-known algorithms to obtain the optimal solution to a 2-D assignment problem, including the Hungarian algorithm [9], the Jonker-Volgenant-Castañón (JVC) algorithm [10], [11], the auction algorithm [12], and the signature method [13]. However, the assignment problem becomes NP-hard when a third dimension is added [4]–[6]. One of the first approaches for solving the 3-D assignment problem was developed by Pierskalla [1], where he proposed a tri-substitution algorithm based on the simplex method. Hansen [14] proposed a primal-dual implicit enumeration algorithm, while [15], [16] proposed branch-and-bound approaches to obtain the optimal solution to such 3-D assignment problems. However, branch-and-bound methods suffer from exponential computational complexity and are unsuitable for large-scale real-world applications where accurate bounds cannot be obtained.

In order to overcome the 3-D assignment problem's inherent computational intractability, a wide range of algorithms have been developed to obtain suboptimal solutions, including greedy heuristics, genetic algorithms, simulated annealing, tabu search, neural networks, and Lagrangian relaxation approaches [2], [17]–[20]. Mazzola [17] proposed a heuristic branch-and-bound method to reduce the computation time. In contrast, Frieze and Yadegar [2] applied Lagrangian relaxation theory to a more general 3-D assignment problem with application to teaching practice scheduling. The Lagrangian relaxation method of obtaining solutions to 3-D assignment problems has become extremely prevalent in data association applications due to the real time computation speed and solution quality [3], [18], [19]. Poore [20] combined these two approaches, proposing a hybrid branch-and-bound and Lagrangian relaxation algorithm to the 3-D assignment problem.

In this paper, we seek to solve the aforementioned 3-D assignment problem, but instead of finding a single solution, we aim to provide a large set of ranked solutions. The process of finding the first best, second best, third best, and so on, solution is known as the m-best optimization problem. The m-best optimization problem occurs in a variety of contexts, including the shortest path [21]–[23], spanning tree [24]–[26], traveling salesman [27], directed network [28], multi-target tracking [29]–[33] and many other problems. The general approach to the m-best optimization problem involves partitioning the solution space into smaller subspaces, which are subproblems of the original problem. Murty's search space decomposition [8] is the most common and widely used technique, where the best solution is found for each partitioned subproblem, given a modified solution subspace. Lawler [34] applied Murty's search space decomposition procedure within a more general framework for a discrete optimization problem. Pascoal [35] proposed a variant of Murty's search space decomposition to reduce the algorithm's complexity. This variant involved solving the partitioned subsets in reverse order. Miller et al. [36] proposed modifications to optimize Murty's search space decomposition procedure to the 2-D assignment problem via: 1) inherited dual variables and partial solutions from the initial subproblems; 2) sorting the subproblems based on lower bounds on the optimal reward before solving the assignment problem; and 3) partitioning in an order based on lower bounds on cumulative reward. These modifications substantially reduce the complexity of Murty's search space decomposition and are implemented in this paper.

Another alternative way to solve the m-best optimization problem is by Gabow's [24] binary heap partition method. Similarly, Hamacher [37] also proposed using a binary search tree procedure, while also combining an approach developed by Carraresi and Sodini [38] to rank the paths. Chegireddy and Hamacher [39] extended this work further and developed an m-best perfect matching algorithm based on the binary partition of the solution space to apply to a bipartite matching problem in $\mathcal{O}(kn^3)$ time. Recently, a modified version of the Chegireddy and Hammacher's algorithm was developed for large datasets [40]. We suggest comparison of our algorithm with those in [40] as future research.

1.3. Paper Organization

The primary focus of this paper is on combining a Lagrangian relaxation method and m-best optimization to obtain a very large number of ranked solutions. Motivated by an approach developed by Pattipati [18], we apply the Lagrangian relaxation approach that successively solves a series of 2-D problems, since a key advantage of using the Lagrangian relaxation method is that it prunes the solution space by computing the upper and lower bounds. The first 2-D problem is a bipartite graph matching problem (2-D assignment problem), which can be solved using either the auction algorithm or the JVC algorithm [10]; the latter is more efficient for dense problem spaces [11]. The feasible solution is obtained by solving a 2-D transportation problem (via a simplex algorithm or Transauction algorithm) reconstructed from the relaxed solution of the 2-D assignment problem. The second step corresponds to imposing the originally relaxed constraint on the first subproblem's solutions. As in [33], we generate m-best solutions by exploiting Murty's search space decomposition procedure; however, unlike the formulation in [33], the present 3-D formulation has general constraints that require a transportation problem to be solved. Moreover, we optimize Murty's search space decomposition via Miller's [36] proposed modifications, resulting in further speedup and improved computational performance. An alternate Lagrangian relaxation method involves first solving a 2-D transportation problem at each iteration of the 3-D assignment algorithm using either a simplex algorithm or the Transauction algorithm, and subsequently reconstructing the feasible solution via a 2-D assignment problem. We will show that the former Lagrangian relaxation method is two orders of magnitude faster than the latter.

This paper is organized as follows. We begin by introducing the problem formulation in Section 2. In Section 3, we solve the m-best 3-D assignment problem via Murty's search space decomposition and the Lagrangian relaxation method. In Section 4, we detail Miller et al.'s [36] search space optimizations and extend them to the 3-D assignment problem. We provide the pseudocode of the fully optimized m-best 3-D assignment solution algorithm in Section 5. In Section 6, we present the results of the m-best 3-D assignment algorithm and the performance of each different optimization technique. Finally, we provide concluding remarks in Section 7.

TABLE I
Summary of Notation

w_{ijk}	Reward of allocating resource i to task j at time k
x_{ijk}	Binary decision variable for the primal problem
y_{ij}	Binary decision variable for the 2-D assignment problem
z_{jk},z_{ik}	Binary decision variables for the 2-D transportation problem
i	Resource index
j	Task index
k	Time index
m_k	Maximum number of assignment allowed for each k
W	Reward tensor
N	Total number of tasks/resources
R	Total number of time units
μ	Lagrange multiplier
q	Upper bound found from the relaxed 2-D assignment problem (dual)
f	Lower bound found via simplex-based transportation or Transauction problem (primal)
$\underline{g}$	Gradient vector for the subgradient update
$\mathcal{P}_0$	Original problem space
A	Solution space
S	Feasible assignment in solution space A
X	Solution tensor
Φ	Column for row solution
ϕ	Optimal reward from the 2-D assignment problem
Ω	Layer for row solution
ω	Optimal reward from the 2-D transportation problem
B	Slack value for upper bound reward computation
C	Relaxed 2-D reward matrix in the 2-D assignment problem
T	2-D reward matrix for the transportation problem

2. PROBLEM FORMULATION

The notation used in the remainder of this paper is listed in Table I.

2.1. Problem Formulation

Given a 3-D reward tensor $W = [w_{ijk}]$ of dimension $N \times N \times R$, our problem is the following:

$$\max_{x_{ijk}\in\{0,1\}} \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{R} w_{ijk}x_{ijk} \tag{1}$$

$$s.t. \sum_{j=1}^{N}\sum_{k=1}^{R} x_{ijk} = 1,\ i = 1,\ldots,N \tag{2}$$

$$\sum_{i=1}^{N}\sum_{k=1}^{R} x_{ijk} = 1,\ j = 1,\ldots,N \tag{3}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ijk} \le m_k,\ k = 1,\ldots,R \tag{4}$$

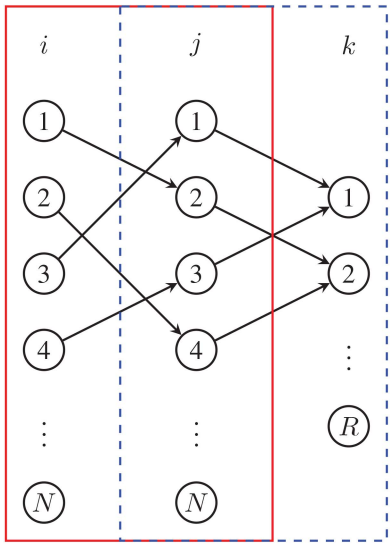


Fig. 2. Network flow view of the 3-D assignment problem, originally presented in [42].

where x_{ijk} is a binary decision variable such that $x_{ijk} = 1$ if resource (row) i is assigned to task (column) j at time (layer) k, and 0 otherwise. Constraints (2) and (3) ensure that each resource i is allocated to exactly one task j and vice versa. Constraint (4) requires that there may be no more than m_k assignments at each time k and makes this assignment problem non-standard.

Figure 2 shows the 3-D assignment problem as a network flow problem. Consider the first set, indexed by i, and the second set, indexed by j, each consisting of N nodes. Also, consider a third set, indexed by k, with a total of R nodes. There are a total of N assignments that may be made between sets i and j based on constraints (2) and (3). We view this as a 2-D assignment problem (indicated by the solid box in Fig. 2). Additionally, each node in set j must be assigned to one of the nodes in set k (indicated by the dashed (blue) box in Fig. 2). Due to constraint (4), for every k, there may be no more than m_k assignment pairs of (i,j) mapped to each layer. This may be viewed as an unbalanced transportation problem, where the nodes in set (i,j) are the sources and the nodes in set k are the sinks. Note that $\{m_k : k = 1,2,\ldots R\}$ should be such that $\sum_{k=1}^{R} m_k \geq N$ so that each (i,j) can be assigned to a node k.

2.2. Special Cases

Note that our 3-D assignment problem formulation covers a wide range of problems.

2.2.1. Tri-index Assignment problem: The problem in (1)–(4) may be viewed as a traditional tri-index assignment problem by setting $m_k = 1$ and $R = N$ [1].

2.2.2. Scheduling problem: By setting $m_k = m$, the problem in (1)–(4) is related to some resource-constrained assignment scheduling problems [41].

2.2.3. Transportation problem: Note that our problem formulation is a special case of the transportation problem. The general transportation problem involves altering the unity constraint to some non-unity values.

2.2.4. Nuclear Fuel Loading Pattern Optimization: In some nuclear reactor fuel assembly loading pattern optimization problems, $m_k = N$ on the right hand side of the constraint (4). In this case, the problem can be reduced to the traditional 2-D assignment problem, since constraint (4) can be subsumed under constraints (2) and (3) and is, thus, unnecessary. The 3-D assignment problem posed in (1) then devolves to a 2-D assignment problem, detailed later in Section 3.1.4. An m-best 2-D assignment problem is adequate for this version of the problem.

3. SOLUTION APPROACH

In order to solve this NP-hard problem, we propose a two-phase solution approach. In phase I, we utilize Murty's search space decomposition to partition the original problem space into a series of subproblems. Each subproblem is then relaxed and solved by a 3-D assignment algorithm in phase II.

3.1. 3-D Assignment Relaxations

We adopt the solution approach of the 3-D assignment problem in [18] by relaxing one of the three constraints and solving the 3-D assignment problem as a series of 2-D subproblems. Since sets i and j have the unity constraint, a similar solution approach can be applied to the 3-D assignment problem here by relaxing either of the two sets of constraints. We then denote Relaxation Method I and Relaxation Method II as the solution approaches for the 3-D assignment problem when constraints (4) or (2)/(3) are relaxed, respectively.

3.1.1. Relaxation Method I: Relaxation Method I is developed by relaxing constraint (4) via a set of Lagrange multipliers $\{\mu_k : k = 1,2,\ldots,R\}$. The result is the Lagrangian function

$$L(x,\mu) = \max_{x_{ijk}\in\{0,1\}} \left(\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{R} (w_{ijk} - \mu_k) x_{ijk} \right) + m_k \sum_{k=1}^{R} \mu_k \tag{5}$$

Equation (5) is then a relaxed 2-D assignment problem of the form,

$$\max_{y_{ij}\in\{0,1\}} \sum_{i=1}^{N}\sum_{j=1}^{N} \max_{k} (w_{ijk} - \mu_k) y_{ij} \tag{6}$$

$$s.t. \sum_{i=1}^{N} y_{ij} = 1,\ j = 1,\ldots,N \tag{7}$$

$$\sum_{j=1}^{N} y_{ij} = 1,\ i = 1,\ldots,N \tag{8}$$

where,

$$y_{ij} = \sum_{k=1}^{R} x_{ijk};\ i,j = 1,\ldots,N. \tag{9}$$

The upper bound q of the relaxed 2-D assignment problem is easily solvable via a 2-D assignment algorithm. To obtain a feasible solution, we reimpose constraint (4) by reconstructing the reward tensor and viewing the asymmetric bipartite graph as a transportation problem based on the solution of the relaxed 2-D assignment problem. For each $\langle i^*, j^* \rangle$ of the relaxed 2-D assignment problem at each iteration, the reward matrix is dynamically updated for each layer k. Given a new reward matrix $\tilde{w}_{\langle i,j \rangle k}$, the transportation variation of the problem is as follows.

$$\max_{z_{jk} \in \{0,1\}} \sum_{j=1}^{N} \sum_{k=1}^{R} \tilde{w}_{\langle i,j \rangle k} z_{jk} \tag{10}$$

$$s.t. \sum_{j=1}^{N} z_{jk} = 1,\ k = 1,\ldots,R \tag{11}$$

$$\sum_{k=1}^{R} z_{jk} \leq m_k,\ j = 1,\ldots,N \tag{12}$$

Through this sequence, we obtain a feasible solution and a lower bound f. The upper and lower bounds serve as measures of the solution quality. The distance between these bounds is referred to as the approximate duality gap (because it is overestimated by $(f - f^*)$, where f^* is the optimal solution). For discrete 3-D assignment problems, the duality gap may be nonzero. The relative approximate duality gap is given by

$$gap = \frac{|q - f|}{f}, \tag{13}$$

where q and f are the upper and and lower bounds, respectively, obtained by solving the series of 2-D subproblems. The 3-D assignment algorithm terminates for a sufficiently small gap, which implies that a near-optimal solution has been obtained. In scenarios where the duality gap is large, the 3-D assignment algorithm updates its Lagrange multipliers via the method proposed in Pattipati [18]. Let us denote $\underline{g}$ as an R-dimensional subgradient vector with components given by

$$g_k = R - \sum_{i=1}^{N} \sum_{j=1}^{N} X_{ijk} \quad k = 1,\ldots,R, \tag{14}$$

where X is the solution tensor related to the optimal value of the relaxed 2-D assignment variables $\{y_{ij}^*\}$ via

$$X_{ijk} = \begin{cases} y_{ij}^*, & \text{if } k = \arg\min_\alpha (w_{ij\alpha} - \mu_\alpha) \\ 0, & \text{otherwise} \end{cases}$$

We then update the Lagrange multipliers by

$$\mu_k = \max\left(\mu_k - \frac{(p - f)}{\|\underline{g}\|_2^2} g_k, 0 \right). \tag{15}$$

After updating the Lagrange multipliers, the algorithm iterates back to the relaxation step. The process continues until either the maximum number of iterations is reached or the duality gap is sufficiently small. The flow diagram of the 3-D assignment algorithm when the constraint in (4) is relaxed is shown in Fig. 3.

3.1.2. Relaxation Method II: Note that a relaxed problem is also obtainable by interchanging the sequence of 2-D subproblems. In other words, we may apply the Lagrangian relaxation on constraints (2) or (3). When constraint (3) is relaxed via Lagrange multipliers μ_j, the Lagrangian function is:

$$L(x,\mu) = \max_{x_{ijk} \in \{0,1\}} \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{R} (w_{ijk} - \mu_j) x_{ijk} \right) + \sum_{j=1}^{N} \mu_j \tag{16}$$

The 3-D assignment problem is then relaxed into a 2-D transportation problem of the form

$$\max_{z_{ik} \in \{0,1\}} \sum_{i=1}^{N} \sum_{k=1}^{R} \max_j (w_{ijk} - \mu_j) z_{ik} \tag{17}$$

$$s.t. \sum_{k=1}^{R} z_{ik} = 1,\ i = 1,\ldots,N \tag{18}$$

$$\sum_{i=1}^{N} z_{ik} \leq m_k,\ k = 1,\ldots,R, \tag{19}$$

where

$$z_{ik} = \sum_{j=1}^{N} x_{ijk};\ i = 1,\ldots,N;\ k = 1,\ldots,R \tag{20}$$

The upper bound q can be obtained by solving the relaxed 2-D transportation problem. The 2-D assignment problem is obtained by reimposing constraint (3) and reconstructing the reward tensor based on the solution of the relaxed 2-D transportation problem. The assignment variation of the problem is as follows.

$$\max_{y_{ij} \in \{0,1\}} \sum_{i=1}^{N} \sum_{j=1}^{N} \tilde{w}_{\langle i,k \rangle j} y_{ij} \tag{21}$$

$$s.t. \sum_{i=1}^{N} y_{ij} = 1,\ j = 1,\ldots,N \tag{22}$$

$$\sum_{j=1}^{N} y_{ij} = 1,\ i = 1,\ldots,N \tag{23}$$

A feasible solution and a lower bound f can be obtained through this sequence. The duality gap is then computed and compared for algorithm termination. The subgradient is updated in a similar fashion to the first relaxation

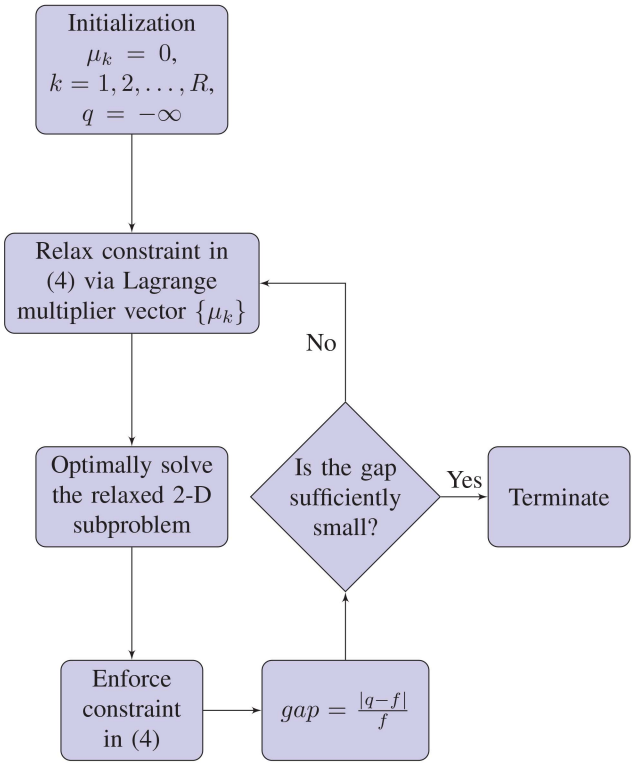


Fig. 3. Flow diagram of the 3-D assignment algorithm when relaxing constraint (4).

method, except that it is with respect to dimension j and uses binary decision variables $\{z_{ik}\}$.

3.1.3. Algorithm selection for 2-D subproblems: To optimize the 3-D assignment algorithm, state-of-the-art 2-D assignment and transportation algorithms were selected for comparison purposes. The JVC and auction algorithms were selected for comparison when solving the 2-D assignment problem. We solve the 2-D transportation problem via three approaches. The first algorithm utilizes the Transauction algorithm developed by Bertsekas and Castañón [43], which solves the transportation problem by mapping it to an assignment problem and obtains a solution via a modified auction algorithm. In the second algorithm, we exploit the findings in [6], [44], [45], where the transportation problem was found to be equivalent to the minimum cost network flow problem, and solve the 2-D transportation problem via a (strongly polynomial) simplex-based method. We refer to this simply as simplex-based transportation. The third algorithm is the RELAX-IV algorithm developed by Bersekas and Tseng [46] and further detailed in [47]. It is one of the most efficient algorithms to solve problems of the network flow type.

3.1.4. Solution approach for a variant of the nuclear FA loading pattern optimization problems: for this problem, constraint (4) is such that $m_k = N$. In this case, the summations over sets i and j are always less than or equal to N for each k, and, consequently, constraint (4) is always satisfied. This implies that the constraints in (4) are inactive and the Lagrange multipliers $\mu_k = 0$ for $k = 1, 2, \ldots, R$. Consequently, the 3-D assignment problem takes the form,

$$\max_{y_{ij} \in \{0,1\}} \sum_{i=1}^{N} \sum_{j=1}^{N} \max_{k}(w_{ijk}) y_{ij} \tag{24}$$

$$s.t. \sum_{i=1}^{N} y_{ij} = 1, \; j = 1, \ldots, N \tag{25}$$

$$\sum_{j=1}^{N} y_{ij} = 1, \; i = 1, \ldots, N \tag{26}$$

This problem can be easily solved by an m-best 2-D assignment algorithm. Furthermore, the approach is the same for the general case when $m_k \geq N$.

3.2. m-best 3-D Assignment

Let $\mathcal{P}_0$ be the original problem in equations (1)–(4) and let A be the corresponding assignment solution space. Further, let A_0^* be the best feasible assignment found by the 3-D assignment algorithm detailed in Section 3.1. In general, to find the $(n+1)$th best solution, we have to partition the $(n+1)$th problem space, $\mathcal{P}_n$, into N subproblems, denoted by $\mathcal{P}_{nr}$, $1 \leq r \leq N$. Then, the complete solution space corresponding to problem space $\mathcal{P}_n$ is

$$A_n = \bigcup_{r=1}^{N} A_{nr} = A - \bigcup_{i=0}^{n-1} A_i^* \quad \text{for } n = 1, 2, \ldots, m \tag{27}$$

$$A_{nr} \cap A_{ns} = \emptyset \quad \text{for } r, s = 1, 2, \ldots, N \quad r \neq s, \tag{28}$$

where A_{nr} denotes a set of tuples in which each i and j appear exactly once, but k may be repeated. Equation (27) is a formalization of the constraint that the solution space A_n for the $(n+1)$th best solution will not contain any of the best solutions obtained for the previous n problems. Here, a complete feasible solution is assumed to be a set of tuples. Hence, some solutions may have a similarity, however, as seen in (28), the set of solution tuples as a whole are unique, differing by at least one element for each of the previous n problems. Let an assignment A_{nr} consist of multiple tuples (in this paper, triples), where we index the triples within by t. Let ℓ_{nr_t} be the individual reward of the tth triple, sub-indexed as $\langle i_{nr_t}, j_{nr_t}, k_{nr_t} \rangle$, in the solution space A_{nr}. We can then augment the triple into a 4-tuple and write a feasible assignment in A_{nr} as

$$S_{nr} = \{\langle i_{nr_t}, j_{nr_t}, k_{nr_t}, \ell_{nr_t} \rangle\} \quad \text{for } t = 1, \ldots, N. \tag{29}$$

The primal value of the corresponding assignment S_{nr} is denoted by f_{nr}, which can be obtained by summing ℓ_{nr_t} over $t = 1, 2, \ldots, N$.

$$f_{nr} = \sum_{t=1}^{N} \ell_{nr_t} \tag{30}$$

The best assignment A^*_{nr} with the corresponding primal value f^*_{nr} in the solution space A_{nr} is found via the 3-D assignment algorithm described earlier and pertains specifically to partition r. The best assignment A^*_n is found by iterating over all active partitions and finding the argument r^* which has the maximum primal value.

$$A^*_n = A^*_{nr^*} \tag{31}$$

$$r^* = \arg\max_r f^*_{nr} \tag{32}$$

Given the original problem space and its optimal assignment, denoted by $\mathcal{P}_0$ and A^*_0, respectively, we partition $\mathcal{P}_0$ into N problem subspaces $\mathcal{P}_{11}$ to $\mathcal{P}_{1N}$ in order to find the next best solution. To generate subproblem $\mathcal{P}_{11}$, we remove the first of N tuples in the assignment A^*_0. We then use the 3-D assignment algorithm to obtain the best possible solution A^*_{11} to problem $\mathcal{P}_{11}$. To partition the subspace $\mathcal{P}_{1s}$, $2 \leq s \leq N$, we remove the sth tuple in A^*_0 as a feasible assignment in $\mathcal{P}_{1s}$, while fixing the first $(s-1)$ triples to those in the original assignment A^*_0. Thus, as the solution and problem spaces are reduced at every search space decomposition, the complexity of the problem decreases substantially, since the first $(s-1)$ triples are reused from the previous assignments. We then only need to find assignments for the remaining $N-s$ assignments, such that the sth triple from the original assignment A^*_0 is not contained in the solution, while satisfying the constraints. The enforcement of tuples to be either in or be removed from the problem spaces $\mathcal{P}_{11}$ to $\mathcal{P}_{1N}$ during partitioning ensures the disjointness of the individual subproblems, as in equation (28).

Each of the best solutions A^*_{11} to A^*_{1N} is saved into a heap and accordingly sorted based on the respective primal values, f^*_{11} to f^*_{1N}. The best solution within the heap is then removed and saved as the second best solution. The problem corresponding to the second best solution is then partitioned into the subproblems $\mathcal{P}_{21}$ to $\mathcal{P}_{2N}$. The best assignment from the top of the heap is then marked as the third best assignment with respect to the original problem $\mathcal{P}_0$. We continue to apply this process until the mth best solution is found or, alternatively, the heap becomes empty.

Murty's search space decomposition is an ingenious way of decomposing the search space, and has a number of applications in combinatorial optimization [34], [48]. Optimizations of the decomposition technique to improve the computational efficiency are discussed in Section 4.

REMARK For small size problems and large m, if we apply the Lagrangian relaxation on constraint (4), the transportation problem reconstructed from the best (i, j) pair of the 2-D assignment problem may contain too many removed arcs and, thus, no feasible solution may exist. In this case, by interchanging the sequence of relaxed problems solved (i.e., solve the 2-D transportation problem first, as opposed to the assignment problem (normally solved first)), we can obtain a feasible solution to the 3-D assignment problem. This situation arises in small size problems (e.g., of dimension $3 \times 3 \times 2$). However, since the tensor dimensions used in this paper are large, the solution space is vast and this anomaly did not arise.

4. OPTIMIZED IMPLEMENTATION OF MURTY'S SEARCH SPACE DECOMPOSITION

We extend the 2-D optimization modifications in [36] to the 3-D assignment problems. These include: 1) inheriting the dual variables and partial solutions from the subproblems being decomposed; 2) sorting the subproblems by an upper bound on reward before solving; and 3) partitioning the subproblems in an optimized order. All three modifications exploit the primal-dual aspects of the JVC algorithm. The following sections explain each modification in detail for the case when the constraints in (4) are relaxed in the m-best 3-D assignment algorithm. Similar optimization techniques can be applied for the case when the constraints in (3) are relaxed.

4.1. Inheriting dual variables and partial solutions during partitioning

Solving the 3-D assignment problem via the JVC algorithm provides dual variables u and v, which can be inherited by the partitioned subproblem using Murty's search space decomposition. The solution tensor X_n, for the problem space $\mathcal{P}_n$ and the reward tensor W, contains N solution triples $\langle i^*, j^*, k^* \rangle$. During each step of Murty's search space decomposition, a new subproblem $\mathcal{P}_{nr}$ is generated, associated with a new reward tensor W'. Removing the triple $\langle i^*, j^*, k^* \rangle$ from the subproblem space $\mathcal{P}_{nr}$ is equivalent to setting $w_{\langle i^*, j^*, k^* \rangle} = -\infty$. This implies we may skip the initialization step for the JVC algorithm and go directly to the augmentation step with only one arc left to assign in the 2-D assignment problem, following the procedure outlined in Algorithm 1. In this case, the initialization step is only required for the first feasible solution to the 3-D assignment problem.

ALGORITHM 1 *Upper bound reward calculation when inheriting dual variables*

1: **for each** $\langle i^*, j^*, k^* \rangle \in A$ **do**
2: $\quad w_{\langle i^*, j^*, k^* \rangle} = -\infty$
3: $\quad u' = u$, $v' = v$,
4: $\quad X' \leftarrow X - X_{\langle i^*, j^*, k^* \rangle}$
5: **end for**

Note that we can not inherit the Lagrange multipliers μ_k from the previous problem $\mathcal{P}_n$ in the process of partitioning the subproblems. The Lagrange multipliers from the previous problem $\mathcal{P}_n$ may be too large for the subproblems $\mathcal{P}_{nr}$, $r = 1, 2, \ldots, N$. This may cause the duality gap to remain above the threshold value required

to terminate. Thus, the algorithm will continue to run until the maximum iteration limit is reached.

4.2. Sorting subproblems via an upper bound

The upper bound reward of individual subproblems is easily obtainable and can be used to avoid solving subproblems that are unlikely to produce the next best solution. For an m-best assignment problem, the best solution from problem $\mathcal{P}_n$ is always better than the best solution obtained from the subproblems obtained by partitioning $\mathcal{P}_{nr}$, $r = 1,2,\ldots,N$. Therefore, for an m-best 2-D assignment problem, the objective function of the solution to $\mathcal{P}_n$ can be used as an initial upper bound on the objective function value of the best solution to its corresponding subproblems. Since 3-D assignment problems may have a nonzero duality gap, the computation of the upper bound can be determined using either the dual value (denoted by ϕ) or the primal value (denoted by ω) as initial upper bounds to the partitioned subproblems.

When a subproblem $\mathcal{P}_{nr}$ is created by removing a triple $\langle i^*, j^*, k^* \rangle$ from a copy of $\mathcal{P}$, we can compute the upper bound objective function value by finding the best slack (i.e., next possible best assignment) of all the alternative assignments for a row i. The upper bound objective function value will be the sum of the initial upper bound and the row slack, denoted by B_r. The calculation of the upper bound is shown in detail in Algorithm 2.

ALGORITHM 2 *Upper bound reward calculation when sorting subproblems*

1: **for each** row i **do**
2: $\quad w_{\langle i^*, j^*, k^* \rangle} = -\infty$
3: $\quad B_r = \max_{j,k}\{w_{ijk} - u(i^*) - v(j) - \mu(k)\}$
4: $\quad f' = f + B_r$
5: **end for**

A similar procedure can be followed for column j to find the column slack, B_c. By combining both the row and the column slack, a tighter upper bound can be obtained. The heap of subproblems can be modified to sort its elements (in descending order) based on each element's respective upper bound reward. This implies that the problems located at the top of the heap are most likely to have the best solutions.

In this optimization method, the initial problem is partitioned into a series of subproblems when it is solved by the 3-D assignment algorithm. Both the original problem and its corresponding subproblems are saved into a heap. During each iteration of Murty's search space decomposition, if the top problem $\mathcal{P}_n$ removed from the heap has a feasible solution, then the solution will be saved as the mth best assignment. If $\mathcal{P}_n$ has not yet been solved (i.e., it has a partial solution), then we find its best solution A_n^* using the 3-D assignment algorithm and add it back into the heap. A new partitioning process is then invoked on $\mathcal{P}_n$ and its solution A_n^*. The process is repeated until the heap is empty or a total of m solutions are obtained. This method allows us to eliminate subproblems by focusing on their corresponding upper bounds, thus reducing the number of problems needed to be solved by the 3-D assignment algorithm.

4.3. Partition in an optimized order

The third optimization method proposed here is to carefully select the *order* in which the partitioning is performed. This modification maximizes the probability that the subsequent smaller subproblems (with a greater number of fixed arcs) have better solutions. For problem $\mathcal{P}_n$ with solution A_n^* that contains N triples, we first compute each upper bound reward that would result from excluding each individual arc. These upper bounds are computed via the method explained in Section 4.2. We then select the triple that corresponds to the lowest upper bound reward computed and exclude it from the current subproblem, while fixing the corresponding arc in the next subproblem.

In this modification, the heuristic tends to ensure that the largest problem (maximum number of unassigned arcs) has the lowest upper bound. In other words, the largest problem has the highest probability of containing the worst solution and to be pushed to the bottom of the heap (and in turn, will most likely remain unsolved upon algorithm termination). The next worst problem will tend to be the second largest subproblem, and so on. By doing this, we increase the chance that the smallest problem (that which has the least amount of unassigned arcs) contains the best solution.

5. PSEUDOCODE

The following variants were used and/or combined for different optimization methods:

(A) Inheritance of the dual variables and partial solutions during partitioning
(B) Sorting subproblems by an upper bound reward before solving, where the upper bound is calculated via:
- i $\omega + B_r$
- ii $\omega + B_r + B_c$
- iii $\phi + B_r$
- iv $\phi + B_r + B_c$

(C) Partitioning the problem in an optimized order

These variants are denoted as listed for the remainder of the paper and may be combined, e.g., when combining variant A with variant B(ii) and variant C, the algorithm variant will be categorized as A+B(i)+C. The pseudocode for Murty's modified search space decomposition, optimized via variants A, B(ii), and C, is detailed in Algorithm 3. These variants assume JVC and Transauction to be applied in the m-best 3-D assignment algorithm.

ALGORITHM 3 *m-best 3D assignment algorithm*

$H \leftarrow \{\}$ Initialize binary heap
$U \leftarrow [\,]$ Initialize solution list
$\langle A_0^*, P_0, f_0^* \rangle$=3DASSIGN($w_{ijk}$)
PARTITION($H, \mathcal{P}_0, A_0^*$) Invoke PARTITION method
$H \leftarrow \langle A_0^*, \mathcal{P}_0, f_0^* \rangle$ Add to the heap
counter=0
while counter $\leq m-1$ and $H \neq \emptyset$ **do**
  $\langle A_n^*, \mathcal{P}_n, f_n^* \rangle = H.pop$
  if A_n^* is feasible **then**
    counter=counter+1
    $U \leftarrow A_n^*, f_n^*$
  else
    $\langle A_n^*, \mathcal{P}_n, f_n^* \rangle$=3DASSIGN($w_{ijk}, \langle A_n^*, \mathcal{P}_n, f_n^* \rangle$)
    if $\exists$ solution **then**
      PARTITION($H, \langle A_n^*, \mathcal{P}_n, f_n^* \rangle$)
      $H \leftarrow \langle A_n^*, \mathcal{P}_n, f_n^* \rangle$
    end if
  end if
end while

function PARTITION($H, \langle A_n^*, \mathcal{P}_n, f_n^* \rangle, w_{ijk}$)
  for each $\langle i^*, j^*, k^* \rangle \in A_n^*$ **do**
    $w_{i^*,j^*,k^*} = -\infty$
  end for
  for each $\langle i^*, j^*, k^* \rangle \in A_n^*$ **do**
    for each row $i^* \in A^*$ **do**
      $B_r = \max_{j,k}\{w_{ijk} - u(i^*) - v(j) - \mu(k)\}$
      $B_c = \max_{i,k}\{w_{ijk} - u(i) - v(j^*) - \mu(k)\}$
      $B_{i^*} = B_r + B_c$
    end for
    $(B, i^*) = \min(B_{i^*} \neq -\infty)$
    $\langle i^*, j^*, k^* \rangle = A_n^*(i^*)$
    $f_{nr}^* = f_n^* + B$
    $A_{nr}^* \leftarrow A_n^* - \langle i^*, j^*, k^* \rangle$
    $\mathcal{P}_{nr} \leftarrow \mathcal{P}_n - \langle i^*, j^*, k^* \rangle$
    $H \leftarrow \langle A_{nr}^*, \mathcal{P}_{nr}, f_{nr}^* \rangle$
    $A_{n(r+1)}^*.FixList \leftarrow \langle i^*, j^*, k^* \rangle$
    for each j,k **do**
      $w[i^*, j, k] = -\infty$
    end for
    for each row $\neq i^*, k$ **do**
      w[row, j^*, k] $= -\infty$
    end for
  end for
end function

function 3DASSIGN($w_{ijk}, \langle A_n^*, \mathcal{P}_n, f_n^* \rangle$)
  $f^* = -\infty$; lb= $-\infty$; $q^* = \infty$; maxIter= 20
  MAX= $true, n_3 = R$
  $FixList, v, \Phi \leftarrow A_n^*$
  for curIter= 1 to maxIter **do**
    $C = \max_k(w_{ijk} - \mu_k)$
    for $\langle i^*, j^*, k^* \rangle \in A_n^*.FixList$ **do**
      $C[i^*, j^*] = w[i^*, j^*, k^*]$
    end for
    if $\Phi == \emptyset$ **then**
      (Φ, u, v, ϕ) =JVC(C, MAX)
    else
      (Φ, u, v, ϕ) =Augment(C, Φ, v, MAX)
    end if
    $q^* = \min(q, \phi + n_3 * \sum_k(\mu_k))$
    for each row **do**
      T[row] = w[row, Φ[row], k] $\forall k$
    end for
    (Ω, ω) =Transportation(T, MAX)
    if $\omega \geq$ lb **then**
      lb= ω
      f^* =lb
    end if
    gap= $\dfrac{|q^* - f^*|}{|f^*|}$
    if gap ≤ 0.05 **then**
      return
    end if
    Update Lagrangian multiplier
  end for
end function

6. RESULTS

The proposed m-best 3-D assignment algorithm was implemented in the MATLAB 2016b and runs on an Intel Core i7-4712HQ CPU processor @2.30 GHz with 16 GB RAM. In all experiments, the top 10^4 ranked solutions were computed.

6.1. Relaxation Method I vs. Relaxation Method II

We first performed 10 Monte Carlo runs to compare the simulation runtimes of the 3-D assignment algorithm when relaxing either constraint (3) or constraint (4). The reward tensor elements were uniformly distributed in the interval $[0,1]$ and of dimension $60 \times 60 \times 8$. The JVC and Transauction algorithms were implemented to solve the 2-D assignment and the transportation problems, respectively. As shown in Table II, speedup of as much as 2.28 and an average speedup of 1.63 were observed when comparing the two relaxation methods. In general, solving a 2-D assignment problem is significantly faster than solving a transportation problem. The transportation problem obtained from relaxing constraint (3) is complex, and thus takes a longer time to solve compared to the transportation problem reconstructed from the best 2-D assignment solution when constraint (4) is relaxed. Relaxing constraint (4) also consistently resulted in a smaller duality gap compared to when constraint (3) was relaxed due to the fact that $|k| = R < N = |j|$. This implies that when constraint (4) is relaxed, a smaller number of elements in the 3-D reward tensor are removed when constructing the 2-D subproblem, i.e., since a smaller number of elements are removed, there is a higher likelihood that a better

TABLE II
10 Monte Carlo Runs for different Lagrangian Relaxation methods

MC	Relaxation on constraint (3)			Relaxation on constraint (4)			Speedup
	Gap	Objective Function Values	Runtime (CPU s)	Gap	Objective Function Values	Runtime (CPU s)	
1	0.015	59.021	0.065	0.003	59.627	0.040	1.628
2	0.010	59.281	0.064	0.004	59.599	0.045	1.422
3	0.013	59.100	0.054	0.003	59.618	0.038	1.413
4	0.011	59.219	0.053	0.002	59.664	0.033	1.613
5	0.011	59.209	0.053	0.001	59.764	0.031	1.692
6	0.010	59.249	0.053	0.002	59.699	0.032	1.664
7	0.009	59.335	0.070	0.002	59.719	0.031	2.279
8	0.013	59.124	0.057	0.005	59.513	0.034	1.700
9	0.012	59.184	0.053	0.008	59.347	0.034	1.563
10	0.010	59.290	0.049	0.003	59.632	0.036	1.344

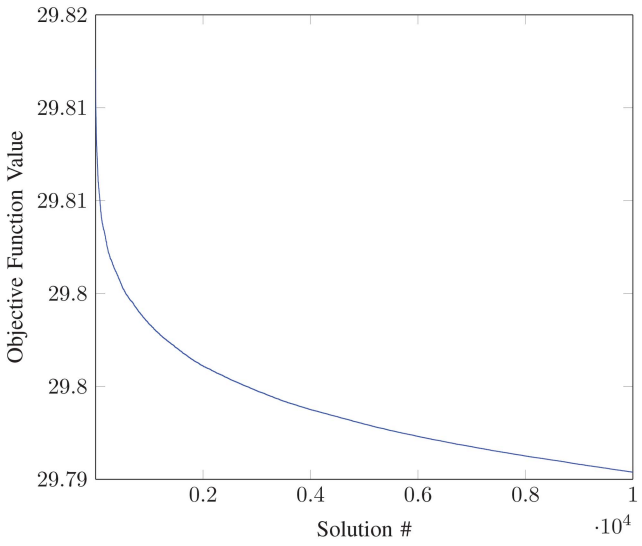


Fig. 4. Example objective function values for a tensor of dimension $30 \times 30 \times 8$ with values uniformly distributed on the interval $[0,1]$.

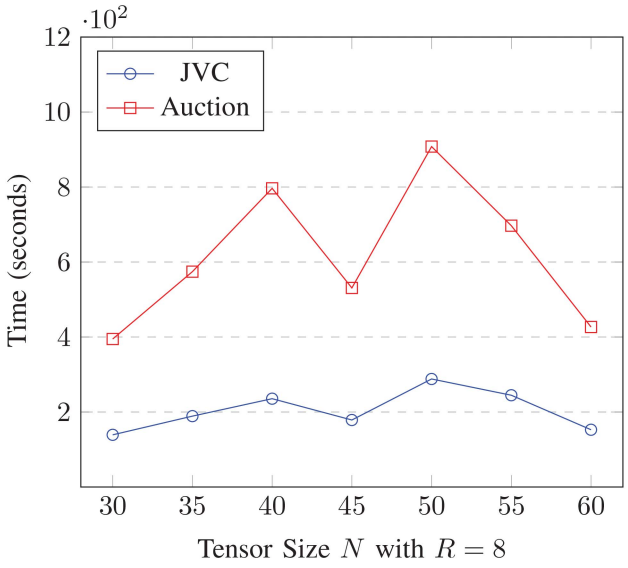


Fig. 5. The CPU runtime for the JVC and auction algorithms were compared as a function of varying tensor dimensions. The JVC algorithm consistently outperformed the auction algorithm.

solution remains. For these reasons, the remaining experiments used the m-best 3-D assignment algorithm with the relaxation of constraint (4) only.

6.2. JVC vs. Auction Algorithm

To measure and quantify which algorithms best solve the 2-D assignment and transportation problems within the 3-D assignment problem, we compared the runtimes of the 3-D assignment algorithm when using the JVC or the auction algorithms for the 2-D assignment problem, and Transauction or simplex-based transportation algorithms for the transportation problem. A tensor was generated with elements sampled from a uniform distribution in the interval $[0,1]$ for tensor sizes ranging from $30 \times 30 \times 8$ to $60 \times 60 \times 8$ with increments of $N = 5$. Any combination of the 2-D assignment algorithms with the transportation algorithms resulted in the same assignments and objective function values. An example of the objective function values of a sample tensor of dimension $30 \times 30 \times 8$ is shown in Fig. 4 when the algorithm was run to obtain the top 10^4 solutions. As shown in Fig. 4, even when 10^4 assignments were obtained, the maximum and minimum objective function values obtained from the assignment solutions had minimal variation and the difference was relatively small for all tensor dimensions tested; however, as shown in Fig. 5, the m-best 3-D assignment algorithm, which invoked the JVC algorithm was, on average, 3 times faster when compared to the case when the auction algorithm was used. The RELAX-IV algorithm was used to solve the transportation problem in this experiment.

6.3. Transportation vs. Transauction vs. RELAX-IV Algorithm

Similar tests were performed to evaluate the best algorithm to solve the transportation problem. Assuming that the JVC algorithm would be invoked to solve the 2-D assignment portion of the problem, Fig. 6

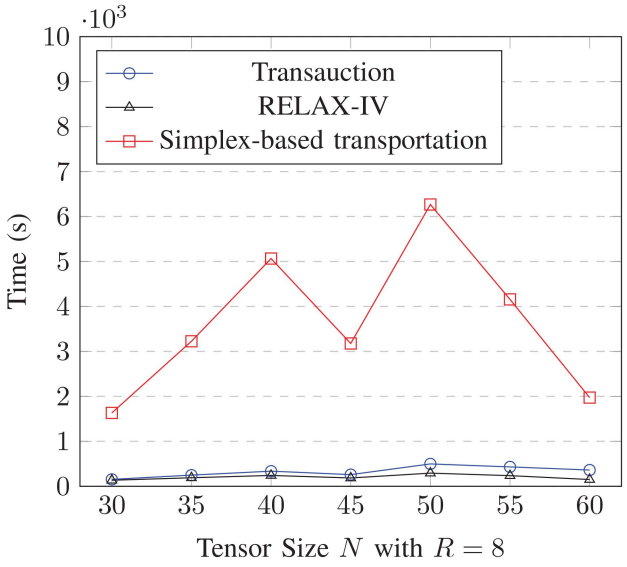


Fig. 6. The CPU runtime for the Transauction and simplex-based transportation algorithms were compared as a function of differing tensor dimensions. The Transauction algorithm remained relatively unaffected by the increase in the reward tensor size, while the transportation algorithm took orders of magnitude more time to find the same assignments.

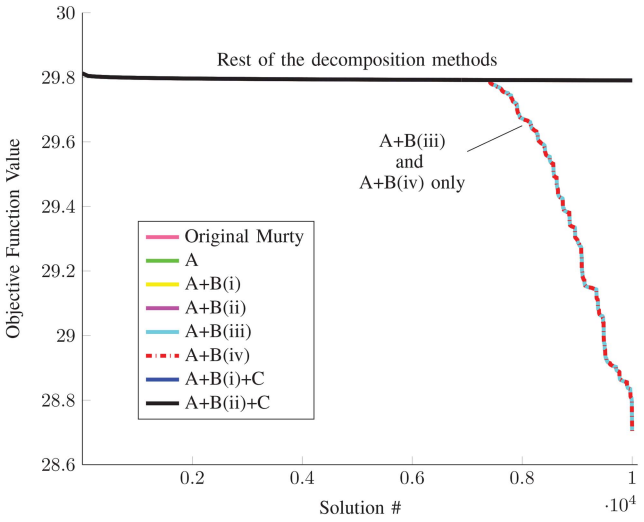


Fig. 7. Objective function values for tensor size $30 \times 30 \times 8$ with various optimization combinations.

demonstrates that the simplex-based transportation algorithm was significantly slower compared to both the Transauction and RELAX-IV algorithms. In general, the RELAX-IV algorithm had the fastest runtime speed. The maximum observed speedup of RELAX-IV in comparison to the simplex-based transportation and the Transauction algorithms was 21.4 and 2.4, respectively. Overall, the RELAX-IV algorithm dominated both the simplex-based transportation and the Transauction approaches to the transportation problem, on average solving it nearly 17 and 1.6 times faster, respectively. Based on these findings, the JVC and RELAX-IV algorithms were selected to solve the 2-D assignment and transportation problems within the m-best 3-D assignment problem, respectively, for the remaining computational experiments. We optimized the m-best 3-D assignment algorithm via the methods detailed in Section 4.

6.4. Solution quality evaluation for decomposition methods

A sample test tensor of dimension $30 \times 30 \times 8$ with elements uniformly distributed in the interval $[0, 1]$ was used to measure the solution quality and to compare the simulation runtimes of the m-best 3-D assignment algorithm when exploiting different combinations of the search space decomposition optimization methods. As shown in Fig. 7, the optimization combinations of A+B(iii) and A+B(iv) resulted in approximately a 10% reduction in the solution quality compared to the original Murty's proposed search space decomposition method. This is because the dual value in our problem setup did not serve as an accurate estimate of the initial upper bound. All other combinations of optimization methods were comparable to the Murty's search space decomposition. Therefore, optimization method combinations A+B(iii) and A+B(iv) were removed from the remaining tests.

6.5. Runtime comparison for decomposition methods

Similar tests were performed on all the remaining combinations of optimization methods for tensor sizes varying from $30 \times 30 \times 8$ to $60 \times 60 \times 8$ with an increment of $N = 5$. Table III shows the simulation runtime in CPU seconds for 10^4 solutions and for the various search space optimization combinations, given a sample test tensor for each incremented dimension. Methods A, A+B(i) and A+B(ii) on average ran 20%, 52% and 32% slower, respectively, within the m-best 3-D assignment algorithm, as compared to the original Murty's search space decomposition. As shown in Fig. 8, methods A+B(i)+C and A+B(ii)+C were able to obtain speedups with very minimum variation in the objective function values originally found by Murty's search space decomposition for all tensor sizes except that of dimension $60 \times 60 \times 8$. The reason for such a slow down is explained later in Section 6.6. Furthermore, combinations A+B(i)+C and A+B(ii)+C were able to obtain objective function values slightly better (higher) than the proposed method by Murty (on the order of 10^{-6}). This phenomenon is due to the Lagrangian relaxation algorithm's approximation of the 3-D assignment problem. The search space decomposition method is suboptimal when applied to the 3-D assignment problem (due to the suboptimal nature of the Lagrangian relaxation algorithm), and so from our analysis we observed that, through the particular optimization method combinations of A+B(i)+C and A+B(ii)+C, better feasible solutions were found. These methods were also significantly faster, offering an average of 2.1 and 2.4

TABLE III
Simulation runtime in CPU seconds for various combinations of decomposition methods

		Decomposition Methods				
Tensor Size	Original Murty	A	A+B(i)	A+B(ii)	A+B(i)+C	A+B(ii)+C
$30 \times 30 \times 8$	139.17	157.32	193.45	124.30	51.69	44.96
$35 \times 35 \times 8$	189.03	230.78	400.78	284.66	74.95	62.67
$40 \times 40 \times 8$	235.56	293.40	429.86	391.12	76.85	73.50
$45 \times 45 \times 8$	178.70	226.16	398.22	337.41	102.21	87.33
$50 \times 50 \times 8$	288.11	370.07	803.69	491.81	99.15	79.73
$55 \times 55 \times 8$	244.75	303.14	489.72	477.44	207.71	204.60
$60 \times 60 \times 8$	152.72	209.78	463.33	427.37	309.65	232.06

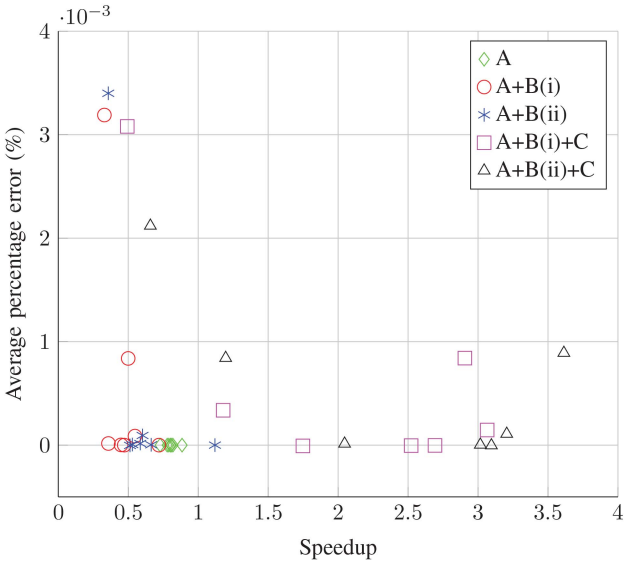


Fig. 8. Percentage error compared against the speedup for the combinations of optimization methods tested for all tensor sizes, varied from $30 \times 30 \times 8$ to $60 \times 60 \times 8$ with an increment of $N = 5$.

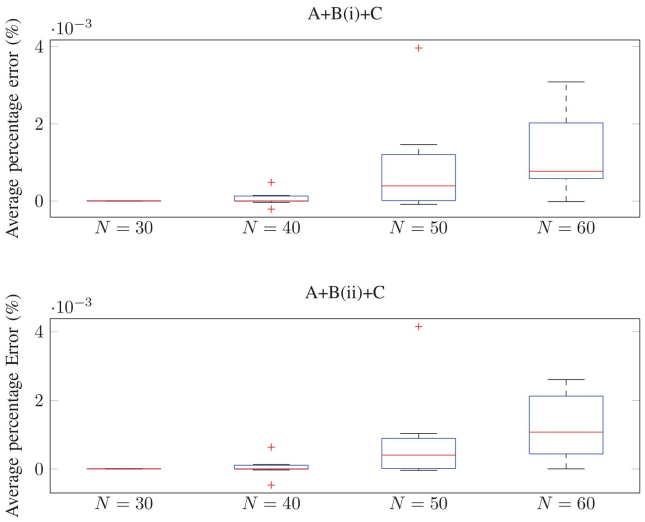


Fig. 9. Box plot for the average percentage error (as compared to the original Murty search space decomposition method) for the optimization method combinations A+B(i)+C and A+B(ii)+C.

speedup, respectively, as illustrated in Fig. 8. To investigate these combinations more thoroughly, Monte Carlo runs were performed on these two combinations only.

6.6. Scalability with N

To measure both the overall scalability and consistency, 10 Monte Carlo runs were performed for each tensor size varying from $30 \times 30 \times 8$ to $60 \times 60 \times 8$ in increments of $N = 10$ and using the two specific optimization method combinations of A+B(i)+C and A+B(ii)+C. Each test tensor was generated with elements uniformly distributed in the interval $[0, 1]$ and 10^4 solutions were obtained for each tensor. In each run, the objective function values and the simulation runtime, were monitored and compared against both combinations, as well as with respect to Murty's decomposition method. Fig. 9 shows the percentage error of the two optimization methods as compared to the original Murty's search space decomposition. The average percentage error increased with each increment in the tensor dimensions. Overall, the optimization method combination of A+B(ii)+C had a lower median compared to the combination of A+B(i)+C; however, the combination of variants A+B(i)+C had less variation with respect to the average percentage error. Table IV details the minimum, maximum, and average runtimes observed in CPU seconds for the Monte Carlo runs. The method A+B(ii)+C had the fastest runtime, as shown in Fig. 10, with an observed maximum average of 3.1 speedup over Murty's search space decomposition method, while having 2.14 speedup on average, when averaged over *all* Monte Carlo runs. The tensor of dimension $60 \times 60 \times 8$ resulted in a slow down of 32% and 25%, respectively, with optimizations A+B(i)+C and A+B(ii)+C.

In the fully optimized m-best 3-D assignment algorithm, there exists a tradeoff (when $N \approx 55$) in computation time between obtaining a feasible solution and the m-best optimization methods (e.g., partitioning or sorting), as seen in Fig. 10. The increase in dimension N does not necessarily mean an increase in the computation time of the 3-D assignment algorithm, since both optimization methods A+B(i)+C and A+B(ii)+C reduce the frequency of calling the 3-D assignment algorithm; however, partitioning and/or sorting the larger subproblems may become more difficult. A more favorable speedup may be observed if the algorithms were to be implemented in a fast object oriented-programming

TABLE IV
Minimum, maximum, and average runtimes in CPU seconds to obtain 10^4 solutions

$30 \times 30 \times 8$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	111.84	53.72	45.01
Mean	146.03	63.50	49.64
Max	195.41	76.68	53.55
$40 \times 40 \times 8$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	147.44	78.07	69.83
Mean	245.27	103.74	79.19
Max	284.48	128.17	92.61
$50 \times 50 \times 8$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	151.15	98.25	80.24
Mean	247.28	155.13	139.05
Max	320.79	271.20	243.48
$60 \times 60 \times 8$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	142.19	226.88	203.46
Mean	214.01	314.57	284.93
Max	294.43	586.10	542.90

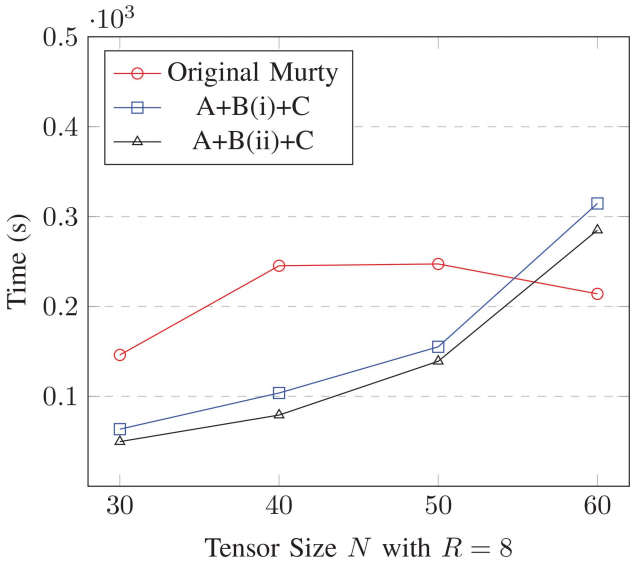


Fig. 10. The average CPU runtimes for 10 Monte Carlo runs for the two optimization method combinations tested.

language. Overall, for a $30 \times 30 \times 8$ tensor, the m-best 3-D assignment algorithm utilizing optimization method A+B(ii)+C took an average of 4.9 milliseonds to obtain a single solution to the 3-D assignment problem.

6.7. Scalability with R

As mentioned in Sections 2.1 and 3.1.4, the value of m_k should be such that $\sum_{k=1}^{R} m_k \geq N$. For problems where $m_k = R \geq N$, the 3-D assignment problem reduces to a 2-D assignment problem. We analyze the scalability of the algorithms with respect to incrementing R by performing 10 Monte Carlo runs for each increment and requesting 10^4 solutions with tensor size $N = 30$ and $R = 6, 10, 15, 20, 25, 29$ ($R = 30$ is omitted because, as mentioned earlier, this devolves into a 2-D assignment problem). Fig. 11 shows the relative percentage error of the two optimization methods, as compared to the original Murty's search space decomposition. The average percentage error is zero for $R \geq 10$, since the problem constraint (4) is less likely to be violated, and therefore, the duality gap is zero. The minimum, average and maximum runtimes are listed in Table V. As shown in Fig. 12, the speed of the original m-best 3-D assignment algorithm increases significantly with R. A maximum speedup of 10.8 and an average speedup of 7.5 were observed when comparing the optimization method A+B(ii)+C with the original m-best 3-D assignment algorithm. The total number of arcs input to the RELAX-IV algorithm is bounded above by R^3; hence increasing R has an exponential impact on the complexity of the problem solved by the algorithm and, in turn, the CPU runtime of the original m-best 3-D assignment. On the other hand, the optimized m-best 3-D assignment is able to reduce the need for the 3-D assignment routine invocation and, therefore, is able to obtain 10^4 solutions in a relatively short amount of time (< 5 minutes). The tensor of dimension $30 \times 30 \times 6$ had an increase in average CPU runtime compared to a tensor of dimension $30 \times 30 \times 10$ when considering the optimized methods A+B(i)+C and A+B(ii)+C. This is due to nonzero duality gap which impacts the partitioning procedure and subsequently requires more subprob-

TABLE V
Minimum, maximum, and average runtimes in CPU seconds to obtain 10^4 solutions

$30 \times 30 \times 6$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	83.33	78.24	62.51
Mean	103.77	93.48	79.00
Max	119.63	108.27	93.91
$30 \times 30 \times 10$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	161.03	56.47	42.03
Mean	232.45	65.78	51.33
Max	276.15	73.7	56.32
$30 \times 30 \times 15$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	477.91	87.00	68.20
Mean	640.89	102.21	75.71
Max	886.40	113.97	84.40
$30 \times 30 \times 20$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	747.44	133.35	108.31
Mean	1077.13	150.95	115.24
Max	1587.85	172.55	125.67
$30 \times 30 \times 25$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	1519.77	186.74	162.13
Mean	1902.02	218.85	176.63
Max	2507.46	230.53	187.83
$30 \times 30 \times 29$	Original Murty	A+B(i)+C	A+B(ii)+C
Min	1704.44	265.57	235.14
Mean	2572.39	298.28	249.87
Max	3343.91	330.11	268.99

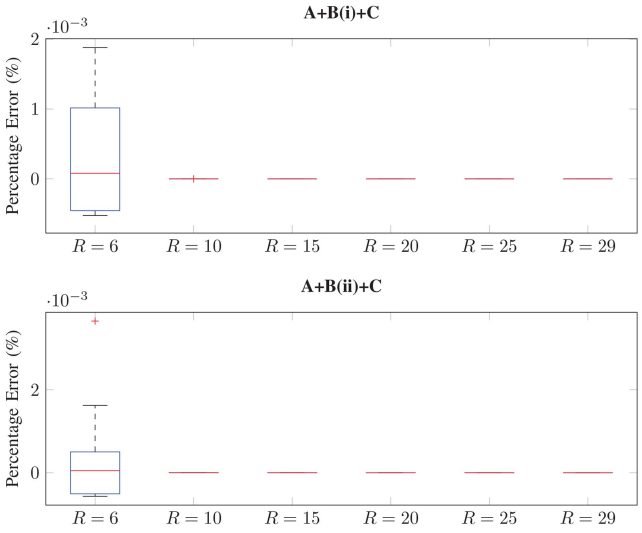


Fig. 11. Box plot for the average percentage error (as compared to the original Murty search space decomposition method) for the optimization method combinations A+B(i)+C and A+B(ii)+C, where $N = 30$.

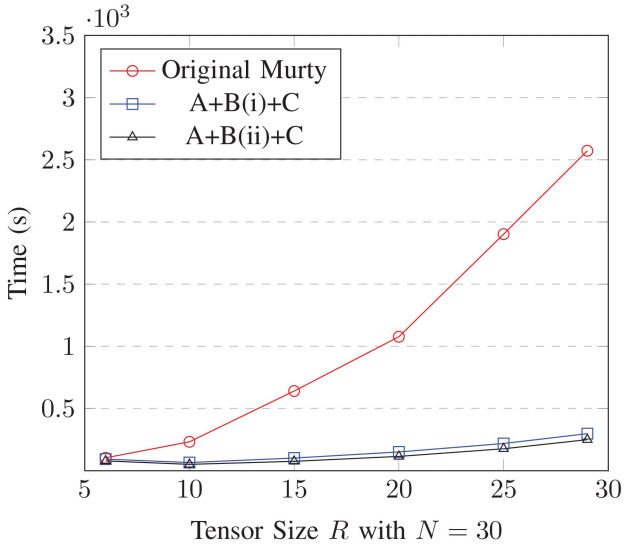


Fig. 12. The average CPU runtimes for 10 Monte Carlo runs with each increment of R for the two optimization method combinations tested.

lems to be solved before obtaining all m-best solutions. Intuitively, due to the nature of the problem, a tensor of dimension $30 \times 30 \times 6$ is more likely to violate constraint (4).

7. CONCLUSION

In this paper, we formulated a 3-D assignment problem and developed an efficient method to solve the problem, including 1) a rigorous mathematical formulation that is applicable to multiple domains; 2) a novel two-phase solution approach to obtain a large number of ranked solutions. The first phase of our solution approach involves partitioning the original problem space into a series of subproblems via Murty's m-best decomposition procedure, while in the second phase, we solve each of these subproblems using a combination of relaxed 2-D assignments through reformulation into either a transportation problem. The solution converges with a sufficiently small duality gap.

We compared the simulation runtime of the m-best 3-D assignment algorithm when relaxing either the assignment constraints or the transportation constraints. We also compared the performance of different combinations of 2-D assignment algorithms with a given transportation algorithm and found the combination of JVC and RELAX-IV algorithms, while relaxing the transportation constraints, to be the best performing combination when one is interested in solving the m-best 3-D assignment problem for a large number of solutions (in this paper, we were interested in obtaining 10^4 ranked solutions).

We also evaluated different decomposition methods and compared their scalability and consistency with Murty's search space decomposition. From our analysis, it can be seen that, when solving for a large number of solutions within a 3-D assignment problem, utilizing dual variable inheritance, tight upper bounds on the feasible reward, and partitioning in an optimized order offer the best performance, solving for all m-best solutions in a fraction of the time of the original Murty's decomposition method, with little to no sacrifice in solution quality. These optimizations offered a maximum speedup of 10.8 over Murty's search space decomposition. On average, it took 49.64 s to obtain 10^4 solutions for a tensor of dimension $30 \times 30 \times 8$ required for the nuclear fuel loading problem, which was well within the 10 minute time limit placed on the algorithm.

ACKNOWLEDGMENT

We would like to thank Dr. Michal Kvasnicka for bringing this unique variation of the m-best 3-D assignment problem to our attention and Dr. David Crouse for initial discussions. We thank them for their valuable comments and suggestions during the planning and development of this research work. We also would like to thank Antonio Frangioni for providing the C++ version of the RELAX-IV algorithm. This research was supported by the U.S. Office of Naval Research under contract #N00014-16-1-2036 and by the Department of Defense High Performance Computing Modernization Program under subproject contract #HPCM034125HQU.

REFERENCES

[1] W. P. Pierskalla
"The tri-substitution method for the three-dimensional assignment problem,"
CORS Journal, vol. 5, pp. 71–81, 1967.

[2] A. M. Frieze and J. Yadegar
"An Algorithm for Solving 3-Dimensional Assignment Problems with Application to Scheduling a Teaching Practice,"
Palgrave Macmillan Journal, vol. 32, no. 11, pp. 989–995, 1981.

[3] S. Deb, M. Yeddanapudi, K. Pattipaii, and Y. Bar-Shalom
"A generalized s-d assignment algorithm for multisensor-multitarget state estimation,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 33, no. 2 PART 1, pp. 523–538, 1997.

[4] R. M. Karp
"Reducibility among combinatorial problems,"
in *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[5] A. Frieze
"Complexity of a 3-dimensional assignment problem,"
European Journal of Operational Research, vol. 13, no. 2, pp. 161–164, 1983.

[6] C. H. Papadimitriou and K. Steiglitz
Combinatorial optimization: algorithms and complexity.
Courier Corporation, 1982.

[7] "Fuel loading completed at fangchenggang 2,"
http://www.world-nuclear-news.org/NN-Fuel-loading-completed-at-Fangchenggang-2-2505164.html, accessed: 2016-09-22.

[8] K. G. Murty
"An Algorithm for Ranking all the Assignments in Order of Increasing Cost,"
Operations Research, vol. 16, no. 3, pp. 682–687, 1968.

[9] H. W. Kuhn
"The hungarian method for the assignment problem,"
Naval research logistics quarterly, vol. 2, no. 1–2, pp. 83–97, 1955.

[10] R. Jonker and A. Volgenant
"A shortest augmenting path algorithm for dense and sparse linear assignment problems,"
Computing, vol. 38, no. 4, pp. 325–340, 1987.

[11] O. Drummond, D. A. Castanón, and M. Bellovin
"Comparison of 2-d assignment algorithms for sparse, rectangular, floating point, cost matrices,"
in *Proceedings of the SDI Panels on Tracking*, vol. 4, 1990, pp. 4–81.

[12] D. P. Bertsekas
"The auction algorithm: A distributed relaxation method for the assignment problem,"
Annals of operations research, vol. 14, no. 1, pp. 105–123, 1988.

[13] M. Balinski
"Signature methods for the assignment problem,"
Operations research, vol. 33, no. 3, pp. 527–536, 1985.

[14] P. Hansen and L. Kaufman
"A primal-dual algorithm for the three-dimensional assignment problem,"
Cahiers du CERO, vol. 15, pp. 327–336, 1973.

[15] E. Balas and M. J. Saltzman
"An algorithm for the three-index assignment problem,"
Operations Research, vol. 39, no. 1, pp. 150–161, 1991.

[16] R. Burkard and R. Rudolf
"Computational investigations on 3-dimensional axial assignment problems,"
Belgian Journal of Operations Research, Statistics and Computer Science, vol. 32, pp. 85–98, 1992.

[17] J. B. Mazzola and A. W. Neebe
"Resource-constrained assignment scheduling,"
Operations Research, vol. 34, no. 4, pp. 560–572, 1986.

[18] K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn
"A new relaxation algorithm and passive sensor data association,"
IEEE Transactions on Automatic Control, vol. 37, no. 2, pp. 198–213, 1992.

[19] A. P. Poore and N. Rijavec
"A lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking,"
SIAM Journal on Optimization, vol. 3, no. 3, pp. 544–563, 1993.

[20] A. B. Poore and X. Yan
"K-near optimal solutions to improve data association in multiframe processing,"
in *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*. International Society for Optics and Photonics, 1999, pp. 435–443.

[21] W. Hoffman and R. Pavley
"A method for the solution of the n th best path problem,"
Journal of the ACM (JACM), vol. 6, no. 4, pp. 506–514, 1959.

[22] E. de Queirós Vieira Martins, M. M. B. Pascoal and J. L. E. D. Santos
"Deviation algorithms for ranking shortest paths,"
International Journal of Foundations of Computer Science, vol. 10, no. 03, pp. 247–261, 1999.

[23] K. N. Androutsopoulos and K. G. Zografos
"Solving the k-shortest path problem with time windows in a time varying network,"
Operations Research Letters, vol. 36, no. 6, pp. 692–695, 2008.

[24] H. N. Gabow
"Two algorithms for generating weighted spanning trees in order,"
SIAM Journal on Computing, vol. 6, no. 1, pp. 139–150, 1977.

[25] H. W. Hamacher and G. Ruhe
"On spanning tree problems with multiple objectives,"
Annals of Operations Research, vol. 52, no. 4, pp. 209–230, 1994.

[26] Ž. Agić
"K-best spanning tree dependency parsing with verb valency lexicon reranking,"
in *24th International Conference on Computational Linguistics (COLING 2012)*, 2012.

[27] E. S. Van der Poort, M. Libura, G. Sierksma, and J. A. van der Veen
"Solving the k-best traveling salesman problem,"
Computers & operations research, vol. 26, no. 4, pp. 409–425, 1999.

[28] P. M. Camerini, L. Fratta, and F. Maffioli
"The k best spanning arborescences of a network,"
Networks, vol. 10, no. 2, pp. 91–109, 1980.

[29] I. J. Cox and M. L. Miller
"On Finding Ranked Assignments with Application to Multitarget Tracking and Motion Correspondence,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 31, no. 1, pp. 486–489, 1995.

[30] I. J. Cox, M. L. Miller, R. Danchick, and G. E. Newnam
"A comparison of two algorithms for determining ranked assignments with application to multitarget tracking and motion correspondence,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 33, no. 1, pp. 295–301, 1997.

[31] R. L. Popp
"Dynamically adaptable m-best 2-d assignment algorithm and multilevel parallelization,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 35, no. 4, pp. 1145–1160, 1999.

[32] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua
"Multiple object tracking using k-shortest paths optimization,"
IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 9, pp. 1806–1819, 2011.

[33] R. L. Popp, K. R. Pattipati, and Y. Bar-Shalom
"m-best S-D assignment algorithm with application to multitarget tracking,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 37, no. 1, pp. 22–39, 2001.

[34] E. L. Lawler
"A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem,"
Management science, vol. 18, no. 7, pp. 401–405, 1972.

[35] M. Pascoal, M. E. Captivo, and J. Clímaco
"A note on a new variant of murty's ranking assignments algorithm,"
Quarterly Journal of the Belgian, French and Italian Operations Research Societies, vol. 1, no. 3, pp. 243–255, 2003.

[36] M. L. Miller, H. S. Stone, and I. J. Cox
"Optimizing Murty's ranked assignment method,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 33, no. 3, pp. 851–862, 1997.

[37] H. W. Hamacher and M. Queyranne
"K best solutions to combinatorial optimization problems,"
Annals of Operations Research, vol. 4, no. 1, pp. 123–143, 1985.

[38] P. Carraresi and C. Sodini
"A binary enumeration tree to find k shortest paths,"
in *Proc. 7th Symp. operations research*, 1983, pp. 177–188.

[39] C. R. Chegireddy and H. W. Hamacher
"Algorithms for finding k-best perfect matchings,"
Discrete applied mathematics, vol. 18, no. 2, pp. 155–165, 1987.

[40] Y. Lin and K. Mouratidis
"Shortlisting top-k assignments,"
in *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*. ACM, 2013, p. 21.

[41] J. B. Mazzola and A. W. Neebe
"Resource-constrained assignment scheduling,"
Operations Research, vol. 34, no. 4, pp. 560–572, 1986.

[42] L. Zhang, D. Sidoti, K. R. Pattipati, and D. Castaỳỳn
"Approaches for solving m-best 3-dimensional dynamic scheduling problems for large m,"
in *Information Fusion (FUSION), 2016 19th International Conference on*. ISIF, 2016, pp. 53–58.

[43] D. P. Bertsekas and D. A. Castanon
"the Auction Algorithm for the Transportation Problem,"
Annals of Operations Research, vol. 20, pp. 67–96, 1989.

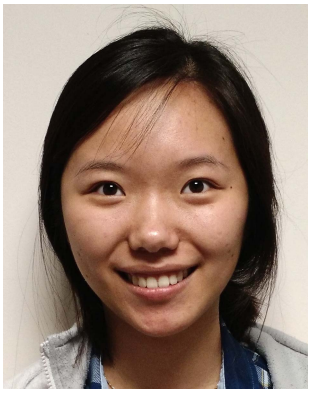
[44] J. B. Orlin, S. A. Plotkin, and É. Tardos
"Polynomial dual network simplex algorithms,"
Mathematical programming, vol. 60, no. 1, pp. 255–276, 1993.

[45] H. A. Taha
Operations research: an introduction.
Macmillan,, 1992.

[46] D. P. Bertsekas, P. Tseng et al.
RELAX-IV: A faster version of the RELAX code for solving minimum cost flow problems.
Massachusetts Institute of Technology, Laboratory for Information and Decision Systems Cambridge, MA, 1994.

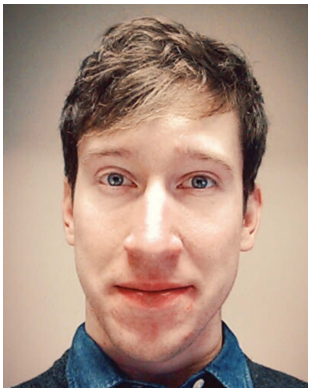
[47] A. Frangioni and A. Manca
"A computational study of cost reoptimization for min-cost flow problems,"
INFORMS Journal on Computing, vol. 18, no. 1, pp. 61–70, 2006.

[48] X. Han, H. Bui, S. Mandal, K. R. Pattipati, and D. L. Kleinman
"Optimization-based decision support software for a team-in-the-loop experiment: Asset package selection and planning,"
IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43, no. 2, pp. 237–251, 2013.



Lingyi Zhang received the B.S. degree in Electrical and Computer Engineering from the University of Connecticut, Storrs in 2014. She is currently pursuing the Ph.D. degree from the Department of Electrical and Computer Engineering at the same university.

Her current research interests include modeling dynamic and uncertain environments for asset allocation and path planning, context-aware decision support systems, and optimization-based techniques for mission planning and coordination.



David Sidoti received the B.S. and M.S. degrees in Electrical and Computer Engineering from the University of Connecticut, Storrs in 2011 and 2016. He is currently a Ph.D. candidate in the Electrical and Computer Engineering Department working under the advisement of Dr. Krishna R. Pattipati at the same university.

His work in dynamic resource management has resulted in numerous transitions to the real world (Naval Research Laboratory–Monterey, Joint Interagency Task Force–South, etc.). He has more than 30 research articles in peer-reviewed journals and international conference proceedings. He was an invited speaker at NATO's *Decision Support and Risk Assessment for Asset Planning 2015 Workshop*. His current interests include multi-objective algorithms for dynamic scheduling and resource management in weather-impacted environments.

Spandana Vallabhaneni received her B.S. degree in Computer Science and Engineering from GITAM University, Visakhapatnam, Andhra Pradesh, India in 2015, and the M.S. degree in Computer Science and Engineering from the University of Connecticut, Storrs, CT, USA, in 2017. She is currently a Software Developer with Cigna, Windsor, CT, USA.

Her current research interests include multi-objective path planning, optimization-based techniques, multiprocessing, and efficient algorithm development and implementation of dynamic resource management applications.

Krishna R. Pattipati (S'77–M'80–SM'91–F'95) received the B.Tech. degree in Electrical Engineering (with highest honors) from the Indian Institute of Technology, Kharagpur, India, in 1975 and the M.S. and Ph.D. degrees in Systems Engineering from the University of Connecticut, Storrs, in 1977 and 1980, respectively.

From 1980 to 1986, he was with ALPHATECH, Inc., Burlington, MA. Since 1986, he has been with the University of Connecticut, where he is currently the Board of Trustees Distinguished Professor and the UTC Professor in Systems Engineering in the department of Electrical and Computer Engineering, and was the founding director of the UTC Institute for Advanced Systems Engineering there during 2013–2015. His research interests are in the application of systems theory and optimization techniques to large-scale systems.

Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics Society as the Outstanding Young Engineer of 1984. He was also a recipient of the Centennial Key to the Future award. He has served as the Editor-in-Chief of the IEEE Transactions on Systems, Man, and Cybernetics: Part B–Cybernetics in 1998–2001, Vice-President for Technical Activities of the IEEE SMC Society in 1998–1999, and Vice-President for Conferences and Meetings of the IEEE SMC Society in 2000–2001. He was the co-recipient of the Andrew P. Sage Award for the Best SMC Transactions Paper for 1999, the Barry Carlton award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," the 2003 AAUP Research Excellence Award, and the 2005 School of Engineering Teaching Excellence Award at the University of Connecticut. He was also the recipient of the Best Technical Paper Awards at the 1985, 1990, 1994, 2002, 2004, 2005, and 2011 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences. He is recognized for his contributions to discrete-optimization algorithms for large-scale systems and team decision making.

David A. Castañón received his B.S. degree in Electrical Engineering from Tulane University in 1971, and his Ph.D. degree in Applied Mathematics from the Massachusetts Institute of Technology in 1976. He was chief scientist at ALPHATECH, Inc. in Burlington, MA until 1990, when he joined Boston University's Department of Electrical and Computer Engineering. He is a member of the IEEE Control Systems Society and the IEEE Signal Processing Society. He has served as general chair of the IEEE Conference on Decision and Control, and in numerous positions on the IEEE Control Systems Society, including Vicepresident for Finance, and President. Within IEEE, he served as Chair of the Society Review Committee, and the Conference Publications Committee. He also served on the Air Force Scientific Advisory Board, and received the Society's Distinguished Member Award. At Boston University, he served twice as interim Department Chair, and subsequently as Department Chair of the Department of Electrical and Computer Engineering. He also served as co-director of the Center for Information and Systems Engineering. He served Deputy Director of the NSF Engineering Research Center for Subsurface Sensing and Imaging, and is currently Associate Director of the Department of Homeland Securty's ALERT Center of Excellence on Explosives Detection and Mitigation. His research interests include stochastic control, estimation, game theory, optimization, and distributed computing, with applications to inverse problems, multitarget tracking, object recognition, sensor management, and security.

General Multivariate Polynomial Target Localization and Initial Estimation

DAVID FREDERIC CROUSE

Target localization and track initiation for any combination of measurement components can be performed by solving systems of simultaneous multivariate polynomials, utilizing cubature integration for covariance estimates. The solutions are approximate minimum mean squared error (MMSE) estimates. Measurement components can be monostatic/bistatic range as well as direction cosines, types of range-rate (Doppler), time delay of arrival (TDOA), and measured frequencies. Combinations of different measurement types are also considered. Results are compared to an approximate Cramér-Rao lower bound (CRLB). No previous work has addressed initial state estimation and localization in such a wide range of problems. Simulations include three specific problems that do not appear to be solved anywhere in the literature.

Manuscript received May 30, 2017; released for publication August 30, 2017.

Refereeing of this contribution was handled by Paolo Braca.

Author's address: Naval Research Laboratory, Attn: Code 5344, 4555 Overlook Ave., SW, Washington, DC 20375 (E-mail: david.crouse @nrl.navy.mil).

This research is supported by the Office of Naval Research through the Naval Research Laboratory (NRL) Base Program.

1. INTRODUCTION

In this paper, the general problem of starting a target track given a diverse mixture of monostatic/bistatic range, time delay of arrival (TDOA), direction of arrival (DOA), and various types of range-rate and/or frequency measurements from multiple sensors, which are not necessarily synchronized, is considered. The minimum amount of information needed to start a target track is a target position estimate and an associated covariance matrix. More desirable, however, is a state estimate consisting of a position and velocity vector as well as an associated covariance matrix.

No previous work has addressed initial state estimation and localization in such a wide range of problems. The track initiation approach in this paper express the problem as a problem of solving a system of simultaneous multivariate polynomials. This expands upon methods for localization in [66], [72] and Doppler-only track initiation in [46]. It is shown that a measurement-conversion approach to the problem is dramatically less computationally demanding than a least-squares solution. Moreover, the polynomial solutions are combined with a cubature integration technique to obtain approximate minimum mean squares error (MMSE) estimates with associated covariance matrices. This use of cubature integration for unbiased estimation and covariance matrix approximation fills a gap in the literature as most similar polynomial-based methods such as those in [46], [66], [72] do not consider covariance matrix estimation at all, making them unsuitable for use with many common tracking algorithms. This paper only considers the use of measurements after detection, as many practical networks will not always have sufficient bandwidth to send the raw or filtered antenna outputs to a fusion center. The work here can be viewed as a prerequisite for formulating a cost function for associating a diverse mixture of measurements between sensors.

The ability to start a track—to obtain a (hopefully unbiased) mean and a consistent covariance matrix or possibly a Gaussian mixture to represent the position or state (position, velocity, etc.) of a target given a set of measurements—is a necessary part of any target-tracking algorithm. Though active radar and sonar systems might be able to provide full 3D range and DOA measurements, a great many applications will offer a larger diversity of measurements.[1]

Additionally, the demand for starting tracks using diverse measurement types is likely to increase with the proliferation of digital multifunction radars. Many such systems can work in both active and passive modes, as

[1]For example, numerous countries are requiring that cell phone carriers be able to localize users to high accuracies and cell towers typically do not measure DOA. In the United States, the FCC is requiring a 50 m horizontal localization accuracy [30]. Similarly, aviation authorities in the United States [29] and the European Union [28] have been implementing passive tracking systems (via multilateration [DOA-only]) for civilian aircraft to augment transponder data from aircraft.

one can see by searching through radars in Jane's Online [37]. As digital radars are able to form increasingly many simultaneous receive beams, one would expect that a radar network would produce a greater diversity of measurements to use for target track initiation.

Much of the literature for target tracking using a diversity of measurements (not just range and DOA) tends to focus on using only one or two types of measurements and usually either try to find an explicit conversion for simultaneous measurements from the measurement domain into the domain of the state, or they perform some type of exact or approximate maximum likelihood (ML) or least squares (LS) estimation using various techniques depending on how difficult the problem is. For example, when considering Doppler-only tracking of an emitter, an explicit solution is available for the position and velocity of the emitter in a specific 2D scenario [71]. Additionally, a large number of 2D or 3D position and velocity estimation algorithms use a brute-force grid search of some type such as in [1], [16]–[18], [34], [40]–[42], [44] and (in Turkish) [43] or a grid search plus a refinement step as in [15], [39], [51]. In [47] an approximate global optimization is attempted using the (not optimal) Nelder-Mead algorithm.[2] A few authors tackle the problem using semidefinite programming in [50], [57] (which can be subject to finite-precision errors), while others try to integrate the initiation and detection directly into the tracking algorithms using particle filters, or Gaussian mixtures and linearization methods coupled with random finite set theory [33], [49], [54], [58], (such papers focus on the bistatic Doppler case and tend to be very complicated). A couple of papers solve systems of simultaneous multivariate polynomials as in [45], [46], [60], [61]. Similar approaches are used for other measurement types. This paper focusses on the final approach: utilizing simultaneous multivariate polynomials for target localization and/or target-state estimation, but with a diverse mixture of not necessarily simultaneous measurements.

The task of solving systems of simultaneous multivariate polynomials for localization and/or target-state estimation is not just limited to Doppler-only estimation, as in [45], [46], [61]. Such an approach is also suggested for solving time of arrival (TOA)-only, TDOA-only, and DOA-only localization problems in [66] as well as for solving TDOA-only estimation problems in [72].

The basic idea behind the track initiation techniques in this paper is similar to that in [26]: An estimator is obtained that provides an error-free estimate in the absence of noise (in this case by solving a system of simultaneous multivariate polynomials). Given noisy measurements, the mean and covariance matrix of the estimator (as obtained via cubature integration) is used as a mean and covariance matrix for target track initiation. The assumption is that the estimator is unbiased, so that the covariance matrix is an accurate representation of the mean squares error (MSE) of the estimate.

Section 2 describes the measurement models used in this paper. A basic assumption in this paper, and in the tracking literature in general, is that measurements are corrupted with Gaussian noise in the measurement domain (not in global Cartesian coordinates), where it is noted that all measurement types can be expressed as multivariate polynomials, sometimes with the help of additional variables. Section 3 then reviews basic aspects of cubature integration, which plays a pivotal role in obtaining unbiased estimates and covariance matrices for the estimates. Using cubature integration, one can evaluate the integrals necessary for determining the Cramér-Rao lower bound (CRLB), which is described in Section 4. The CRLB can often be used to determine whether combinations of measurements of different types from various sensors might be able to produce usable estimates (sufficiently accurate) without having to run Monte Carlo simulations.

Section 5 discusses algorithms for solving simultaneous multivariate polynomials. Available solvers are fast for many practical problems, and are even faster with parallelization. In some instances, one might use a "track-initiation" routine like in this paper to fuse passive measurements before passing them to a target-tracking algorithm, (i.e. via track function or by treating the fused measurements as a single "measurement"). If in such an instance, one finds that the track-initiation algorithm is slightly too slow for real-time use, it is worth noting that the delayed tracks/ measurements can be treated as out-of-sequence measurements, and methods for making use of such measurements in Kalman filters [3], [62], [74], interacting multiple model filters (IMMs) [2], and particle filters [11], [52], among many others, exist.

Section 6 discusses the use of 2D assignment algorithms for clustering estimates. This algorithm is needed to handle instances in Sections 7 and 1 where multiple solutions are obtained.

Section 7 combines the results of the previous sections to provide the efficient measurement-conversion algorithm that is the main topic of this paper. The algorithm tries to find the expected value of the target state (or just target position) given the minimal set of measurements needed to make the state or position observable. Multivariate polynomial expressions to handle many different scenarios are provided and simulations demonstrate the effectiveness of the estimation algorithm for three localization scenarios that are not currently present in the literature.

As in [66], it is also possible to perform track initiation with redundant information using ML/LS track-initiation routines. Section 1 discusses how this can

[2]It is worth noting that the dividing rectangles (DIRECT) algorithm of [38] can be used for globally optimal optimization without the same possibility of getting stuck at a suboptimal point as in the Nelder-Mead algorithm.

be done in general. However, the computational complexity is significantly higher than the measurement-conversion approach. Hope at real-time use of such methods necessitates more efficient multivariate polynomial solving routines than are used in this paper. The results are summarized in Section 9.

In [23] and [24] target track initiation given bistatic range and DOA measurements in without and with the effects of atmospheric refraction are considered. This paper solves track initiation and localization problems using a wider variety of measurements than are present in the literature, producing approximate MMSE estimates and covariance matrices. However, the effects of atmospheric refraction are neglected, though the results of this paper are still useful. When refractive effects matter, the estimates produced by the algorithms in this paper can be used as initial estimates in iterative algorithms with refraction, and the systems of equations derived in (5) can be used as start systems in homotopy algorithms for solving more difficult problems involving refraction. One type of homotopy solver is given in [26].

2. THE MEASUREMENT MODELS

The measurements considered in this paper are transformations of the target state $\mathbf{x}$. The measurements are assumed to be of the form

$$\hat{\mathbf{Z}} = h(\mathbf{x}, \mathbf{w}) \tag{1}$$

where $\mathbf{w}$ is a Gaussian random variable corrupting the measurement and h is a deterministic function. Though most results of this paper can be used with non-additive noise, in all simulation examples presented here, it is assumed that the measurement noise is additive, so (1) can be reduced to

$$\hat{\mathbf{Z}} = h(\mathbf{x}) + \mathbf{w} \tag{2}$$

If h is bijective (invertible) with respect to $\mathbf{x}$ and $\hat{\mathbf{Z}}$ such that there exists an inverse function $\mathbf{x} = h^{-1}(\hat{\mathbf{Z}}, \mathbf{w})$, then the measurement-conversion method of Section 7 should be used. Otherwise, the ML/LS method of Section 8 can be used. The cubature routine used by both is in Section 3.

An example of a bijective function would be the transformation of a Cartesian position to range and direction cosines. An example of a non-bijective function would be the transformation of a Cartesian position to three TDOAs. This transformation is non-bijective, because multiple solutions for a position can exist when given an arbitrary set of TDOAs. However, Section 7 provides *ad-hoc* approaches to be able to handle the case of having more than one or zero solutions.

In [23], expressions for the non-relativistic (Newtonian mechanics) bistatic range, range, rate, and DOA in terms of direction cosines ignoring atmospheric effects and target motion during the time it takes the signal to travel from the transmitter to the target and then to the receiver, are given and are used here. However, to be able to use non-simultaneous measurements for track initiation, a few minor changes are made.

Let $\mathbf{x}$ be the n-dimensional state of the target. The state is assumed to contain at least position components. Let $\mathbf{H}$ be a matrix such that $\mathbf{Hx}$ extracts the position components of the state and $\mathbf{H}_v$ a matrix that extracts any velocity components of the state (this is only needed when considering range-rate/frequency-shift measurements and a state with velocity components). For example, when considering tracking in 3D, and one has a state consisting of both position and velocity components with position components coming first, then $\mathbf{H} = [\mathbf{I}_{3,3}, \mathbf{0}_{3,3}]$ and $\mathbf{H}_v = [\mathbf{0}_{3,3}, \mathbf{I}_{3,3}]$ where $\mathbf{I}$ and $\mathbf{0}$ denote identity and zero matrices of the given dimensions.

For purposes of track initiation using non-simultaneous measurements, process noise in the dynamic model of the target shall be ignored. Additionally, only discrete-time dynamic models are considered. Let $\mathbf{F}$ be an $n \times n$ state transition matrix that propagates the target state from the time at which it is to be estimated to the time of a measurement. State transition matrices for some basic discrete-time dynamic models are provided in [4, Ch. 6]. The product $\mathbf{Fx}$ is taken to be the target state at the time of a measurement under consideration. For example, if one has a 3D state with position components before velocity components, then a standard linear propagation model just multiplies the velocity components by the time interval of propagation, T, adding them to the position components, so

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Note that T can be negative if one wishes to have a target estimate at the end of a batch rather than at the beginning. Define

$$\mathbf{F}_h \triangleq \mathbf{HF} \tag{4}$$

$$\mathbf{F}_v \triangleq \mathbf{H}_v\mathbf{F} \tag{5}$$

to simplify notation later on.

We shall now provide expressions for the components of measurements considered in this paper. Measurements may consist of one or more simultaneous components, all of which will be considered to be corrupted with (possibly correlated) additive Gaussian noise. A bistatic range measurement without noise is given by

$$r_B = \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\| + \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_2\| \tag{6}$$

the l_2 norm (the square root of the sum of the squares of the components of the argument) is given by $\|\ldots\|$,

$\mathbf{l}_1$ is the Cartesian location of the transmitter, and $\mathbf{l}_2$ is the location of the receiver. A monostatic (round-trip) range measurement would have $\mathbf{l}_1 = \mathbf{l}_2$. A TDOA measurement between two sensors is very similar. In such an instance, one can measure the time of arrival of a signal at two sensors, time just being distance divided by speed c. Thus, a TOA measurement at the ith sensor would be

$$\tau = \frac{1}{c}\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\| + \tau_0 \tag{7}$$

where τ_0 is the unknown transmission time of the emitter. A TDOA measurement thus cancels the unknown transmission time resulting in

$$\text{TDOA} = \frac{1}{c}(\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\| - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_2\|) \tag{8}$$

if Sensor 1 is used as the reference sensor.[3]

Another measurement type considered is DOA. Usually, as discussed in [23], it is preferable if direction cosine measurements are used (when considering measurements in 3D).[4] The local coordinate system of the receiver is aligned such that the third (unused) component is normal to the surface of the receiver. As it is assumed that the target is in "front" of the receiver, the third component is not needed (it is assumed positive). Let $\mathbf{l}$ be the 3×1 Cartesian location of the receiver. Let $\mathbf{M}$ be a matrix that rotates a position vector from the *global* coordinate system into the *local* coordinate system of the receiver. Define $\mathbf{H}_u$ to be a 2×3 matrix that extracts the $u - v$ components from the rotated matrix. For example, if as in [23], the z-axis is chosen to be the one pointing out from the radar, and the position components are in order (x, y, z), then

$$\mathbf{H}_u = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{9}$$

such that $\mathbf{H}_u\mathbf{u}$ is a vector containing the (x, y) components of the unit vector $\mathbf{u}$. Define

$$\mathbf{M}_u = \mathbf{H}_u\mathbf{M} \tag{10}$$

Thus, a $u - v$ direction cosine measurement $\mathbf{u}$ can be written as

$$\mathbf{u}_{uv} = \frac{1}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|}\mathbf{M}_u(\mathbf{F}_h\mathbf{x} - \mathbf{l}) \tag{11}$$

The final types of measurements that are to be considered are related to the range-rate/Doppler shift of the target. In [23], the non-relativistic (Newtonian mechanics) bistatic range-rate (ignoring atmospheric effects) of a target is given. For a receiver at $\mathbf{l}_1$ and transmitter at $\mathbf{l}_2$ traveling at velocities $\dot{\mathbf{l}}_1$ and $\dot{\mathbf{l}}_2$, the bistatic range rate is

$$\dot{r} = \left(\frac{\mathbf{F}_h\mathbf{x} - \mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\|}\right)'(\mathbf{F}_v\mathbf{x} - \dot{\mathbf{l}}_1) + \left(\frac{\mathbf{F}_h\mathbf{x} - \mathbf{l}_2}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_2\|}\right)'(\mathbf{F}_v\mathbf{x} - \dot{\mathbf{l}}_2). \tag{12}$$

If one is observing a range-rate measurement from a transmitter, then the target is the transmitter and the second half of the equation is eliminated, so the range rate reduces to

$$\dot{r} = \left(\frac{\mathbf{F}_h\mathbf{x} - \mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\|}\right)'(\mathbf{F}_v\mathbf{x} - \dot{\mathbf{l}}_1). \tag{13}$$

Now, however, that range-rate measurements must be derived from frequency offsets. If the transmitter broadcasts at a frequency of f_{Tx}, the actual frequency received due to Doppler shifts resulting from the given range rate is [31, Ch. 34-6][5]

$$f = \left(1 - \frac{\dot{r}}{c}\right) f_{\text{Tx}} \tag{14}$$

where c is the speed of propagation of the waves. However, if one is passively observing a target, then f_{Tx} will be unknown. Thus, we shall also consider using the ratio of received frequencies. That is, individual sensors cannot measure "Doppler" or "range rate" because the transmission frequency of the emitter is unknown. Rather, they measure the (average) received signal frequency. Based on (14), one can use the *ratio* of frequency measurements from multiple sensors to help localize a target. For example, if one gets frequency measurement at Sensors i and j, the frequency ratio is given in (15).[6]

$$\frac{f_i}{f_j} = \frac{1 - \frac{\dot{r}_i}{c}}{1 - \frac{\dot{r}_j}{c}} = \frac{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|(\mathbf{F}_h\mathbf{x} - \mathbf{l}_i)'(\mathbf{F}_v\mathbf{x} - \dot{\mathbf{l}}_i) - c\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\|\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\|(\mathbf{F}_h\mathbf{x} - \mathbf{l}_j)'(\mathbf{F}_v\mathbf{x} - \dot{\mathbf{l}}_j) - c\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\|\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|} \tag{15}$$

All of the above measurement models along with the frequency ratio are not polynomial even though the point of this paper is that one can use multivariate polynomial solving routines along with cubature integration

[3]Note that one does not always "measure" a TDOA. Often, two times are measured and then the difference is taken. In such an instance, if one assumes that each measurement is corrupted with Gaussian noise, then the difference will also be Gaussian distributed and thus can be treated as a single measurement.

[4]In 2D, this means that one component of a direction vector in the local coordinate system of the receiver is used. However, for simplicity, the discussion here only considers the 3D case.

[5]This is under Newtonian mechanics, meaning that the range rate is far lower than the speed of light.

[6]Note that one typically needs many digits of precision to make good use of frequency measurements. As another example, a radio emitter moving at a range rate of 10 m/s causes a 2 GHz transmission frequency to shift by less than 67 Hz. That is a shift of about $3.3 \times 10^{-6}\%$. Frequency ratios are generally never measured. Rather, individual frequencies are measured (and can be approximated as being corrupted by additive Gaussian noise—an approximation that would probably be very bad for the ratio).

to obtain mean and covariance estimates to start target tracks. However, all of them can be made polynomial either by manipulating the equations as in Section 7 for the measurement-conversion approach, or through the introduction of additional variables/equations, as is suggested in Section 8 for LS estimation. For example, a term of the form $\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_i\|$ can be replaced by a variable r_i while adding the equation

$$r_i^2 - \|\mathbf{F}_h\mathbf{x}-\mathbf{l}_i\|^2 = 0, \tag{16}$$

which is a polynomial equation. Similarly, a term of the form $1/\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_i\|$ can be placed by a variable $\tilde{r}_i$ where either

$$\tilde{r}_i^2\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_i\|^2 - 1 = 0 \tag{17}$$

or if r_i is already needed,

$$\tilde{r}_i r_i - 1 = 0. \tag{18}$$

3. CUBATURE INTEGRATION FOR MEAN AND COVARIANCE ESTIMATION

Given an estimator $\mathbf{g}(\mathbf{Z})$, we would like to obtain a mean and covariance matrix for use in a target-tracking algorithm. The basic idea in this paper is to use the numerically computed mean and covariance matrix of the estimator, assuming that the estimator is unbiased and the covariance matrix is an accurate representation of the actual target mean-squared error.

The mean and covariance matrix of an estimator $g(\mathbf{Z})$ are defined to be

$$\begin{aligned}\hat{\mathbf{x}} &\triangleq \mathrm{E}\{\mathbf{g}(\mathbf{Z}) \mid \hat{\mathbf{Z}}\}\\ &= \int_{\mathbf{Z}\in\mathbb{R}^{n_M}} \mathbf{g}(\mathbf{Z})p(\mathbf{Z}\mid\hat{\mathbf{Z}})d\mathbf{Z} \qquad (19)\\ \mathbf{P} &= \mathrm{E}\{(g(\mathbf{Z})-\hat{\mathbf{x}})(\mathbf{g}(\mathbf{Z})-\hat{\mathbf{x}})'\}\\ &= \int_{\mathbf{Z}\in\mathbb{R}^{n_M}} (\mathbf{g}(\mathbf{Z})-\hat{\mathbf{x}})(\mathbf{g}(\mathbf{Z})-\hat{\mathbf{x}})' p(\mathbf{Z}\mid\hat{\mathbf{Z}})d\mathbf{Z} \qquad (20)\end{aligned}$$

where the stacked set of measurements is n_M-dimensional. The measurement is $\hat{\mathbf{Z}}$ and $\mathbf{Z}$ represents the "noise-free" measurement. Under the assumption that the conditional probability distribution function (PDF) of the noise-free measurement is multivariate Gaussian distributed with mean $\hat{\mathbf{Z}}$ and covariance matrix $\mathbf{R}$ the integrals can be evaluated to a high precision with relatively low computational cost using cubature integration.

As discussed in [23] with respect to target tracking and measurement conversion, cubature integration is based on the fundamental theorem of Gaussian integration. Basically, the multivariate integral of a multivariate function $\mathbf{f}(\mathbf{x})$ of an n-dimensional variable $\mathbf{x}$, weighted by a Gaussian PDF with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ can be evaluated *exactly* as

$$\int_{\mathbf{x}\in\mathbf{R}^n} \mathbf{f}\{\mathbf{x}\}\mathcal{N}\{\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}\}d\mathbf{x} = \sum_{i=0}^{N_c}\omega_i\mathbf{g}\{\boldsymbol{\xi}_i\}, \tag{21}$$

$\mathcal{N}$ is the multivariate normal evaluated at the first parameter, with mean and covariance matrix given by the second and third parameters; and ω_i are cubature weights and $\boldsymbol{\xi}_i$ are n-dimensional cubature points. The assumption for perfect equality is that $\mathbf{f}(\mathbf{x})$ is a multivariate polynomial function and the cubature points and weights have been designed such that equality is possible for all functions up to a certain degree. As the solutions to simultaneous multivariate polynomials used in this paper are not polynomials themselves, the use of cubature integration is just an approximation. However, if a high enough degree of precision is chosen, it can be a very good approximation. The fundamental theorem of Gaussian integration basically says that a suitable set of cubature points and weights exists. Note that cubature formulae for weighting functions other than just the normal distribution also exist.

Many tables of cubature points can be found in [65] and an online collection of formulae is described in [19] and found at http://nines.cs.kuleuven.be/ecf/. Matlab code for generating cubature points and weights of various degrees and dimensionalities is provided as part of the Tracker Component Library online at https://github.com/DavidFCrouse/Tracker-Component-Library/. Formulae for cubature points used in the examples in this paper are given in Appendix B.

4. THE CRAMÉR-RAO LOWER BOUND

This paper offers multiple ways in which to assemble estimation algorithms. However, it is not always clear what combinations of measurement types might be best or whether one has sensors that can even achieve the needed accuracy to produce meaningful estimates given a diversity of measurement types. The CRLB provides a lower bound on the attainable accuracy of an unbiased estimator given a specific set of measurements, without simulation. The integral involved in the CRLB can be approximated using cubature integration as in Section 3. The use of cubature integration for CRLB computation is discussed in [23]. Here, we review the basic idea and provide expressions needed for each of the measurement types used.

As given in [4, Ch. 2.7.2], under certain regularity conditions (which are expanded in [5]), the CRLB for non-random vector parameters is given by the inverse of the Fisher information matrix. Specifically,

$$\mathrm{E}\{(\mathbf{x}_0-\hat{\mathbf{x}}\{\mathbf{Z}\})(\mathbf{x}_0-\hat{\mathbf{x}}\{\mathbf{Z}\})\} \geq \mathbf{J}^{-1} \tag{22}$$

where $\mathbf{x}_0$ is the true value of the quantity being estimated, $\mathbf{Z}$ is a set of observations, $\hat{\mathbf{x}}\{\mathbf{Z}\}$ is an estimator and $\mathbf{J}$ is the Fisher information matrix. The expected value is over the conditional PDF of $\mathbf{Z}$ given $\mathbf{x}_0$. As the inequality involves matrices, it has to be considered in terms of the eigenvalues of the matrices. The trace of the CRLB and be used as a lower bound on the MSE of an unbiased estimator. If the Fisher Information ma-

trix is singular, then the full state of the target is unobservable, even though individual components might be observable.

For non-random vector parameters, the Fisher information matrix is given by

$$\mathbf{J} = \mathrm{E}\{(\nabla_{\mathbf{x}} \ln p(\mathbf{Z} \mid \mathbf{x}))(\nabla_{\mathbf{x}} \ln p(\mathbf{Z} \mid \mathbf{x})) \mid \mathbf{Z}\}|_{x=x_0} \tag{23}$$

$$= \int_{\mathbf{Z}} (\nabla_{\mathbf{x}} \ln p(\mathbf{Z} \mid \mathbf{x}))(\nabla_{\mathbf{x}} \ln p(\mathbf{Z} \mid \mathbf{x}))|_{\mathbf{x}=\mathbf{x}_0} p(\mathbf{Z} \mid \mathbf{x}_0) d\mathbf{z} \tag{24}$$

where $\nabla_{\mathbf{x}}$ is the gradient operator with respect to $\mathbf{x}$. It creates a column vector of partial derivatives of its argument (here, $\ln p(\mathbf{Z} \mid \mathbf{x})$) with respect to the components of $\mathbf{x}$. Under the assumption that $p(\mathbf{Z} \mid \mathbf{x})$ is Gaussian, the results in Section 3 can be used to numerically solve the integrals.

Thus, to compute the CRLB for the examples in this paper, the remaining part that must be determined is $\nabla_{\mathbf{x}} \ln p(\mathbf{Z} \mid \mathbf{x})$. To simplify the analysis of the following sections, it is assumed that the components of the measurements used are independent. Under such an assumption, the Fisher information matrix of multiple measurements is just the sum of the Fisher information matrix matrices for each individual measurement. However, it can sometimes before more convenient to apply the independence assumption just to the gradient to get

$$\nabla_{\mathbf{x}} \ln p(\mathbf{Z} \mid \mathbf{x}) = \sum_{i=1}^{N} \nabla_{\mathbf{x}} \ln p(\mathbf{z}_i \mid \mathbf{x}) \tag{25}$$

where $\mathbf{z}_i$ is the ith measurement component. In the case of a direction-cosine DOA measurement in 3D, we shall take both components together. Thus, to consider the combined effect of different measurement types, we need but evaluate $\nabla_{\mathbf{x}} \ln p(\mathbf{z}_i \mid \mathbf{x})$ for each measurement type and sum them together. The rest of this section provides expressions for these gradients.

4.1. Range Measurements

Assuming a bistatic range measurement of the form in (6) corrupted with zero-mean Gaussian noise having variance σ_r^2, the logarithm of the likelihood is

$$\ln p(\mathbf{u}_{uv} \mid \mathbf{x}) = -\frac{1}{2} \ln(2\pi\sigma_r^2) - \frac{1}{2\sigma_r^2}(r_B - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\| - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_2\|)^2 \tag{26}$$

and thus the gradient necessary for the CRLB is

$$\nabla_{\mathbf{x}} \ln p(r_B \mid \mathbf{x}) = -\frac{1}{\sigma_r^2}(r_B - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\| - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_2\|) \cdot \left(\frac{\mathbf{F}_h'\mathbf{l}_1 - \mathbf{F}_h'\mathbf{F}_h\mathbf{x}}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\|} + \frac{\mathbf{F}_h'\mathbf{l}_2 - \mathbf{F}_h'\mathbf{F}_h\mathbf{x}}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_2\|} \right) \tag{27}$$

4.2. TDOA Measurements

Assuming a TDOA measurement of the form in (8) corrupted with zero-mean Gaussian noise having variance σ_{TDOA}^2, the logarithm of the likelihood is

$$\ln p(\mathbf{u}_{uv} \mid \mathbf{x}) = -\frac{1}{2} \ln(2\pi\sigma_{\mathrm{TDOA}}^2) - \frac{1}{2\sigma_{\mathrm{TDOA}}^2} \left(\mathrm{TDOA} - \frac{1}{c}(\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\| - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_2\|) \right)^2 \tag{28}$$

and thus the gradient necessary for the CRLB is

$$\nabla_{\mathbf{x}} \ln p(\mathrm{TDOA} \mid \mathbf{x}) = -\frac{1}{c\sigma_{\mathrm{TDOA}}^2} \left(\mathrm{TDOA} - \frac{1}{c}(\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\| - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_2\|) \right) \cdot \left(\frac{\mathbf{F}_h'\mathbf{l}_1 - \mathbf{F}_h'\mathbf{F}_h\mathbf{x}}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\|} - \frac{\mathbf{F}_h'\mathbf{l}_2 - \mathbf{F}_h'\mathbf{F}_h\mathbf{x}}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_2\|} \right) \tag{29}$$

4.3. DOA Measurements

Assuming a DOA measurement of the form in (11), which consists of two components corrupted with zero-mean Gaussian noise having (symmetric) covariance matrix $\mathbf{R}_{uv}$, the logarithm of the likelihood is given in (30).

$$\ln p(\mathbf{u}_{uv} \mid \mathbf{x}) = -\frac{1}{2} \ln(|2\pi\mathbf{R}_{uv}|) - \frac{1}{2} \left(\mathbf{u}_{uv} - \frac{1}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|} \mathbf{M}_u(\mathbf{F}_h\mathbf{x} - \mathbf{l}) \right)' \cdot \mathbf{R}_{uv}^{-1} \left(\mathbf{u}_{uv} - \frac{1}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|} \mathbf{M}_u(\mathbf{F}_h\mathbf{x} - \mathbf{l}) \right) \tag{30}$$

$$= -\frac{1}{2} \ln(|2\pi\mathbf{R}_{uv}|) - \frac{1}{2}\mathbf{u}_{uv}'\mathbf{R}_{uv}^{-1}\mathbf{u}_{uv} - \frac{1}{2\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|^2}(\mathbf{x}'\mathbf{F}_h' - \mathbf{l}')\mathbf{M}_u'\mathbf{R}_{uv}^{-1}\mathbf{M}_u(\mathbf{F}_h\mathbf{x} - \mathbf{l}) + \frac{1}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|}(\mathbf{x}'\mathbf{F}_h' - \mathbf{l}')\mathbf{M}_u'\mathbf{R}_{uv}^{-1}\mathbf{u}_{uv} \tag{31}$$

Thus, the gradient necessary for the CRLB is

$$\nabla_{\mathbf{x}} \ln p(\mathbf{u}_{uv} \mid \mathbf{x}) = \frac{(\mathbf{x}'\mathbf{F}_h' - \mathbf{l}')\mathbf{M}_u'\mathbf{R}_{uv}^{-1}\mathbf{M}_u(\mathbf{F}_h\mathbf{x} - \mathbf{l})}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|^4}(\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}) - \frac{1}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|^2}\mathbf{F}_h'\mathbf{M}_u'\mathbf{R}_{uv}^{-1}\mathbf{M}_u(\mathbf{F}_h\mathbf{x} - \mathbf{l}) \tag{32}$$

$$- \frac{(\mathbf{x}'\mathbf{F}_h' - \mathbf{l}')\mathbf{M}_u'\mathbf{R}_{uv}^{-1}\mathbf{u}_{uv}}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|^3}(\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}) + \frac{1}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|}\mathbf{F}_h'\mathbf{M}_u'\mathbf{R}_{uv}^{-1}\mathbf{u}_{uv} \tag{33}$$

4.4. Emitter Range-Rate Measurements

Assuming an emitter range-rate measurement of the form in (13) corrupted with zero-mean Gaussian noise having variance $\sigma_{\dot{r}}^2$, the logarithm of the likelihood assuming that the emitter is stationary (the only case considered in the examples) is

$$\ln p(\dot{r} \mid \mathbf{x}) = -\frac{1}{2}\ln(2\pi\sigma_{\dot{r}}^2) - \frac{1}{2\sigma_{\dot{r}}^2}\left(\dot{r} + \frac{(\mathbf{F}_h\mathbf{x}-\mathbf{l}_1)'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)^2. \tag{34}$$

Thus, the gradient necessary for the CRLB is

$$\nabla_{\mathbf{x}} \ln p(\dot{r} \mid \mathbf{x}) = \frac{1}{\sigma_{\dot{r}}^2}\left(\dot{r} + \frac{(\mathbf{F}_h\mathbf{x}-\mathbf{l}_1)'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right) \cdot \left(\frac{\mathbf{F}_h'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|} + \frac{(\mathbf{F}_h\mathbf{x}-\mathbf{l}_1)'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|^3}(\mathbf{F}_h'\mathbf{l}_1 - \mathbf{F}_h'\mathbf{F}_h\mathbf{x})\right). \tag{35}$$

4.5. Frequency (Ratio) Measurements

In this instance, the measurement components are frequencies (e.g. single frequency tones or the center frequencies of more complex signals) measured from moving sensors, assuming a stationary emitter. Each receiver will measure a different tone due to the Doppler shift associated with its movement with respect to the emitter.

Even though (15) provides an expression for the ratio of two such frequencies, and Section 7.3.4 shows how to create multivariate polynomial equations that (with others) can be used to localize such an emitter. It is easier to compute the CRLB from the individual frequencies, which will presumably be the actual measurements, rather than the ratio. Similarly, the cubature integration in Section 3 when used to estimate the moments of the converted measurements would have to be done using the actual frequencies rather than the ratio as it cannot be assumed that the ratio is Gaussian distributed.

However, if the individual frequencies are used, then the unknown transmit frequency f_{Tx} must be added to the "state" for purposes of estimating the CRLB. Thus, in this section, we provide the gradient with respect to $\mathbf{x}$, the typical state one would think of, as well as with respect to f_{Tx}, which must be included just for purposes of computing the CRLB.

Assuming a frequency measurement of the form in (14) corrupted with zero-mean Gaussian noise having variance σ_f^2, the logarithm of the likelihood assuming that the emitter is stationary is

$$\ln p(f \mid \mathbf{x}, f_{\text{Tx}}) = -\frac{1}{2}\ln(2\pi\sigma_f^2) - \frac{1}{2\sigma_f^2}\left(f - \left(1 + \frac{1}{c}\left(\frac{\mathbf{F}_h\mathbf{x}-\mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)'\dot{\mathbf{l}}_1\right) f_{\text{Tx}}\right)^2 \tag{36}$$

Thus, the gradient with respect to the state $\mathbf{x}$ and the partial derivative with respect to the unknown transmission frequency f_{Tx}, necessary for the CRLB, are

$$\nabla_{\mathbf{x}} \ln p(f \mid \mathbf{x}, f_{\text{Tx}}) = \frac{f_{\text{Tx}}}{c\sigma_f^2}\left(f - \left(1 + \frac{1}{c}\left(\frac{\mathbf{F}_h\mathbf{x}-\mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)'\dot{\mathbf{l}}_1\right) f_{\text{Tx}}\right) \cdot \left(\frac{\mathbf{F}_h'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|} + \frac{(\mathbf{F}_h\mathbf{x}-\mathbf{l}_1)'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|^3}(\mathbf{F}_h'\mathbf{l}_1 - \mathbf{F}_h'\mathbf{F}_h\mathbf{x})\right) \tag{37}$$

$$\frac{\partial}{\partial f_{\text{Tx}}} \ln p(f \mid \mathbf{x}, f_{\text{Tx}}) = \frac{1}{\sigma_f^2}\left(f - \left(1 + \frac{1}{c}\left(\frac{\mathbf{F}_h\mathbf{x}-\mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)'\dot{\mathbf{l}}_1\right) f_{\text{Tx}}\right) \cdot \left(1 + \frac{1}{c}\left(\frac{\mathbf{F}_h\mathbf{x}-\mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)'\dot{\mathbf{l}}_1\right) \tag{38}$$

5. SOLVING SIMULTANEOUS MULTIVARIATE POLYNOMIALS

A key step in the track-initiation routines in this paper is the solution of simultaneous multivariate polynomial systems. Here, we are only interested in systems where all solutions are "zero dimensional." Zero dimensional solutions are individual points. The alternative, "positive-dimensional" solutions, represent curves, surfaces, volumes, etc., and thus represent an infinite number of solutions. For example, the bivariate system of polynomials

$$0 = -1880 + 1176x - 240x^2 + 16x^3 - 1780y + 1156xy - 240x^2y + 16x^3y - 420y^2 + 284xy^2 - 60x^2y^2 + 4x^3y^2 \tag{39}$$

$$0 = 1974 - 1780x + 578x^2 - 80x^3 + 4x^4 + 882y - 840xy + 284x^2y - 40x^3y + 2x^4y \tag{40}$$

has two zero-dimensional solutions, $(3,-2)$ and $(7,-2)$, as well as one positive dimensional solution,

$$y = \frac{-47 + 20x - 2x^2}{21 - 10x + x^2} \tag{41}$$

Though positive dimensional solutions can be numerically investigated using tools such as Bertini_real [12], available from http://bertinireal.com, which relies on Bertini [8], available from https://bertini.nd.edu, only unique solutions are of interest here.

Many believe that solving even moderate-sized multivariate polynomial systems is too computationally demanding. This is true if one uses the oldest algorithms, which are based on the computation of Gröbner bases,

which arise in the field of abstract algebra.[7] However, the field of numerical algebraic geometry has produced better techniques.

The field of algebraic geometry studies the zeros of multivariate polynomials. Of interest to engineers is the related field of numerical algebraic geometry, which studies numerical techniques for finding the roots of multivariate polynomial systems.[8] A number of methods of solving simultaneous multivariate polynomials exist in the literature. Notable ones are:

1) Utilizing a standard homotopy algorithm in the complex domain. Such an algorithm uses a start polynomial system $g(\mathbf{x})$ with known zeros to solve the desired polynomial system $f(\mathbf{x})$. It does this by writing differential equations in terms of a "helper" variable of some sort. For example, a linear homotopy will often have the form

$$0 = \gamma(1-\lambda)g(\mathbf{x}) + \lambda f(\mathbf{x}) \tag{42}$$

where λ starts at 0 and is integrated to 1 (sometimes $1-\lambda$ is used instead of λ) and γ is a random complex number. The choice of the start system as well as the type of homotopy and the ability to deal with finite-precision errors and "path jumping" makes this a difficult problem. Algorithms to solve this problem tend to be broken into two categories: practical numerical algorithms, which tend to use adaptive step sizes, and primarily theoretical algorithms that guarantee global convergence with a hard bound on the number of steps, but which often fail in practice as their proofs assume unlimited numerical precision. Notable open-source, practical algorithms include
   - Bertini, which is online at https://bertini.nd.edu and is well-documented in the book [8]. An interface for Matlab is described in [7] and is available for download from http://www.mathworks.com/matlabcentral/fileexchange/48536-bertinilab. Version 1.5 of the software is written in C. Version 2 is under development in C++. Version 1.5 of the program is used in the examples in this paper.
   - PHCpack, which is available online at http://homepages.math.uic.edu/~jan/download.html is documented in [67] and also online at http://homepages.math.uic.edu/~jan/phcpack_doc_html/index.html and which includes a Matlab interface, which is documented in [32]. The software is written in Ada. As Ada is not commonly used, it is worth noting that free and commercial Ada compilers can be obtained from http://libre.adacore.com. Version 2.4.14 of the program is used in the examples in this paper.
   - HomLab, which is online at https://www3.nd.edu/~cwample1/HomLab/main.html. The solver is not entirely free as the terms of the license require that one purchase the associated book [64]. The software is written in Matlab. It is not used in the examples in this paper.

   The less practical but more theoretically "nice" homotopy algorithms for solving simultaneous multivariate polynomials are "certified" algorithms. Being certified assures convergence to an exact solution assuming infinite precision algebra. The first such algorithm is described in [10], though one might wish to consult the book [14] to fully understand the requisite math used.
2) A different type of homotopy algorithm is the "probability 1" homotopy. This type of algorithm can be used both for polynomial and non-polynomial systems. It is used for difficult polynomial and non-polynomial target-track-initiation problems in [26], though the results there must be considered as heuristic as the implementation in [26] does not trace its paths in the complex domain and the proofs for the "probability 1" success of the algorithm are only valid in the complex domain.[9] Probability 1 homotopy algorithms for solving polynomials were not used in this paper as later literature indicates that it is inferior to newer homotopy algorithms.
3) Transforming the problem of solving multivariate polynomials into a generalized eigenvalue problem through the use of Sylvester or Macaulay matrices to get all real and complex solutions. This method is used in [72] for TDOA-only estimation. A very understandable description of such an approach is in [27].[10]
4) All of the previous algorithms find all real and complex solutions to simultaneous multivariate polynomials. An alternative method is the Khovanskii-Rolle continuation for real solutions described in [9], for which an implementation in Maple is available online at https://www3.nd.edu/~dbates1/Rolle/. The algorithm is appealing as it could, in theory, be significantly faster than methods that must search the entire real and complex space. However, the cur-

[7] An introduction to Gröbner bases is given in [13] and notes that examples of Gröbner basis computations for polynomials in three or four variables may fail to terminate in a reasonable amount of time or may exceed the available memory of a computer.

[8] An introduction to the field is given in [63]. An introduction to some of the most common practical and theoretical aspects of the area can be found in the book documenting the Bertini solver [8] and in Part III of [14].

[9] Implementations of such probability 1 algorithms for polynomials are described in [69], [70], [73] and the source code associated with those papers is available in Fortran at http://netlib.org/toms/.

[10] A more efficient implementation of such an approach, "A Maple/Matlab/C Resultant-Based Solver," is documented in [68] and available online at http://gamma.cs.unc.edu/MARS/. The algorithm is implemented in a combination of Maple, Matlab and C. The algorithm depends on being able to accurately determine the rank of very large sparse matrices, which can pose finite-precision issues. This algorithm was not considered in the examples in this paper.

rent version of the algorithm can only solve bivariate polynomial systems.

Due to the time and difficulty in implementing a numerically stable simultaneous multivariate polynomials solver, this paper simply uses Bertini, PHCpack, and the certified polynomial solver in Macaulay2.[11] For the simulations in this paper, all of the programs were (inefficiently) called from Matlab using scripts that wrote input to disk, had the programs read the files, and wrote their results to disk, which were then read back into Matlab.

When using the solver algorithms, it will often become necessary to shift and scale the equations so that various convergence criteria become relatively scale invariant. In the simulations implemented for this paper, this was done in three ways:

1) The mean sensor location was subtracted from all sensor locations (and then added back to the estimate in the end). This increases precision if, for example, one is tracking in a coordinate system with the center of the Earth as the origin, but all sensors are within a small area on the surface.
2) The sensor locations are then scaled so that the most distant sensor location has magnitude 1. Additionally, the speed of propagation c used for TDOA and frequency measurements is scaled accordingly. Range measurements are also scaled. In the end before removing the mean sensor location shift, the entire state vector must be scaled back.
3) After the equations are formed, the coefficients for the equations are normalized so that the magnitude of the largest coefficient is 1.

The complexity of these algorithms is related to the number of zeros of the polynomial system that must be found, which is related to Bézout's theorem, given in [14, Ch. 16.5]. Assuming that a polynomial system has a finite number of solutions (no positive dimensional solutions) Bézout's theorem says that the maximum possible number of solutions equals the total degree of the system. The degree of each equation in a polynomial systems equals the degree of the highest monomial term. Given a term consisting of a constant times the product of variables of the form $x_1^{a_1}x_2^{a_2}\dots x_n^{a_n}$ with $a_1, a_2, \dots, a_n > 0$, the degree of the term is the product $a_1 a_2 \dots a_n$. This is the same definition used for the degree or order of a polynomial equation when choosing a cubature formula as in Section 3. The measurement-conversion-based estimation method of Section 7 is significantly faster than the ML based approach in Section 8, because Bézout's number is much lower for equivalent problems.[12]

In some simple instances, explicit solutions are available, though there can sometimes be a minor loss in precision compared to using a homotopy solver. Appendix A demonstrates this by providing a direct method of solving simultaneous bivariate equations and showing that one loses a few digits of precision with the direct solution versus using PHCpack, which is fixed precision.

Finally, though the ML/LS solutions in Section 8 are often too complex to solve in real time using standard multivariate polynomial solvers, it is worth noting that one can often find the *value* of the minimum, or an approximation of it, very efficiently using semidefinite programming. This is because the LS cost function can be factored into a sum-of-squares polynomial, as described in [56]. Related details on the theory are given in [55].[13]

6. ASSIGNMENT OF MULTIPLE SOLUTIONS FOR CLUSTERING

The cubature method of obtaining a mean and covariance matrix given multiple measurements outlined in Section 3 is ignorant of the case where a finite number of solutions > 1 to the localization or state estimation problem is possible. That is, when the evaluation of $\mathbf{g}\{\boldsymbol{\xi}_i\}$ in (21) produces multiple solutions for each cubature point. In such an instance, if $\mathbf{g}\{\boldsymbol{\xi}_i\}$ always produces, for example, two solutions, then In such an instance, one might want to start a track with two hypotheses, one at each location. However, one cannot put two solutions into the integral (21) to obtain mean and covariance components necessary to initialize a Kalman filter-like algorithm. However, if one could associated which of the solutions for each cubature point comes from the same "hypothesis," then one could evaluate the integrals twice, once with each set of converted points.

This section presents a clustering algorithm for associating converted cubature points. The primary contribution of this section is the formulation of an appropriate cost function that allows the clustering problem to be expressed as a multidimensional assignment problem, which can be solved using already existing solvers.

A problem arises if one does not always find the same number of solutions for every converted cubature point. We shall assume that such an occurrence is unlikely and will simply discard clusters that do not have

[11] All three polynomial solving routines used are open source, though the terms of their licenses do not necessarily allow one to transfer the desired subroutines into commercial radar software. However, all of the solvers can be run as separate processes without any copyleft clauses being forced into the calling program.

[12] In many instances, it is possible to accelerate the speed of the solvers by solving a similar problem once, and then using the results to create the best possible start system for the homotopy solver that can be reused for many variants of the problem in question. That is, many algorithms can take advantage of the so-called "cheater's homotopy," described in [48]. The tool Paramotopy [6], which is available online at http://www.paramotopy.com, can help parallelize the use of a cheater's homotopy to solve multiple variants of a particular problem.

[13] Free software for performing such an optimization is SOSTOOLS, which can be downloaded from http://www.cds.caltech.edu/sostools/.

a full set of points at the end of the assignment. In the simulations, the failure rate for when the true target is not in any cluster is computed to show that this is a rare event.

the results. To simplify things, we shall assume that the cubature points are sorted such that $\boldsymbol{\xi}_1$ is the cubature point that produced the most solutions. The two-dimensional cost matrix is given in (44).

$$\mathbf{C} \triangleq \left[\begin{array}{cccccccc} \overbrace{d((1,1),(2,1))} & \cdots & \overbrace{d((1,1),(2,N_{C,2}))} & \overbrace{d((1,1),\emptyset)} & \overbrace{\infty} & \cdots & \overbrace{\infty} \\ d((1,2),(2,1)) & \cdots & d((1,2),(2,N_{C,2})) & \infty & d((2,1),\emptyset) & \cdots & \infty \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ d((1,N_{C,1}),(2,1)) & \cdots & d((1,N_{C,1}),(2,N_{C,2})) & \infty & \infty & \cdots & d((N_{C,1}),1),\emptyset) \end{array}\right] \quad (44)$$

Assignment Costs / Non-Assignment Costs

To associate the solutions of converted cubature points, we have to establish a cost function. One would like to cluster solutions of converted cubature points that are "close" together. However, one cannot simply take the magnitude of the difference of the computed points, which will generally represent target states, because the l_2 norm of the difference will mix position components having incompatible units (e.g. mix position and velocity components). Thus, the solution chosen here is to only perform the clustering using the position components. Such an approach has also been used in minimum mean-optimal subpattern assignment (MMOSPA) algorithms for the display of uncertain targets as in [21], [22].

Assume that we are given a set of N_C cubature points $\boldsymbol{\xi}_i$ with associated weights w_i so as to perform numerical integration as in (21). Define $\tilde{\boldsymbol{\xi}}_{i,s}$ to be the sth solution produced by the function $\mathbf{g}\{\boldsymbol{\xi}_i\}$. Let the number of solutions produced by the ith converted cubature point be $\tilde{N}_{C,i}$. It is assumed that all converted points produced at least one solution or else the attempt at producing any results from the integral in (21) fails. Let the matrix $\mathbf{H}$ be such that $\mathbf{H}\tilde{\boldsymbol{\xi}}_{i,s}$ extracts the position components of converted point $\tilde{\boldsymbol{\xi}}_{i,s}$. The pairwise cost function for points is

$$d((i_1,s_1),(i_2,s_2)) \triangleq \begin{cases} \|\mathbf{H}\tilde{\boldsymbol{\xi}}_{i_1,s_1} - \mathbf{H}\tilde{\boldsymbol{\xi}}_{i_2,s_2}\| & i_1 \neq i_2 \text{ and } i_2 \neq \emptyset \\ \infty & i_1 = i_2 \\ c_{\max} & i_2 = \emptyset \end{cases} \quad (43)$$

where Ø represents an empty set (assigned to nothing) and $c_{\max}$ is an upper bound for the allowable distance between $\tilde{\boldsymbol{\xi}}_{i,s}$ and anything one would consider assigning it to. The ∞ costs say that we do not wish to assign two converted points originating from the same $\boldsymbol{\xi}_i$ together. For simplicity, we assume that the cubature points are ordered such that point $i = 1$ produces the least solutions.

To start we will initially consider the case where $N_C = 2$ and then present two approaches to generalize

Let the cost value in row i and column j of the cost matrix be $c_{i,j}$. The 2D assignment problem for clustering with $N_C = 2$ is

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_{i=1}^{N_{C,1}} \sum_{j=1}^{N_{C,2}+N_{C,1}} c_{i,j}x_{i,j} \quad (45)$$

$$\text{subject to } \sum_{j=1}^{N_{C,2}} x_{i,j} = 1 \quad \forall i$$

Every row is assigned to a column.

$$\sum_{i=1}^{N_{C,1}} x_{i,j} \leq 1 \quad \forall j \quad (46)$$

Not every column is assigned to a row.

$$x_{i,j} \in \{0,1\} \quad \forall x_{i,j} \quad (47)$$

Equivalent to $x_{i,j} \geq 0 \quad \forall i,j,$ (48)

where the matrix $\mathbf{X}$ is the set of all of the $x_{i,j}$. If $x_{i,j} = 1$, then the item in row i is assigned to the item in column j. Implicitly, the cost of not assigning a column to a row is zero. This is a standard 2D assignment problem and can be solved in strong polynomial time as described in [20], where Matlab code is also provided in an appendix.

The assignment set $\mathbf{X}$ tells us which rows are assigned to which columns. Because the assigned pairs always originate from different $\boldsymbol{\xi}_i$, they can be used in (21) to get different estimates. If no pairs assign, then there are no solutions. For the simulations in this paper, we just assume that $d((N_{C,1}),1),\emptyset)$, so there is no possibility of declaring two solutions too far apart to assign. In such an instance, one can just reduce the upper limit of the second cost function to $N_{C,2}$. In the following discussion, it is assumed for simplicity that no non-assignment costs are present.

For $N_C = k$ cubature points, a natural extension of the problem is (again letting $\boldsymbol{\xi}_1$ be the cubature point with the fewest converted solutions),

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_{i_1=1}^{N_{C,1}} \sum_{i_2=1}^{N_{C,2}} \cdots \sum_{i_k=1}^{N_{C,k}} c_{i_1,i_2,\ldots,i_k} x_{i_1,i_2,\ldots,i_k} \quad (49)$$

$$\text{subject to } \sum_{i_2=1}^{N_{C,2}} \cdots \sum_{i_k=1}^{N_{C,k}} x_{i_1,i_2,\ldots,i_k} = 1 \quad \forall i \quad (50)$$

$$\sum_{i_1=1}^{N_{C,1}} \sum_{i_3=1}^{N_{C,3}} \cdots \sum_{i_k=1}^{N_{C,k}} x_{i_1,i_2,\ldots,i_k} \leq 1 \quad \forall j \quad (51)$$

$$\vdots \quad (52)$$

$$\sum_{i_1=1}^{N_{C,1}} \cdots \sum_{i_{k-1}=1}^{N_{C,k-1}} x_{i_1,i_2,\ldots,i_k} \leq 1 \quad \forall j \quad (53)$$

$$x_{i_1,i_2,\ldots i_k} \in \{0,1\} \quad \forall i_1, i_2, \ldots, i_k \quad (54)$$

where the cost hyper-matrix, consisting of the elements of $c_{i_1,i_2,\ldots i_k}$ has yet to be specified. A particular value $c_{i_1,i_2,\ldots i_k}$ represents the cost of having $\tilde{\boldsymbol{\xi}}_{1,i_1}, \tilde{\boldsymbol{\xi}}_{2,i_2}, \ldots, \tilde{\boldsymbol{\xi}}_{k,i_k}$ in one cluster. One possible choice for the values of $c_{i_1,i_2,\ldots i_k}$ is the sum of all pairwise distances for cubature points in a common cluster:

$$c_{i_1,i_2,\ldots i_k} = \sum_{n=1}^{N_C} \sum_{m=n+1}^{N_C} \|\mathbf{H}\tilde{\boldsymbol{\xi}}_{n,i_n} - \mathbf{H}\tilde{\boldsymbol{\xi}}_{m,i_m}\|. \quad (55)$$

Another option is to take the distance of the points from their mean value, where here the "mean" value is the weighted mean as would be used in the numeric integral in (21)

$$c_{i_1,i_2,\ldots i_k} = \sum_{n=1}^{N_C} \|\mathbf{H}\tilde{\boldsymbol{\xi}}_{n,i_n} - \mathbf{H}\boldsymbol{\mu}_{n,i_1,i_2,\ldots,i_k}\|, \quad (56)$$

where

$$\boldsymbol{\mu}_n = \sum_{n=1}^{N_C} w_n \tilde{\boldsymbol{\xi}}_{n,i_n}. \quad (57)$$

Unfortunately, the optimization problem for $N_c > 2$ in (49) is a multiframe assignment problem, which is NP complete [53, Ch. 15.7]. That means that no known polynomial time algorithms exist to solve it, and it is unlikely one will be found. Thus, as in [22] a low-complexity approximation shall be used. This approximation will produce globally optimal results if all of the different solutions of the converted measurements are far apart.

The approximate algorithm is just sequential 2D assignment. The cubature points are arranged in order of an increasing number of solutions. Then:

1) Create a cost matrix as in (44) and perform assignment of solutions originating from $\boldsymbol{\xi}_1$ to those of $\boldsymbol{\xi}_2$.
2) Create a cost matrix as in (44) and perform assignment of solutions originating from $\boldsymbol{\xi}_2$ that were assigned to those in $\boldsymbol{\xi}_1$ to those of $\boldsymbol{\xi}_3$.
3) Continue assigning solutions originating from $\boldsymbol{\xi}_i$ that were assigned to those in $\boldsymbol{\xi}_{i-1}$ to those in $\boldsymbol{\xi}_{i+1}$ until a complete set of assignments is obtained.

The above sequential algorithm is used in the simulations. One would expect it to bias the estimates if multiple solutions are close together. Also, there are certain scenarios where one can, in theory, get bad results depending on the reliability of the multivariate polynomial solver. For example, suppose that a polynomial has two solutions, but the solver only finds one. If it finds a different solution for different cubature points, then the final result can be useless. Note that other clustering algorithms might also work, such as that used in contact sifting for sonar in [35].

7. ESTIMATION VIA MEASUREMENT CONVERSION

This section describes the primary technique for track initiation in this paper: using the minimum number of measurement components needed for target observability, and inverting the measurement function. Combined with the cubature moment estimation method of Section 3 this provides the conditional mean and covariance matrix of the target state given the measurements. However, a heuristic approximation has to be made for such an approach to be viable.

Section 7.1 describes the basic algorithm assuming that the measurement function h in (1) is bijective. However, as demonstrated in Section 7.2, the measurement function is generally not bijective. Thus, a heuristic algorithms is used in practice, which deals with h having no real solutions or multiple real solutions.

Subsection 7.3 provides expressions for estimating the position of a target given bistatic range and/or TDOA measurements. Assuming that the target is a stationary emitter and the receivers are mobile, expressions for Doppler measurement and frequency ratios (usually, the measurements are frequencies but the frequency ratio is used in the inverse function) are also provided. The use of frequency ratios for polynomial-based track initiation appears to be new. Frequency ratio-based estimation of any type is not as prevalent in the literature. It is used in an aircraft-tracking problem in the expired patent [59]. The techniques in this section differ from the polynomial-based localization methods of [66] as the algorithms are not optimizing a LS cost function and are thus significantly faster, though they cannot make use of redundant information. Also note that [66] does not provide any means of obtaining a covariance matrix for the estimates. Simulations are provided showing the performance of the estimator.

Subsection 7.4 then provides expressions for measurement equations where the measurements are not all simultaneous for reference. Bistatic range, TDOA, and DOA are considered. It is also worth highlighting the

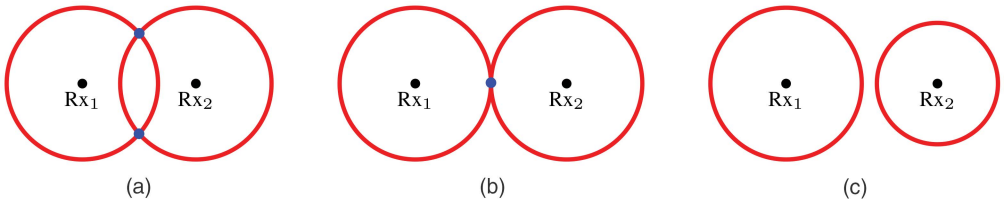


Fig. 1. A monostatic range measurement in 2D defines a circle around a sensor. Depending on the target location, there can be 2 or 1 solutions, as in (a) and (b). However, if measurements are noisy, there can be no real solutions if one tries to invert the measurement into Cartesian space as in (c). Conditioned on range measurements, the value of the noise that could be contained in the measurement must be constrained so that the measurement, after accounting subtracting the noise, can be converted into a real solution in Cartesian coordinates. (a) Two Real Solutions. (b) One Real Solution. (c) No Real Solutions.

special solutions [45], [46] for initiating a track consisting of position and velocity in 3D given six simultaneous Doppler measurements (with stationary sensors). Simulations are not given for this section, though the correctness of the polynomial equations was verified by solving a number of individual test cases.

7.1. The Measurement-Conversion Algorithm

Suppose that one would like to compute $\mathrm{E}\{\mathbf{x} \mid \hat{\mathbf{Z}}\}$, where $\hat{\mathbf{Z}}$ is the set of all available measurements and everything is in the real domain. This can be done by evaluating the integral

$$\mathrm{E}\{\mathbf{x} \mid \hat{\mathbf{Z}}\} = \int_{\mathbf{x}\in\mathbb{R}^n} \mathbf{x} p(\mathbf{x} \mid \hat{\mathbf{Z}}) d\mathbf{x} \tag{58}$$

where $p(\mathbf{x} \mid \hat{\mathbf{Z}})$ is the conditional distribution of $\mathbf{x}$ given $\hat{\mathbf{Z}}$. Assume that the measurements have the form of (1) and that the measurement function h is bijective. In such an instance, using the total probability theorem, one can write

$$\begin{aligned} p(\mathbf{x} \mid \hat{\mathbf{Z}}) &= p(\mathbf{x} \mid \mathbf{w}) p(\mathbf{w}) \\ &= \delta(\mathbf{x} - h^{-1}(\hat{\mathbf{Z}}, \mathbf{w})) p(\mathbf{w}) \end{aligned} \tag{59}$$

where δ represents the Dirac delta function. Assuming that the noise $\mathbf{w}$ is also n-dimensional, (58) becomes

$$\mathrm{E}\{\mathbf{x} \mid \hat{\mathbf{Z}}\} = \int_{\mathbf{w}\in\mathbb{R}^n} \int_{\mathbf{x}\in\mathbb{R}^n} \mathbf{x} \delta(\mathbf{x} - h^{-1}(\hat{\mathbf{Z}}, \mathbf{w})) p(\mathbf{w}) d\mathbf{x} d\mathbf{w} \tag{60}$$

$$= \int_{\mathbf{w}\in\mathbb{R}^n} h^{-1}(\hat{\mathbf{Z}}, \mathbf{w}) p(\mathbf{w}) d\mathbf{w} \tag{61}$$

where the simplification comes from the fact that a definite integral involving the Dirac delta function times a function equals the function evaluated where the argument of the Dirac delta function is zero.[14] For all of the examples in this paper, the noise corrupting the measurement is assumed to be additive as in (2). This means that

$$\mathbf{x} = h^{-1}(\hat{\mathbf{Z}} - \mathbf{w}). \tag{62}$$

Additionally, it is assumed that the noise corrupting the measurements is distributed multivariate Gaussian with zero mean and covariance matrix $\mathbf{R}$ (which will be block diagonal if the measurements are independent). Thus, the integral in (61) can be simplified to

$$\hat{\mathbf{x}} = \mathrm{E}\{\mathbf{x} \mid \hat{\mathbf{Z}}\} = \int_{\mathbf{Z}\in\mathbb{R}^n} h^{-1}(\mathbf{Z}) \mathcal{N}\{\mathbf{Z}; \hat{\mathbf{Z}}, \mathbf{R}\} d\mathbf{Z} \tag{63}$$

where $\mathcal{N}\{a, b, c\}$ represents the univariate or multivariate normal distribution with mean b and covariance matrix (or variance) c evaluated at point a. Thus, the estimator here is the inverse of the measurement function, and when using the cubature method of Section 3 with a high enough degree set of cubature points, one obtains the conditional mean and a conditional covariance matrix can also be found.

Next the question arises how one constructs the inverse measurement function h^{-1}. This is done by formulating the expression for each measurement type (TDOA, DOA, etc) within the measurement function as a multivariate polynomial. Given enough, one can solve the system using one of the off-the-shelf solvers mentioned in Section 5. However, one will quickly find that there are often no real solutions, or there are multiple solutions to the polynomials. This violates the assumption that h is bijective. Thus, the key to the algorithm in this paper is the use of the following heuristic approximation when no solutions are present. Additionally, multiple solutions shall be clustered and handled separately.

7.2. The Heuristic Approximation for Viability

Unfortunately, the measurement-conversion approach described in the previous section usually fails. As an example, consider the simple problem of estimating the location of a target in two dimensions given (one-way) monostatic range measurements from two sensors. For simplicity, both sensors are assumed to be located on the y-axis and each at a distance of l_x from the origin. Thus, the equations to be solved are

$$z_1 = \sqrt{(x - l_x)^2 + y^2} \tag{64}$$

$$z_2 = \sqrt{(x + l_x)^2 + y^2} \tag{65}$$

[14] Note the addition of the extra integral over $\mathbf{w}$ in (60) made necessary by the total probability theorem.

Inverting the system of equations, one gets the two solutions

$$x = \frac{z_2^2 - z_1^2}{4l_x} \tag{66}$$

$$y = \pm\frac{1}{4l_x}\sqrt{8l_x^2(z_1^2 + z_2^2) - 16l_x^4} \tag{67}$$

These two solutions can either be unique, coincide, or be imaginary. All three examples are illustrated in Fig. 1. If the solutions are *always* real, then the 2D assignment algorithm of Section 6 can be used to sort each of the two solutions for the cubature points (in the integrals in Section 7.1) into clusters and a separate mean and covariance matrix can be found for each. This also applies if only one solution is occasionally obtained as that single solution is really two solutions (a repeated root). For example, PHCpack will repeat a solution when there is a repeated root.

However, even that somewhat heuristic approach fails if there ever fails to be any real solutions. On the other hand, a simple, heuristic fix is to compute complex solutions that are "nearly" real, and discard the imaginary part. In (66), it is clear that no matter how the measurements z_1 and z_2 are perturbed, the resulting x component will always be real. In (67), it can be seen that if the true target has a y component close to 0, then small perturbations in z_1 and z_2 will either slightly push the solutions apart in y, cause them to come completely together, or cause them to become imaginary. However, if the perturbation makes the solutions imaginary, then it can be seen that the real component of y is zero, which is actually the LS real estimate (as seen in Fig. 1c, the place where the two circles come closest for any fixed value of x_1. Thus, just computing the imaginary solution and taking the real part works quite well.

Thus, the heuristic solution in this paper is to keep some complex solutions, discarding the imaginary part, and when computing the integrals in Section 7.1, cluster multiple solutions to the measurement conversion as in Section 6.

7.3. Equations for Static Estimation

This section provides equations for different measurements types when localizing a target (or an emitter in the case of the Doppler/frequency measurements). The 2D or 3D target-location vector is denoted by $\mathbf{t}$ to differentiate it from a full target state, which is considered in Section 7.4. It is noted how manipulations to the equations might induce false solutions that must be discarded when using certain measurement types. Simulation results then follow.

7.3.1. TDOA Measurements: The basic formulation for solving TDOA measurement-conversion problems using multivariate polynomial rooting is given in [72]. When considering TDOA measurements, the same general procedure is used for the derivation, with a minor correction to the sign used in the definition of y.

For a single received signal, let $\mathbf{l}_1$ be the Cartesian location of the first receiver (in 2D or 3D) and $\mathbf{l}_2$ be the Cartesian location of the second receiver. The model for a TDOA measurement is

$$\frac{1}{c}(\|\mathbf{t} - \mathbf{l}_1\| - \|\mathbf{t} - \mathbf{l}_2\|) = \text{TDOA} \tag{68}$$

where c is the speed of propagation in the medium. Making the following substitutions

$$\mathbf{u} = \frac{1}{2}(\mathbf{l}_1 + \mathbf{l}_2^{\text{Rx}}) \qquad \mathbf{v} = \frac{1}{2}(\mathbf{l}_1 - \mathbf{l}_2) \tag{69}$$

$$\mathbf{y} = \mathbf{t} - \mathbf{u} \qquad \delta = \frac{\text{TDOA}c}{2} \tag{70}$$

and substituting into (68), one gets

$$\|\mathbf{y} - \mathbf{v}\| - \|\mathbf{y} + \mathbf{v}\| = 2\delta. \tag{71}$$

This can be turned into a polynomial by isolating $\|\mathbf{y} - \mathbf{v}\|$ on one side, squaring the result, and then isolating $\|\mathbf{y} + \mathbf{v}\|$ and squaring the result:

$$\|\mathbf{y} - \mathbf{v}\|^2 = (2\delta + \|\mathbf{y} + \mathbf{v}\|)^2 \tag{72}$$

$$\mathbf{y}'\mathbf{y} + \mathbf{v}'\mathbf{v} - 2\mathbf{y}'\mathbf{v} = 4\delta^2 + 4\delta\|\mathbf{y} + \mathbf{v}\| + \mathbf{y}'\mathbf{y} + \mathbf{v}'\mathbf{v} + 2\mathbf{y}'\mathbf{v} \tag{73}$$

$$\mathbf{y}'\mathbf{v} + \delta^2 = -\delta\|\mathbf{y} + \mathbf{v}\| \tag{74}$$

$$(\mathbf{y}'\mathbf{v} + \delta^2)^2 = \delta^2\|\mathbf{y} + \mathbf{v}\|^2 \tag{75}$$

$$(\mathbf{y}'\mathbf{v})^2 + \delta^4 + 2\delta^2\mathbf{y}'\mathbf{v} = \delta^2(\mathbf{y}'\mathbf{y} + \mathbf{v}'\mathbf{v} + 2\mathbf{y}'\mathbf{v}) \tag{76}$$

$$(\mathbf{y}'\mathbf{v})^2 - \delta^2(\mathbf{y}'\mathbf{y} + \mathbf{v}'\mathbf{v}) + \delta^4 = 0. \tag{77}$$

In order to solve the problem, one must substitute the $\mathbf{t}$ values back in, because the $\mathbf{y}$ terms are different for each equation in the polynomial system that must be solved. Thus, one gets

$$((\mathbf{t} - \mathbf{u})'\mathbf{v})^2 - \delta^2(\|\mathbf{t} - \mathbf{u}\|^2 + \|\mathbf{v}\|^2) + \delta^4 = 0 \tag{78}$$

$$(\mathbf{t}'\mathbf{v})^2 - \delta^2\|\mathbf{t}\|^2 + \mathbf{t}'\overbrace{(2\delta^2\mathbf{u} - 2(\mathbf{u}'\mathbf{v})\mathbf{v})}^{\tilde{\mathbf{l}}} + \overbrace{(\delta^4 - \delta^2\|\mathbf{v}\|^2 + (\mathbf{u}'\mathbf{v})^2 - \delta^2\|\mathbf{u}\|^2)}^{\tilde{c}} = 0 \tag{79}$$

In two dimensions, (78) expands in terms of monomials to

$$0 = \tilde{c} + \tilde{l}_1 t_1 + \tilde{l}_2 t_2 + 2v_1 v_2 t_1 t_2 + t_1^2(v_1 - \delta)(v_1 + \delta) + t_2^2(v_2 - \delta)(v_2 + \delta) \tag{80}$$

and in three dimensions, (78) expands to

$$0 = \tilde{c} + \tilde{l}_1 t_1 + \tilde{l}_2 t_2 + 2v_1 v_2 t_1 t_2 + \tilde{l}_3 t_3 + 2v_1 v_3 t_1 t_3 + 2v_2 v_3 t_2 t_3 + t_1^2(v_1 - \delta)(v_1 + \delta) + t_2^2(v_2 - \delta)(v_2 + \delta) + t_3^2(v_3 - \delta)(v_3 + \delta) \tag{81}$$

The squaring of the equation to remove the square roots can introduce false solutions. That is, solutions where the TDOA computed from them is not equal to the true TDOA. Consequently, an estimation algorithm utilizing TDOA equations should look for such inconsistencies and discard extra solutions.

7.3.2. Bistatic Range Measurements: The procedure for using bistatic range-only measurements is very similar to that for using TDOA-only measurements. Let $\mathbf{l}_1$ be the location of the transmitter and $\mathbf{l}_2$ be the location of the receiver. The measurement equation is

$$\|\mathbf{t}-\mathbf{l}_1\| + \|\mathbf{t}-\mathbf{l}_2\| = r, \tag{82}$$

which has essentially the same form as (68) for the TDOA case, except the sign of one of the norms is flipped. Hence, the problem can be solved in the same manner. Define

$$\mathbf{u} = \frac{1}{2}(\mathbf{l}_1 + \mathbf{l}_2) \qquad \mathbf{v} = \frac{1}{2}(\mathbf{l}_1 - \mathbf{l}_2) \tag{83}$$

$$\mathbf{y} = \mathbf{t} - \mathbf{u} \qquad \delta = \frac{r}{2}. \tag{84}$$

Substituting into (82), one gets

$$\|\mathbf{y}-\mathbf{v}\| + \|\mathbf{y}+\mathbf{v}\| = 2\delta. \tag{85}$$

This equation has the same form as (71), but with a flipped sign on the second term. The simplification proceeds as in Section 7.3.1:

$$\|\mathbf{y}-\mathbf{v}\|^2 = (2\delta - \|\mathbf{y}+\mathbf{v}\|)^2 \tag{86}$$

$$\mathbf{y}'\mathbf{y} + \mathbf{v}'\mathbf{v} - 2\mathbf{y}'\mathbf{v} = 4\delta^2 - 4\delta\|\mathbf{y}+\mathbf{v}\| + \mathbf{y}'\mathbf{y} + \mathbf{v}'\mathbf{v} + 2\mathbf{y}'\mathbf{v} \tag{87}$$

$$\mathbf{y}'\mathbf{v} + \delta^2 = \delta\|\mathbf{y}+\mathbf{v}\|. \tag{88}$$

The square of (88) has the same form as (75). The final solution has the same form as (77):

$$(\mathbf{y}'\mathbf{v})^2 - \delta^2(\mathbf{y}'\mathbf{y} + \mathbf{v}'\mathbf{v}) + \delta^4 = 0. \tag{89}$$

Substituting the $\mathbf{t}$ values back in, one gets (79), where as in the TDOA case, the 2D and 3D simplifications are (80) and (81), respectively.

The equations used for a range measurement are the same as those for a TDOA measurement, except the definitions of $\mathbf{u}$, $\mathbf{v}$, and δ are different. The same equations apply for monostatic measurements. In such an instance $\mathbf{v} = \mathbf{0}$. Also, though there is squaring in the range equations, no sign flips can occur as in the TDOA case. Thus, bistatic range-only estimation will often have more solutions than TDOA-only estimation.

7.3.3. Emitter Doppler Measurements: In this instance, we are considering measuring the Doppler offset observed by a moving receiver picking up a signal from a stationary emitter that broadcasts at a known frequency. Let $\mathbf{l}$ and $\dot{\mathbf{l}}$ be the location and velocity vectors of the receiver. As discussed in the measurement model in Section 2, the measured range rate simplifies to

$$\dot{r} = -\left(\frac{\mathbf{t}-\mathbf{l}}{\|\mathbf{t}-\mathbf{l}\|}\right)'\dot{\mathbf{l}}. \tag{90}$$

This equation can be turned into a multivariate polynomial as

$$(\dot{r}\|\mathbf{t}-\mathbf{l}\|)^2 = (\mathbf{t}'\dot{\mathbf{l}} - \mathbf{l}'\dot{\mathbf{l}})^2 \tag{91}$$

$$\dot{r}^2(\|\mathbf{t}\|^2 + \|\mathbf{l}\|^2 - 2\mathbf{t}'\mathbf{l}) = (\mathbf{t}'\dot{\mathbf{l}})^2 + (\mathbf{l}'\dot{\mathbf{l}})^2 - 2(\mathbf{t}'\dot{\mathbf{l}})(\mathbf{l}'\dot{\mathbf{l}}). \tag{92}$$

Consequently, the final polynomial equation is

$$\dot{r}^2\|\mathbf{t}\|^2 - (\mathbf{t}'\dot{\mathbf{l}})^2 + \mathbf{t}'\overbrace{(2\dot{\mathbf{l}}(\mathbf{l}'\dot{\mathbf{l}}) - 2\dot{r}^2\mathbf{l})}^{\tilde{\mathbf{l}}} + \overbrace{\dot{r}^2\|\mathbf{l}\|^2 - (\mathbf{l}'\dot{\mathbf{l}})^2}^{\tilde{c}} = 0. \tag{93}$$

In two dimensions, (93) expands to

$$0 = \tilde{c} + \tilde{l}_1 t_1 + (\dot{r}^2 - \dot{l}_1^2)t_1^2 + \tilde{l}_2 t_2 - 2\dot{l}_1\dot{l}_2 t_1 t_2 + (\dot{r}^2 - \dot{l}_2^2)t_2^2. \tag{94}$$

In three dimensions, (93) expands to

$$\begin{aligned} 0 = \tilde{c} + \tilde{l}_1 t_1 + (\dot{r}^2 - \dot{l}_1^2)t_1^2 + \tilde{l}_2 t_2 - 2\dot{l}_1\dot{l}_2 t_1 t_2 + (\dot{r}^2 - \dot{l}_2^2)t_2^2 \\ + \tilde{l}_3 t_3 - 2\dot{l}_1\dot{l}_3 t_1 t_3 - 2\dot{l}_2\dot{l}_3 t_2 t_3 + (\dot{r}^2 - \dot{l}_3^2)t_3^2. \end{aligned} \tag{95}$$

The scaling method described in Section 5 is particularly important when using Doppler measurements in 3D. Due to the squaring of the equations, one can also expect there to be added solutions that must be eliminated as described in Section 7.3.1.

7.3.4. Emitter Frequency-Ratio Measurements: In the case of a stationary emitter, taking Sensor 1 in the numerator, the measurement model in Section 2, simplifies to

$$f_{i,j}^R = \frac{r_j(\mathbf{t}-\mathbf{l}_i)'\dot{\mathbf{l}}_i + cr_i r_j}{r_i(\mathbf{t}-\mathbf{l}_j)'\dot{\mathbf{l}}_j + cr_i r_j} \tag{96}$$

where the r terms are given by the polynomial equation

$$r_i^2 = \|\mathbf{t}-\mathbf{l}_i^{\mathrm{Rx}}\|^2 = \|\mathbf{t}\|^2 + \|\mathbf{l}_i\|^2 - 2\mathbf{t}'\mathbf{l}_i. \tag{97}$$

Equation (96) can be written as a polynomial that must be zeroed as

$$\begin{aligned} 0 = f_{i,j}^R r_i \mathbf{t}'\dot{\mathbf{l}}_j - r_j\mathbf{t}'\dot{\mathbf{l}}_i - f_{i,j}^R r_1(\mathbf{l}_j)'\dot{\mathbf{l}}_j + r_j(\mathbf{l}_i)'\dot{\mathbf{l}}_i \\ + (f_{i,j}^R - 1)cr_i r_j. \end{aligned} \tag{98}$$

In 2D, the two equations to consider expand to

$$0 = -l_{i,1}^2 - l_{i,2}^2 + r_i^2 + 2l_{i,1}t_1 - t_1^2 + 2l_{i,2}t_2 - t_2^2 \tag{99}$$

$$\begin{aligned} 0 = -f_{i,j}^R(\dot{l}_{j,1}l_{j,1} + \dot{l}_{j,2}l_{j,2})r_i + (\dot{l}_{i,1}l_{i,1} + \dot{l}_{i,2}l_{i,2})r_j \\ + c(f_{i,j}^R - 1)r_i r_j + f_{i,j}^R\dot{l}_{j,1}r_1 t_1 - \dot{l}_{i,1}r_j t_1 \\ + f_{i,j}^R\dot{l}_{j,2}r_i t_2 - \dot{l}_{i,2}r_j t_2 \end{aligned} \tag{100}$$

and in 3D, they are

$$0 = -l_{i,1}^2 - l_{i,2}^2 - l_{i,3}^2 + r_i^2 + 2l_{i,1}t_1 - t_1^2 + 2l_{i,2}t_2 - t_2^2 + 2l_{i,3}t_3 - t_3^2 \tag{101}$$

$$\begin{aligned} 0 = &-f_{i,j}^R(\dot{l}_{j,1}l_{j,1} + \dot{l}_{j,2}l_{j,2} + \dot{l}_{j,3}l_{j,3})r_i \\ &+ (\dot{l}_{i,1}l_{i,1} + \dot{l}_{i,2}l_{i,2} + \dot{l}_{i,3}l_{i,3})r_j \\ &+ c(-1 + f_{i,j}^R)r_i r_j + f_{i,j}^R\dot{l}_{j,1}r_i t_1 - \dot{l}_{i,1}r_j t_1 \\ &+ f_{i,j}^R\dot{l}_{j,2}r_i t_2 - \dot{l}_{i,2}r_j t_2 + f_{i,j}^R\dot{l}_{j,3}r_i t_3 - \dot{l}_{i,3}r_j t_3. \end{aligned} \tag{102}$$

Due to the squaring of the equations, one can also expect there to be added solutions that must be eliminated as described in Section 7.3.1.

In practice, one will generally measure frequencies and not ratios. However, taking the ratio can be considered to be the first step in computing the measurement function.

7.3.5. Simulation Examples: To demonstrate the results, we consider a simulation where the polynomial solving algorithms of the previous subsections are solved using the algorithms of Section 5. Complex solutions are mapped to the real domain as in Section 7.2, so that the measurement-conversion algorithm of Section 7.1 can be used, whereby multiple polynomial solutions for each cubature point in the integrals are clustered into different sets as in Section 6. The cubature points of Appendix B are used.

Allowing slightly complex solutions, as is the heuristic in Section 7.2, can increase the number of useless solutions. However, one can discard many of them by making sure that $h(\hat{\mathbf{x}})$ does not differ "too much" from the true solution in any one component. Such solutions will arise due to the squaring of equations in the previous sections. In such circumstances, these false solutions, when converted into the measurement domain, will produce measurements with components having the wrong sign. Again, a simple comparison of the converted solutions to the actual measurements can help eliminate such false solutions.

In the simulations, we assess the root mean squared error (RMSE) of the estimates and the normalized estimation error squared (NEES), which, as mentioned in [23], is a common method of assessing the consistency of covariance estimates. The RMSE is compared to the CRLB to determine how well the estimator approaches the bound. However, if there truly are, for example, two solutions, then the likelihood function would be bimodal and the mean would be between the solutions. If the solutions are far apart, then the mean might be far from either one. Choosing the solution closest to the truth for analysis effectively introduces a bias and it can be possible to have RMSE values below the CRLB.

Three scenarios are considered. Sensors considered in the scenarios are placed at the following latitudes (degrees), longitudes (degrees) and World Geodetic System-1984 (WGS-84) ellipsoidal altitudes (meters):

Name	Latitude °	Longitude °	Altitude m
s_1	20.265901	−155.857544	8000
s_2	19.878939	−155.107727	7500
s_3	19.661825	−156.091003	6000
s_4	20.069960	−155.434570	5000

The sensors are in the air around the northern part of the island of Hawaii. The conversion between the WGS-84 ellipsoidal coordinates and global Cartesian coordinates is described in [25]. Also provided are expressions for unit vectors in the direction of local East-North and up, which can help when establishing the heading of the sensors.

In all three scenarios, the true target position is varied over a grid in latitude in the range $(18.5°, 19.75°)$ and longitude in the range $(-156°, -154.75°)$. The altitude of the true target is fixed at 4205 m, which, ignoring geoid undulations, is approximately the height of the tallest mountain on Hawaii. It is assumed that all sensors always have direct lines of sight to the target and atmospheric refraction and diffraction over the terrain are neglected. Of course, Scenarios 2 and 3 would be more realistic if the target were fixed to the ground as the target is assumed to be a *stationary* emitter in such an instance. However, the inclusion of terrain data would needlessly complicate the example.

Three scenarios are considered:

1) The first scenario is representative of multiple aircraft working to localize a cooperative stationary emitter. In this instance, only Sensors s_1 and s_2 are used. However, now a velocity for Sensor 2 is provided and the emitter is assumed stationary. The range measurements are to and from Sensor s_1 (a round-trip monostatic measurement) and from Sensor s_1 to s_2. The range-rate measurement is made by Sensor s_2, which is traveling 300 m/s in level flight 45° East of North. The unit vectors for the directions can be obtained as in [25]. The standard deviation of the range measurements is 1 m and the standard deviation of the range-rate measurement is 0.1 m. There might be a scenario where a sensor pings a target with a repeater and the target pings back. In such an instance, the measured ranges would be assumed to be derived from measured time delays minus the time it takes the repeated to respond. Fifty Monte Carlo runs are performed and the grid of target locations used for the plots is 15×15 in latitude and longitude.

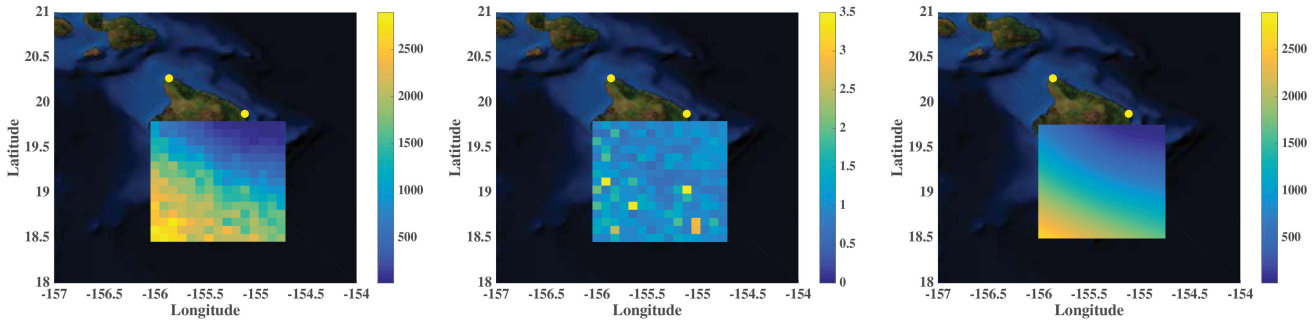


Fig. 2. The RMSE in meters and NEES of the simulations in Scenario 1 (range and emitter range-rate measurements) on a 15×15 grid shown on top of a map after 50 Monte Carlo runs using Bertini. Also, the square root of the trace of the approximate CRLB (in meters, on a fine grid) is shown for comparison to the RMSE. The yellow dots represent locations of the sensors. The island is about 120 kilometers wide. The color bar for the NEES is clipped. The color scale for the CRLB and RMSE are the same. (a) RMSE. (b) NEES. (c) CRLB.

2) The second scenario is a slight variant on an "easy" scenario. All four sensors are used. Three of them produce TDOA measurements, while a bistatic path is given between a pair of them. A standard deviation of 10 m is used for the range measurement. For TDOA measurements, a standard deviation of 3.3356×10^{-8} s is used; this equals 10 m/c seconds, where c = 299,792,458 m/s is the speed of light in a vacuum, which was used as the transmission speed in the simulation. The TDOA measurements are between Sensors s_1 and s_2 and between Sensors s_1 and s_3. The bistatic range measurement is between Sensors s_3 and s_4. This scenario might occur when one tries to fuse measurements of an emitter from passive sensors with an active radar range measurement. The altitudes of the aircraft imply detections taken by aircraft. Two hundred Monte Carlo runs are performed and the grid of target locations used for the plots is 15×15 in latitude and longitude.
3) The final scenario is a case where all four sensors are moving and only make frequency measurements as they are assumed to not know the transmission frequency of the target, which is an emitter. All sensors are moving level with respect to the WGS-84 reference ellipsoid. Sensors 1 and 2 are moving at 300 m/s with Sensor s_1 going 45° west of North and Sensor s_2 going 45° East of North. Sensors 3 and 4 are traveling, respectively, East and North at 250 m/s. The transmission frequency of the emitter is 8 GHz (X-Band) and the standard deviation of the measurements is 1 Hz. Note that it does not matter which frequency ratios from the four frequency measurements are used in the multivariate polynomial solver (given the individual frequency measurements) as long as none of the ratios are redundant. Fifty Monte Carlo runs are performed and the grid of target locations used for the plots used is 15×15 in latitude and longitude.

The three scenarios do not appear to be solved by algorithms that are already present in the literature, even though they represent simple uses of multivariate polynomial solving algorithms and cubature integration.

The shortest augmenting path 2D assignment algorithm of [20] was used for clustering converted cubature points, as discussed in Section 6. Bertini, PHCpack and the certified solver in Macaulay2, called from Matlab, were considered as polynomial solvers. However, Macaulay2's certified solver tended to be too slow and failed much more than the others, so it was not used in any of the simulations plotted here (failure being defined as no solutions being produced or the only solution being over 1000 km from the true target). PHCpack was used for Scenario 2, as it is easier to parallelize in Matlab. Bertini was used for the other two scenarios. When using Bertini, the options `SecurityMaxNorm`, `EndpointFiniteThreshold` and `PathTruncationThreshold` were all set to 10^9 to improve reliability and FinalTol was set to 10^{-14}.

Figure 2 shows the RMSE, NEES, and CRLB of Scenario 1. Bertini was used as the polynomial solver as PHCpack was found to occasionally fail. The 95% bounds for the NEES are 0.787 and 1.239 (based on the inverse cumulative distribution function [CDF] of the chi-squared distribution with 150 degrees of freedom). Averaging the MSE over all of the cells in the plot and then taking the square root, one gets an accuracy of 1512 m, whereas averaging the trace of the CRLB over the same points and taking the square root, one gets a bound of 1276 m, indicating that the estimates are good, but do not quite hit the CRLB. The median of the NEES values in all the cells is 0.9931, indicating good covariance consistency, though the mean is 1.58. This discrepancy is explained by two outlier points, one having a value of 103.18 and another a value of 14.43.

The RMSE, NEES, and CRLB of Scenario 2 are shown in Fig. 3. PHCpack was used as the polynomial solver. The 95% bounds for the NEES are 0.890 and 1.116, which as discussed in [23] represent the solution of the inverse CDF of the chi-squared distribution with 600 degrees of freedom evaluated at 0.025 and 0.0975

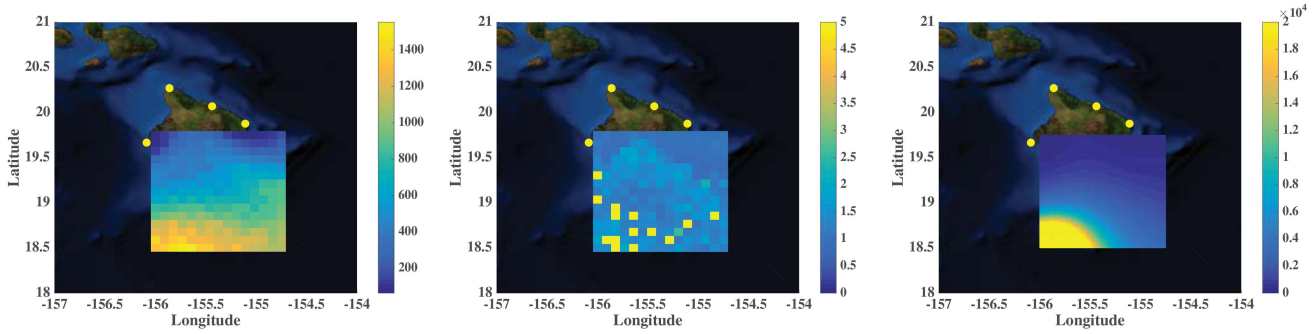


Fig. 3. The RMSE in meters and NEES of the simulations in Scenario 2 (TDOA and range measurements) on a 15×15 grid shown on top of a map after 200 Monte Carlo runs using PHCpack. Also, the square root of the trace of the CRLB (in meters, on a fine grid) is shown for comparison to the RMSE. The yellow dots represent locations of the sensors. The NEES is generally within the 95% bounds for consistency. The color scale on the CRLB was capped at 2×10^4 m even though it goes much higher. The RMSE is below the CRLB in some instances, because the true likelihood is multimodel and by choosing the solution nearest the truth for analysis, we bias the estimator as seen in Figure 4. (a) RMSE. (b) NEES. (c) CRLB.

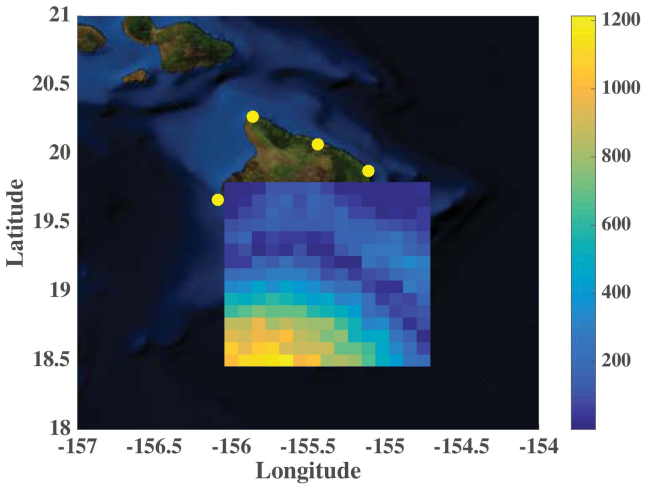


Fig. 4. The magnitude of the bias of the estimates in Scenario 2. The bias is caused by only selecting the closest hypothesis for comparison to the RMSE. The true distribution is multimodal and as the modes separate, the bias becomes larger. Additionally, due to the estimator itself not being guaranteed unbiased (due to the projection of complex results into the real domain as in Section 7.2), the estimator (independent of the different modes) is not completely guaranteed to be unbiased. This complicates the analysis of the results.

and divided by 600 (the dimensionality times the number of Monte Carlo runs). Averaging the MSE over all of the cells in the plot and then taking the square root, one gets an accuracy of 803 m, whereas averaging the trace of the CRLB over the same points and taking the square root, one gets a bound of 143,773 m, with most of the contribution coming from the lower-left part of the viewing region. Initially, one would have to assume that something is wrong as the RMSE of an unbiased estimator cannot be lower than the CRLB. However, as seen in Fig. 4, one can see that the estimator is very biased. The maximum bias of the estimator is 1,214 m. This is an artifact of only choosing the best estimate to use for the analysis, whereas the true distribution is multimodal.

The median NEES of the estimates is 1.28, which is slightly inconsistent, though not very much so. On the other hand, the mean is 5,955. Again, as in Scenario 1, this is caused by a number of outliers, in this case 13 out of the 225 grid points.

The RMSE, NEES, and CRLB of Scenario 3 are shown in Fig. 5 . The third scenario is the most difficult: received frequency-only localization. This is a problem that is subject to finite-precision errors due to the number of digits that must be carried in frequency measurements that go into frequency ratios. This is also the slowest scenario for the multivariate polynomial solvers. A single solution of the polynomial system without any type of threading/parallelization takes about 1.25 s in PHCpack and 64 s in Bertini.

Despite the difficulty of the problem and the slow run times of the solvers, one can see in Fig. 5 that over much of the region, estimates better than 10 km accuracy are possible. This suggests that with additional refinement and optimization, one should be able to obtain a usable estimation routine.

7.4. Equations for Dynamic Estimation

We provide equations for different measurement types when estimating the state of a target using non-simultaneous measurement. These measurements will typically consist of position and velocity components, though other components can be included as well. The state is denoted by $\mathbf{x}$.

Due to the number of terms present, the equations are written in vector form rather than expanded into individual terms as in the previous section. It is noted how manipulations of the equations might induce false solutions that must be discarded.

7.4.1. TDOA Measurements: For non-simultaneous TDOA measurements, the measurement equation is essentially the same as in (68), except the state $\mathbf{x}$ must be

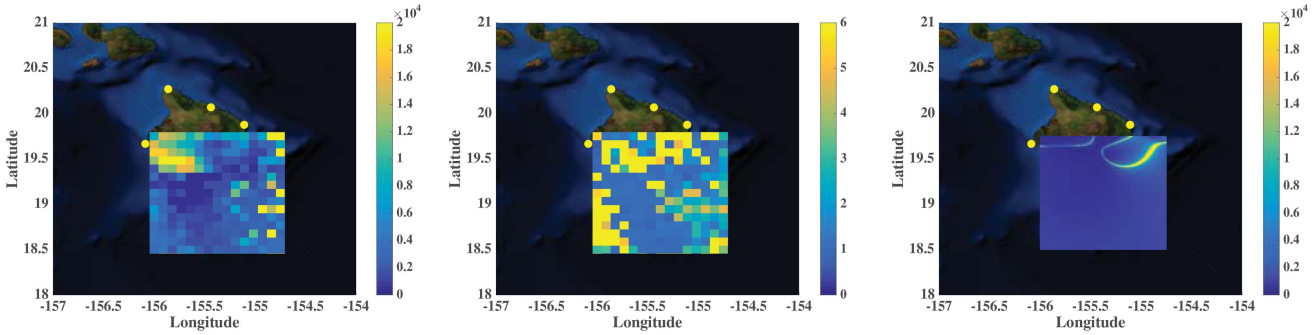


Fig. 5. The RMSE in meters and NEES of the simulations in Scenario 3 (only frequency measurements–unknown carrier frequency) on a 15×15 grid shown on top of a map after 50 Monte Carlo runs using PHCpack, which did fail on a single Monte Carlo run. On a few test points, Bertini was much slower, but also performed better. Also, the square root of the trace of the CRLB (in meters, on a fine grid) is shown for comparison to the RMSE. Color bars in all three plots are clipped. Though the RMSE is often less than 10 km, possibly providing usable estimates, it is worse than the CRLB and there are often bad estimates. However, it is the first instance of frequency-only track initiation in the literature when the transmission frequency is completely unknown to the receivers. Using a better polynomial solver along with higher-order cubature points is expected to improve performance. (a) RMSE. (b) NEES. (c) CRLB.

used in the equations, taking into account the propagation of the state to the time of the measurement. Using the notation of Section 2, this is

$$\text{TDOA} = \frac{1}{c}(\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\| - \|\mathbf{F}_h\mathbf{t} - \mathbf{l}_2\|). \tag{103}$$

The same simplifications can be performed as in Section 7.3.1, except in this case $\mathbf{y} = \mathbf{F}_h\mathbf{x} - \mathbf{u}$. This means that (79) becomes

$$0 = ((\mathbf{F}_h\mathbf{x} - \mathbf{u})'\mathbf{v})^2 - \delta^2((\mathbf{F}_h\mathbf{x} - \mathbf{u})'(\mathbf{F}_h\mathbf{x} - \mathbf{u}) + \|\mathbf{v}\|^2) + \delta^4 \tag{104}$$

$$0 = (\mathbf{x}'\mathbf{F}_h'\mathbf{v} - \mathbf{u}'\mathbf{v})^2 - \delta^2\mathbf{x}'\mathbf{F}_h'\mathbf{F}_h\mathbf{x} + 2\delta^2\mathbf{u}'\mathbf{F}_h\mathbf{x} - \delta^2\|\mathbf{u}\|^2 - \delta^2\|\mathbf{v}\|^2 + \delta^4 \tag{105}$$

$$0 = (\mathbf{x}'\mathbf{F}_h'\mathbf{v})^2 - 2(\mathbf{x}'\mathbf{F}_h'\mathbf{v})(\mathbf{u}'\mathbf{v}) - \delta^2\mathbf{x}'\mathbf{F}_h'\mathbf{F}_h\mathbf{x} + 2\delta^2\mathbf{u}'\mathbf{F}_h\mathbf{x} - \delta^2\|\mathbf{u}\|^2 - \delta^2\|\mathbf{v}\|^2 + \delta^4 + (\mathbf{u}'\mathbf{v})^2. \tag{106}$$

Again, as in Section 7.3.1, the squaring involved in manipulating the equations can add extra solutions that must be eliminated.

7.4.2. Bistatic Range Measurements: Including the effects of non-simultaneous measurement times, the range measurement equation is as in (82):

$$r = \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\| + \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_1\|. \tag{107}$$

As with non-simultaneous TDOA measurements, the non-simultaneous range measurements are the same as in Section 7.3.2 except $\mathbf{y} = \mathbf{F}_h\mathbf{x} - \mathbf{u}$. The final equation thus is thus (106), where it is noted that $\delta = r/2$ unlike in the TDOA case which includes scaling with c.

7.4.3. DOA Measurements: Given $u - v$ components forming a DOA measurement, the measurement equation of (11) is

$$\mathbf{u}_{uv} = \frac{1}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\|}\mathbf{M}_u(\mathbf{F}\mathbf{x} - \mathbf{l}). \tag{108}$$

As in the previous sections, various manipulations and squaring can eliminate the non-polynomial terms. Here, we use a Hadamard product (element-by-element multiplication) via the binary operator $\circ$ to help eliminate the square root term:

$$\mathbf{u}_{uv}\|\mathbf{F}_h\mathbf{x} - \mathbf{l}\| = \mathbf{M}_u(\mathbf{F}\mathbf{x} - \mathbf{l}) \tag{109}$$

$$\mathbf{u}_{uv} \circ \mathbf{u}_{uv}(\mathbf{F}\mathbf{x} - \mathbf{l})'(\mathbf{F}\mathbf{x} - \mathbf{l}) = (\mathbf{M}_u\mathbf{F}\mathbf{x} - \mathbf{M}_u\mathbf{l}) \circ (\mathbf{M}_u\mathbf{F}\mathbf{x} - \mathbf{M}_u\mathbf{l}). \tag{110}$$

The final two equations are expressed in vector form as

$$\mathbf{0} = (\mathbf{u}_{uv} \circ \mathbf{u}_{uv})\mathbf{x}'\mathbf{F}'\mathbf{F}\mathbf{x} - (\mathbf{M}_u\mathbf{F}\mathbf{x}) \circ (\mathbf{M}_u\mathbf{F}\mathbf{x}) - 2(\mathbf{u}_{uv} \circ \mathbf{u}_{uv})\mathbf{x}'\mathbf{F}'\mathbf{l} + 2(\mathbf{M}_u\mathbf{F}\mathbf{x}) \circ (\mathbf{M}_u\mathbf{l}) + (\mathbf{u}_{uv} \circ \mathbf{u}_{uv})\|\mathbf{l}\|^2 - (\mathbf{M}_u\mathbf{l}) \circ (\mathbf{M}_u\mathbf{l}). \tag{111}$$

8. MAXIMUM-LIKELIHOOD/LEAST-SQUARES TRACK INITIATION

Unlike in Section 7, the case where the measurement function $\mathbf{Z} = \mathbf{h}(\mathbf{x})$ is not bijective is considered. This can either occur if not enough measurements are available for the target to be observable (an instance that is not considered here) or if extra (redundant) measurements are given. For example, two simultaneous non-collocated bistatic range and DOA measurements offer more degrees of freedom than are necessary to uniquely specify the target position. In such an instance, we would like to find the ML/LS estimate.

The likelihood of the measurements is assumed Gaussian distributed, $\mathcal{N}\{\mathbf{Z}; \hat{\mathbf{Z}}, \mathbf{R}\}$ in the local coordinate system of the receiver. As is most common, we shall assume that the different measurements are independent, though this only affects the indexation used below and not the final solution. Let $\hat{\mathbf{z}}_i$ denote the ith measurement and $\mathbf{R}_i$ the covariance matrix associated with the ith measurement. Thus, the ith measurement is distributed $\mathcal{N}\{\mathbf{z}; \hat{\mathbf{z}}_i, \mathbf{R}_i\}$. The joint PDF of the measurements is the

product of the individual independent distributions. The ith distribution conditional distribution of $\mathbf{x}$ is

$$p(\mathbf{x} \mid \hat{\mathbf{z}}_i) = |2\pi\mathbf{R}_i|^{-1/2} e^{-(1/2)(h_i(\mathbf{x})-\hat{\mathbf{z}}_i)'\mathbf{R}_i^{-1}(h_i(\mathbf{x})-\hat{\mathbf{z}}_i)} \quad (112)$$

where $\mathbf{z}_i = h_i(\mathbf{x})$ is the transformation of $\mathbf{x}$ from the state domain into the measurement domain of the ith measurement. This transformation is assumed to be unique. The total likelihood for N measurements is

$$p(\mathbf{x} \mid \hat{\mathbf{z}}_i) = \prod_{i=1}^{n} p(\mathbf{x} \mid \hat{\mathbf{z}}_i). \quad (113)$$

To obtain the ML estimate, we can take the logarithm of the likelihood, discard constant terms and flip the sign of the result to get the following LS optimization problem in terms of the sum of Mahalanobis distances

$$\hat{\mathbf{x}}_{ML} = \arg\min_{\mathbf{x}} \sum_{i=1}^{N} (h_i(\mathbf{x}) - \hat{\mathbf{z}}_i)'\mathbf{R}_i^{-1}(h_i(\mathbf{x}) - \hat{\mathbf{z}}_i). \quad (114)$$

The minimum of the right-hand side of the cost function is at a point where the gradient with respect to $\mathbf{x}$ is zero. This leads to the equation for the ith component of the gradient, which corresponds to the partial derivative with respect to the ith component of $\mathbf{x}$ to be

$$0 = \sum_{i=1}^{N} (h_i(\mathbf{x}) - \mathbf{z})'\mathbf{R}_i^{-1} \left(\frac{\partial}{\partial x_i} h_i(\mathbf{x})' \right). \quad (115)$$

If $h_i(\mathbf{x})$ is a polynomial, then all N equations (115) represent a system of multivariate polynomials. In Section 2, it is shown how common seemingly non-polynomial measurement equations, such as range and DOA, can be transformed into polynomials by adding additional variables and equations to the system.

However, the polynomial systems arising from these types of LS problems are much more difficult than those arising from simple measurement conversion as in Section 7. This can be demonstrated just by considering the use of range-only measurements.

One term in the sum in (114) for a bistatic range-only measurement between Sensors i and j, designated as $\mathcal{C}_{i,j}$, when estimating a full target state is given by

$$\mathcal{C}_{i,j} = \frac{1}{\sigma_r^2}(r^M - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\| - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|)^2 \quad (116)$$

where σ_r^2 is the variance of the measurement. The gradient with respect to the elements of $\mathbf{x}$ is

$$\nabla_{\mathbf{x}}\mathcal{C}_{i,j} = -\frac{2}{\sigma_r^2}(r^M - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\| - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|) \cdot \left(\frac{\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}_i}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\|} + \frac{\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}_j}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|} \right). \quad (117)$$

Introducing the additional equations in terms of r_i, r_j, $\tilde{r}_i$ and $\tilde{r}_j$, one gets

$$r_i^2 - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\|^2 = 0 \quad (118)$$

$$r_j^2 - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|^2 = 0 \quad (119)$$

$$\tilde{r}_i r_i - 1 = 0 \quad (120)$$

$$\tilde{r}_j r_j - 1 = 0 \quad (121)$$

and one can write (117) as

$$\nabla_{\mathbf{x}}\mathcal{C}_{i,j} = -\frac{2}{\sigma_r^2}((\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}_j)(\tilde{r}_j(r^M - r_i) - 1) + (\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}_i)(\tilde{r}_i(r^M - r_j) - 1)) \quad (122)$$

which is a set of simultaneous multivariate polynomial equations. of order three. For a simple localization problem, where the target state only consists of position components in 3D, the gradient (122) represents three equations plus an additional two to four equations depending on whether previous measurements share a common transmitter or receiver in the bistatic path. For the worst-case scenario, where each measurement takes a different bistatic path without any common receivers or transmitters, there is a total of three third-order equations and six second order equations leading to a Bézout's number of 1728. In contrast, using the measurement-conversion approach of Section 7.3.2, one simply has three second-order equations, leading to a Bézout's number of 6. Due to the high computational complexity of the ML/LS method of target-state estimation, detailed examples are not considered here, though a number are presented in [66].

As an example of the difference in speed of the measurement-conversion approach versus the LS method, in [66] a TDOA localization problem was given with

$$\mathbf{l}_1 = \begin{bmatrix} -9.088503107295082 \\ -3.592899795686701 \\ 11.379375304771440 \end{bmatrix} \quad (123)$$

$$\mathbf{l}_2 = \begin{bmatrix} -8.737071884578572 \\ 12.184039601570143 \\ 0.461252502515841 \end{bmatrix} \quad (124)$$

$$\mathbf{l}_3 = \begin{bmatrix} -10.997619107424038 \\ -0.372458566000544 \\ 10.193804421541278 \end{bmatrix} \quad (125)$$

$$\mathbf{l}_4 = \begin{bmatrix} -12.099924003565013 \\ -2.341482530709476 \\ 8.550397573582972 \end{bmatrix} \quad (126)$$

where $\rho_{i,j} = \text{TDOA}_{i,j}/c$ is the range difference between Sensor i and Sensor j and

$$\rho_{1,2} = -6.634727887894795 \quad (127)$$

$$\rho_{1,3} = -1.197770770524833 \quad (128)$$

$$\rho_{1,4} = -2.553916244191934. \quad (129)$$

That is, they use the LS estimation method even though they could use the measurement-conversion method of Section 7.3.1. They reported that it took about 3 s for their program, HOM4PS-2 to solve the problem on a computer with a 3 GHz Intel Core 2 Duo processor. However, when we solved the same polynomial system using the measurement-conversion method of Section 7.3.1 with the PHCpack solver in black-box mode as called from Matlab without parallelization on a 2.93 GHz Intel Xeon processor, the problem was solved in approximately 0.04 s, not counting the time it takes to formulate the problem within Matlab. That is a 7.5-fold improvement in speed just by reformulating the problem, even though a purportedly slower polynomial solver was used.

9. CONCLUSIONS

Track initiation utilizing all common (range, DOA, TDOA, range-rate, frequency-ratio) refraction-free (or corrected) measurement types from one or more sensors can be performed utilizing simultaneous multivariate polynomial solving algorithms. Though many authors focus on directly solving ML/LS problems, as described in Section 8, the systems of multivariate polynomials resulting from such an approach have very high Bézout's numbers and thus are very slow to solve. This paper presents a less computationally demanding approach via the heuristic measurement-conversion algorithm of Section 7, which with the use of cubature integration can produce covariance matrices that are often consistent, barring occasional outliers. Previous polynomial-based techniques have not addressed as wide a variety of problems and have overlooked the computation of covariance matrices, which are essential in multivariate track initiation.

Accurate measurement conversion for track initiation in 3D was demonstrated in three simulation scenarios that do not appear to exist in the literature. These are

1) The use of two bistatic range measurements and one emitter range-rate measurement by two moving sensors to localize a stationary emitter.
2) The use of two TDOA measurements and one bistatic range measurement to localize a cooperative target with both active and passive measurements.
3) The use of four frequency measurements by four moving sensors to localize a non-cooperative stationary emitter.

Multivariate polynomial expressions for equations needed for full target-state estimation (position and velocity) using a number of dynamic scenarios were also provided. It is worth noting that despite occasional outliers for covariance matrix estimates, the algorithms never failed when using Bertini as the multivariate polynomial solver. Thus, the approach in this paper is significantly better suited for use in real systems than the more general heuristic probability-1 homotopy method of [26], which occasionally failed in simulation.

Whereas multivariate polynomial solvers have traditionally been primarily of interest to the robotics community, it is clear that such algorithms have wide application for active and passive target-track initiation. With optimization and parallelization of such methods, it is likely that target-track initiation with generic combinations of multistatic measurement types can be performed in real time at significantly lower hardware cost than using brute-force techniques.

ACKNOWLEDGEMENTS

The author would also like to thank Dan Schonick of NRL for help in assessing the numeric instability of second order cone and semidefinite programming formulations of track-initiation problems in the literature using off-the-shelf solvers, and also Fred Daum of Raytheon for pointing out the work of Pierre Lairez.

APPENDIX A SOLVING A BIVARIATE PAIR OF SECOND-DEGREE EQUATIONS

Here, we present a direct method of solving simultaneous second-order bivariate equations. Though faster, it is often slightly less numerically accurate than using a homotopy-based algorithm for solving for the roots of such a system.

Given a general pair of second-order bivariate polynomials,

$$a_6 + a_5x + a_4x^2 + a_3y + a_2xy + a_1y^2 = 0 \quad (130)$$

$$b_6 + b_5x + b_4x^2 + b_3y + b_2xy + b_1y^2 = 0, \quad (131)$$

one can eliminate x to get a univariate equation of the form

$$c_0 + c_1y + c_2y^2 + c_3y^3 + c_4y^4 = 0 \quad (132)$$

having coefficients (pay attention to parentheses)

$$\begin{aligned} c_0 &= a_6^2b_4^2 + b_6(a_5^2b_4 - a_4a_5b_5 + a_4^2b_6) \\ &\quad + a_6(-a_5b_4b_5 + a_4(b_5^2 - 2b_4b_6)) \end{aligned} \quad (133)$$

$$\begin{aligned} c_1 &= a_5^2b_3b_4 + a_6b_4(2a_3b_4 - a_2b_5) + 2a_4^2b_3b_6 \\ &\quad - a_5(a_6b_2b_4 + a_4b_3b_5 + a_3b_4b_5 + a_4b_2b_6 - 2a_2b_4b_6) \\ &\quad + a_4(-2a_6b_3b_4 + 2a_6b_2b_5 + a_3b_5^2 - 2a_3b_4b_6 - a_2b_5b_6) \end{aligned} \quad (134)$$

$$
\begin{aligned}
c_2 = {} & b_4(a_5^2 b_1 + (a_3^2 + 2a_1 a_6)b_4 - a_5(a_3 b_2 - 2a_2 b_3 + a_1 b_5) \\
& - a_2(a_6 b_2 + a_3 b_5) + a_2^2 b_6) + a_4^2(b_3^2 + 2b_1 b_6) \\
& - a_4(-a_6 b_2^2 + a_5 b_2 b_3 + 2a_6 b_1 b_4 + 2a_3 b_3 b_4 + a_5 b_1 b_5 \\
& - 2a_3 b_2 b_5 + a_2 b_3 b_5 - a_1 b_5^2 + a_2 b_2 b_6 + 2a_1 b_4 b_6)
\end{aligned} \tag{135}
$$

$$
\begin{aligned}
c_3 = {} & 2a_4^2 b_1 b_3 - a_4(a_5 b_1 b_2 - a_3 b_2^2 + a_2 b_2 b_3 + 2a_3 b_1 b_4 \\
& + 2a_1 b_3 b_4 + a_2 b_1 b_5 - 2a_1 b_2 b_5) \\
& + b_4(-a_1 a_5 b_2 + a_2^2 b_3 + 2a_1 a_3 b_4 \\
& + a_2(2a_5 b_1 - a_3 b_2 - a_1 b_5))
\end{aligned} \tag{136}
$$

$$
\begin{aligned}
c_4 = {} & a_4^2 b_1^2 + b_4(a_2^2 b_1 - a_1 a_2 b_2 + a_1^2 b_4) \\
& + a_4(-a_2 b_1 b_2 + a_1(b_2^2 - 2b_1 b_4)).
\end{aligned} \tag{137}
$$

Though explicit formulae are available for solving fourth-order polynomial equations, one would expect many of them to be numerically unstable, because, as is shown in [36, Ch. 24.3.3], the explicit solution of cubic polynomial equations is numerically unstable. For simplicity, we solve the equation using the roots function in Matlab, which finds the eigenvalues of a matrix whose eigenvalues coincide with the roots of the equation.

Given y, one can substitute back into (130) to get x via the quadratic formula:

$$
x = \frac{-a_5 - a_2 y \pm \sqrt{(a_5 + a_2 y)^2 - 4a_4(a_6 + a_3 y + a_1 y^2)}}{2a_4}. \tag{138}
$$

Substituting into (131), one similarly gets

$$
x = \frac{-b_5 - b_2 y \pm \sqrt{(b_5 + b_2 y)^2 - 4b_4(b_6 + b_3 y + b_1 y^2)}}{2b_4}. \tag{139}
$$

The correct value of x to use with y solved from (132) is the one that is common to both equations. With finite precision, just choose the pair of solutions that are closest and average them. Note, however, that such a method will fail to find all roots if there are repeated roots in y. For example, if $(1,2)$ and $(2,2)$ were both solutions, then only one of them would be found, even if it is known that the root in y is repeated, because both y values would map to the same x value with the above method. Thus, if two y values are equal, there must be a special case where the repeated one takes the second closest pair of solutions.

As an example, consider the system of equations

$$
0 = -x^2 + 2xy + y^2 + 5x - 3y - 4 \tag{140}
$$

$$
0 = x^2 + 2xy + y^2 - 1. \tag{141}
$$

The exact roots are $(4,-5)$, $(1,0)$, $(3,-2)$, $(0,-1)$. Using the above method in Matlab (which uses double-precision arithmetic), one obtains the correct solutions, where the greatest magnitude difference in any component from the true value is about 7.1×10^{-15}. When solving using PHCpack, which unlike Bertini does not offer the option of extended precision arithmetic, the greatest magnitude difference in any component from the true value is about 2.8×10^{-17}.

APPENDIX B THE CUBATURE POINTS USED IN THE SIMULATIONS

As mentioned in Section 3, cubature points and weights are needed to efficiently numerically evaluate integrals involving polynomials to a high degree. Here, we choose to use the cubature points of $E_n^{r^2}$ 5-3 on page 317 of [65]. The points scaled for a standard normal distribution are also used in [23]. Here, we reproduce the listing of the points from [23] for the standard normal distribution:

Fifth-Order Cubature Points and Weights

Weight (ω_i)	Point ($\boldsymbol{\xi}_i$)
$\dfrac{4}{(d+2)^2}$	$[\pm a]$
$\dfrac{(d-2)^2}{2^d(d+2)^2}$	$(\pm b, \pm b, \ldots, \pm b)$

The points are given as shown above, where

$$
a = \sqrt{\frac{d+2}{2}} \qquad b = \sqrt{\frac{d+2}{d-2}}, \tag{142}
$$

and d is the dimensionality of the points generated. The $\pm$ indicates that all possible combinations of negative and positive elements should be used. The bracket notation for the first set of points indicates that all possible vectors with that single nonzero element should be generated. There are $2d$ points of the first type and 2^d points of the second type. These points can be used for integrals involving an arbitrary Gaussian weighting with $d > 2$.

To use the above points and weights with a normal distribution having mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, each of the points should be transformed using

$$
\boldsymbol{\xi}_i^{\text{transformed}} = \boldsymbol{\mu} + \Sigma^{1/2}\boldsymbol{\xi}_i \tag{143}
$$

where the square root of the covariance matrix is taken to be a lower-triangular Cholesky decomposition of the matrix rather than a true square root (the `chol` command in Matlab with the '`lower`' option).

REFERENCES

[1] S. Ayazgök
"Target tracking and sensor placement for Doppler-only measurements,"
Master's thesis, Middle East Technical University, Ankara, Turkey, Sep. 2015.

[2] Y. Bar-Shalom and H. Chen
"IMM estimator with out-of-sequence measurements,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 41, no. 1, pp. 90–98, Jan. 2005.

[3] Y. Bar-Shalom, H. Chen, and M. Mallick
"One-step solution for the multistep out-of-sequence-measurement problem in tracking,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 40, no. 1, pp. 27–37, Jan. 2004.

[4] Y. Bar-Shalom, X. R. Li, and T. Kiruabarajan
Estimation with Applications to Tracking and Navigation.
New York: Wiley Interscience, 2001.

[5] Y. Bar-Shalom, R. Osborne III, P. Willett, and F. Daum
"CRLB for likelihood functions with parameter dependent support,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 50, no. 3, pp. 2399–2405, Jul. 2014.

[6] D. J. Bates, D. A. Brake, and M. E. Niemerg
"Paramotopy: Parameter homotopies in parallel,"
ACM Transactions on Mathematical Software, Mar. 2013, paper submitted, preprint available. [Online]. Available: http://www.math.colostate.edu/~bates/preprints/BBN_paramotopy.pdf.

[7] D. J. Bates, A. J. Newell, and M. Niemerg
"BertiniLab: A MATLAB interface for solving systems of polynomial equations,"
Numerical Algorithms, vol. 71, no. 1, pp. 229–244, Jan. 2016.

[8] D. J. Bates, A. J. Sommese, J. D. Hauenstein, and C. W. Wampler
Numerically Solving Polynomial Systems with Bertini.
Philadelphia: Society for Industrial and Applied Mathematics, 2013.

[9] D. J. Bates and F. Sottile
(2011, Oct.)
Khovanskii-rolle continuation for real solutions.

[10] C. Beltrén and A. Leykin
"Certified numerical homotopy tracking,"
Experimental mathematics, vol. 21, no. 1, 2012.

[11] K. Berntorp, A. Robertsson, and K.-E. Årzén
"Rao-Blackwellized particle filters with out-of-sequence measurement processing,"
IEEE Transactions on Signal Processing, vol. 62, no. 24, pp. 6454–6467, 15 Dec. 2014.

[12] D. A. Brake, D. J. Bates, W. Hao, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler
"Bertini_real: Software for one- and two-dimensional real algebraic sets,"
in *Proceedings of the 4th International Congress on Mathematical Software*, vol. 8592, Seoul, South Korea, 5–9 Aug. 2014, pp. 175–182.

[13] B. Buchberger
"Gröbner bases: A short introduction for systems theorists,"
in *Computer Aided Systems Theory—EUROCAST 2001*, R. Moreno-D´az, B. Buchberger, and J. L. Freire, Eds. Springer, 2001, vol. 2178, pp. 1–19.

[14] P. Bürgisser and F. Cucker
Condition.
Heidelberg: Springer, 2013.

[15] M. Caruso, P. Lombardo, and M. Sedehi
"A weighted least squares approach for multi-target Doppler-only localization,"
in *Proceedings of the IEEE Radar Conference*, Ottawa, Canada, 29 Apr.–3 May 2013.

[16] Y. T. Chan and J. J. Towers
"Passive localization from Doppler-shifted frequency measurements,"
IEEE Transactions on Signal Processing, vol. 40, no. 10, pp. 2594–2598, Oct. 1992.

[17] ———
"Sequential localization of a radiating source by Doppler-shifted frequency measurements,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 28, no. 4, pp. 1084–1090, Oct. 1992.

[18] Y.-T. Chan and F. L. Jardine
"Target localization and tracking from Doppler-shift measurements,"
IEEE Journal of Oceanic Engineering, vol. 15, no. 3, pp. 251–257, Jul. 1990.

[19] R. Cools
"An encyclopaedia of cubature formulas,"
Journal of Complexity, vol. 19, no. 3, pp. 445–453, Jun. 2003.

[20] D. F. Crouse
"On implementing 2D rectangular assignment algorithms,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 52, no. 4, pp. 1679–1696, Aug. 1016.

[21] ———
"Developing a real-time track display that operators do not hate,"
IEEE Transactions on Signal Processing, vol. 59, no. 7, pp. 3441–3447, Jul. 2011.

[22] ———
"Advances in displaying uncertain estimates of multiple targets,"
in *Proceedings of SPIE: Signal Processing, Sensor Fusion, and Target recognition XXII*, Baltimore, MD, 29 Apr. 2013.

[23] ———
"Basic tracking using 3D monostatic and bistatic measurements,"
IEEE Aerospace and Electronic Systems Magazine, vol. 29, no. 8, Part II, pp. 4–53, Aug. 2014.

[24] ———
"Basic tracking using 3D monostatic and bistatic measurements in refractive environments,"
IEEE Aerospace and Electronic Systems Magazine, vol. 29, no. 8, Part II, pp. 54–75, Aug. 2014.

[25] ———
"An overview of major terrestrial, celestial and temporal coordinate systems,"
Naval Research laboratory, Washington, DC, Tech. Rep., 2014, release Pending.

[26] ———
"Target track initiation in difficult scenarios using probability-1 homotopy methods and cubature integration,"
in *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, 5–12 Mar. 2016.

[27] P. Dreesen
"Back to the roots: Polynomial system solving using linear algebra,"
Ph.D. dissertation, Katholieke Universiteit Leuven, Leuven, Flanders, Belgium, Sep. 2013.

[28] EUROCONTROL.
(2012, Feb.)
EUROCONTROL CASCADE programme. [Online]. Available: https://www.eurocontrol.int/sites/default/files/article/content/documents/nm/surveillance/cascade/cascade-programme-tryipt-19-03-2012.pdf.

[29] Federal Aviation Administration.
(2014, Jul.)
Wide area multilateration WAM). [Online]. Available: https://www.faa.gov/nextgen/library/media/getSmart_WAM.pdf.

[30] Federal Communications Commission.
(2015, 3 Apr.)
Wireless E911 location accuracy requirements. [Online]. Available: https://www.federalregister.gov/articles/2015/03/04/2015-04424/wireless-e911-location-accuracy-requirements.

[31] R. P. Feynman, R. B. Leighton, and M. Sands
The Feynman Lectures on Physics.
Menlo Park, CA: Addison-Wesley Publishing Company, 1963, vol. 1.

[32] Y. Guan and J. Verschelde
"PHCLab: A MATLAB/Octave interface to PHCpack,"
in *Software for Algebraic Geometry*, D. N. Arnold and A. Scheel, Eds. New York: Springer, 2008, vol. 148, pp. 15–32.

[33] M. B. Guldogan, D. Lindgren, F. Gustafsson, H. Habberstab, and U. Orguner
"Multi-target tracking with PHD filter using Doppler-only measurements,"
Digital Signal Processing, vol. 27, pp. 1–11, Apr. 2014.

[34] E. Hanusa, D. Krout, and M. R. Gupta
"Estimation of position from multistatic Doppler measurements,"
in *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, United Kingdom, 26–29 Jul. 2010.

[35] ———
"Clutter rejection by clustering likelihood-based similarities,"
in *Proceedings of the 14th International Conference on Information Fusion*, Chicago, IL, 5–8 Jul. 2011.

[36] N. J. Higham
Accuracy and Stability of Numerical Algorithms.
Philadelphia: SIAM, 1996.

[37] I. Inc.
(2016) IHS aerospace, defense and security.
[Online]. Available: https://janes.ihs.com.

[38] D. R. Jones, C. D. Peritunen, and B. E. Stuckman
"Lipschitzian optimization without the Lipschitz constant,"
Journal of Optimization Theory and Application, vol. 79, no. 1, pp. 157–181, Oct. 1993.

[39] S. Joshi
"Multi-target tracking via nonlinear least squares using doppler measurements from a passive radar system,"
Master's thesis, Georgia Institute of technology, May 2007.

[40] Y. Kalkan and B. Baykal
"MIMO radar target localization by using Doppler shift measurement,"
in *Proceedings of the 6th European Radar Conference*, Rome, Italy, 30 Sep.–2 Oct. 2009, pp. 489–492.

[41] ———
"Target localization methods for frequency-only MIMO radars,"
in *Proceedings of the 7th European Radar Conference*, Paris, France, 30 Sep.–1 Oct. 2010, pp. 396–399.

[42] ———
"Target localization and velocity estimation methods for frequency-only MIMO radars,"
in *Proceedings of the IEEE Radar Conference*, Kansas City, MO, 23–27 May 2011, pp. 458–463.

[43] Y. Kalman and B. Baykal
"Frequency based target localization methods for MIMO radar,"
in *Proceedings of the IEEE 19th Signal Processing and Communications Applications Conference*, Antalya, Turkey, 20–22 Apr. 2011, pp. 74–77, in Turkish.

[44] P. Krysik, M. Wieglo, J. Misiurewicz, and A. Kurowska
"Doppler-only tracking in GSM-based passive radar,"
in *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, Spain, 7–10 Jul. 2014.

[45] Y.-C. Kuo, W.-W. Lin, and S.-T. Yau
"A novel efficient homotopy continuation method in tracking,"
Communications in Information and Systems, vol. 14, no. 1, pp. 57–78, 2014.

[46] T.-L. Lee, S.-S. Lin, W.-W. Lin, S.-T. Yau, and J. Zhu
"Polynomial calculations in Doppler tracking,"
Communications in Information and Systems, vol. 12, no. 2, pp. 157–184, 2012.

[47] I. Levesque and J. Bondaryk
"Performance issues concerning Doppler-only localization of submarine targets,"
NATO Saclant undersea Research Center, Tech. Rep. SR-325, Jul. 2000.

[48] T. Y. Li, T. Sauer, and J. A. Yorke
"The cheater's homotopy: An efficient procedure for solving systems of polynomial equations,"
SIAM Journal on Numerical Analysis, vol. 26, no. 5, pp. 1241–1251, 1989.

[49] M. Liang, D. Y. Kim, and X. Kai
"Multi-Bernoulli filter for target tracking with multi-static Doppler only measurement,"
Signal Processing, vol. 109, pp. 102–110, Mar. 2015.

[50] J. Liu, Y. D. Zhang, and M. G. Amin
"Target localization in moving radar platform exploiting range and Doppler information through semidefinite relaxation,"
in *Proceedings of SPIE: Wireless Sensing, Localization, and Processing V*, vol. 7706, Orlando, FL, 5 Apr. 2010.

[51] D. G. Mixon
"Doppler-only multistatic radar,"
Ph.D. dissertation, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, Mar. 2006.

[52] M. Orton and A. Marrs
"Particle filters for tracking with out-of-sequence measurements,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 41, no. 2, pp. 693–702, Apr. 2005.

[53] C. H. Papadimitriou
Combinatorial Optimization: Algorithms and Complexity.
Englewood Cliffs, NJ: Prentice-Hall Inc, 1982.

[54] F. Papi
"Multi-sensor δ-GLMB filter for multi-target tracking using Doppler only measurements,"
in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, 25–30 Mar. 2015, pp. 83–89.

[55] P. A. Parillo
"Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization,"
Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 2000. [Online]. Available: http://www.mit.edu/˜parrilo/pubs/files/thesis.pdf.

[56] P. A. Parilo and B. Sturmfels
"Minimizing polynomial functions,"
in *Proceedings of the DIMACS Workshop Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science*, vol. 60, DIMACS Center, Piscataway, NJ, 12–16 Mar. 2001.

[57] J. S. Picard and A. J. Weiss
"Time-delay and Doppler-shift based geolocation by semidefinite programming,"
in *Proceedings of the 20th European Signal Processing Conference*, Bucharest, Romania, 27–31 Aug. 2012, pp. 1189–1193.

[58] B. Ristic and A. Farina
"Joint detection and tracking using multi-static Doppler-shift measurements,"
in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, 25–30 Mar. 2012, pp. 3881–3884.

[59] C. M. Rose
"Apparatus and method for locating an emitter using RF carrier of PRF measurement ratios,"
U.S. Patent 6 163 297, Dec. 19, 2007.

[60] I. Shames, A. N. Bishop, M. Smith, and B. D. O. Anderson
"Analysis of target velocity and position estimation via Doppler-shift measurements,"
in *Proceedings of the Australian Control Conference*, Melbourne, Australia, 10–11 Nov. 2011, pp. 507–512.

[61] ———
"Doppler shift target localization,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 49, no. 1, pp. 266–276, Jan. 2013.

[62] X. Shen, E. Song, Y. Zhu, and Y. Luo
"Globally optimal distributed kalman fusion with local out-of-sequence-measurement updates,"
IEEE Transactions on Automatic Control, vol. 54, no. 8, pp. 1928–1934, Aug. 2009.

[63] A. J. Sommese, J. Verschelde, and C. W. Wampler
"Introduction to numerical algebraic geometry,"
in *Solving Polynomial Equations*, A. Dickenstein and I. Z. Emiris, Eds. Springer, 2005, ch. 8, pp. 301–337.

[64] A. J. Sommese and C. W. Wampler II
The Numerical Solution of Systems of Polynomials Arising in Engineering and Science.
New Jersey: World Scientific, 2005.

[65] A. Stroud
Approximate Calculation of Multiple Integrals.
Edgewood Cliffs, NJ: Prentice-Hall, Inc., 1971.

[66] Y. Tian and Y. Cheng
"Solving optimal positioning problems via homotopy continuation,"
in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, National Harbor, MD, 13–17 Jan. 2014.

[67] J. Verschelde
"Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation,"
ACM Transactions on Mathematical Software, vol. 25, no. 2, pp. 251–276, Jun. 1999.

[68] A. Wallack, I. Z. Emiris, and D. Manocha
"MARS: A Maple/Matlab/C resultant-based solver,"
in *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, Rostock, Germany, 13–15 Aug. 1998, pp. 244–251.

[69] L. T. Watson, S. C. Billups, and A. P. Morgan
"Algorithm 652: HOMPACK: A suit of codes for globally convergent homotopy algorithms,"
ACM Transactions on Mathematical Software, vol. 13, no. 3, pp. 281–310, Sep. 1987.

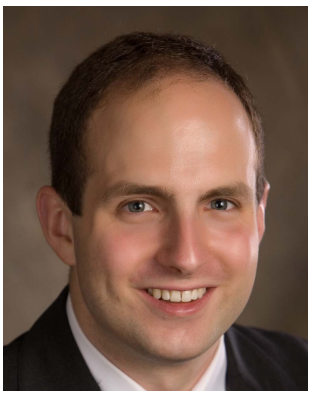
[70] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walkder
"Algorithm 777: HOMPACK90: A suite of Fortran 90 codes for globally convergent homotopy algorithms,"
ACM Transactions on Mathematical Software, vol. 23, no. 4, pp. 514–549, Dec. 1997.

[71] R. J. Webster
"An exact trajectory solution from Doppler shift measurements,"
IEEE Transactions on Aerospace and Electronic Systems, vol. AES-18, no. 2, pp. 249–252, Mar. 1982.

[72] M. P. Williams
"Solving polynomial equations using linear algebra,"
Johns Hopkins Technical Digest, vol. 28, no. 4, pp. 354–363, 2010.

[73] S. M. Wise, A. J. Sommese, and L. T. Watson
"Algorithm 801: POLSYS_PLP: A partitioned linear product homotopy code for solving polynomial systems of equations,"
ACM Transactions on Mathematical Software, vol. 26, no. 1, pp. 176–200, Mar. 2000.

[74] S. Zhang and Y. Bar-Shalom
"Optimal update with multiple out-of-sequence measurements with arbitrary arriving order,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 4, pp. 3116–3132, Oct. 2012.



David Frederic Crouse received B.S., M.S., and Ph.D. degrees in Electrical Engineering in 2005, 2008, and 2011 from the University of Connecticut (UCONN). He also received a B.A. degree in German from UCONN for which he spent a year at the Ruprecht-Karls Universität in Heidelberg, Germany. He is the recipient of the 2016 Young Investigator Award from the International Society on Information Fusion and the recipient of a 2015 Alan Berman Research Publication award from the Naval Research Laboratory for the paper "Basic Tracking Using Nonlinear Continuous-Time Dynamic Models."

He is currently employed at the Naval Research Laboratory in Washington, D.C. and serves as an associate editor at the IEEE Aerospace and Electronic Systems Magazine. His interests lie in the areas of stochastic signal processing and tracking.

Mutual Information for Optimal Asset Allocation

K. C. CHANG
ZHI TIAN

Dynamic asset allocation in financial investment with an optimal equity growth principle based on mutual information in communication theory is considered. Specifically, the asset allocation formula using Kelly's criteria derived from channel capacity of a binary symmetric channel is developed. The goal is to determine the optimal fraction of equity to be invested between a risk-free asset and a risky asset in a repeated trading activity. The analytical operating curve to predict trading performance is provided. An extension for dynamic multi-asset allocation is also presented. An out-of-sample simulation based on historical market data demonstrates the effectiveness of the methodology.

Manuscript received December 21, 2016; revised June 8, 2017; released for publication December 8, 2017.

Refereeing of this contribution was handled by Ramona Georgescu.

Authors' addresses: K. C. Chang, Dept. of Systems Engineering and Operations Research, George Mason University, Fairfax, VA, U.S.A. (E-mail: kchang@gmu.edu). Z. Tian, Dept. of Electrical and Computer Engineering, George Mason University, Fairfax, VA, U.S.A. (E-mail: ztain1@gmu.edu).

1. INTRODUCTION

In modern financial investment world, dynamic asset allocation allows frequent rebalancing of portfolio over time in order to achieve certain objectives. Investors have to sift through a large amount of data in order to analyze the market behavior, predict future market directions, and make sound trading decisions. Given the complexity of the markets and the high stake of trading decisions, financial engineering and risk analysis have emerged as an important research field [1–2].

In particular, one of the critical questions is how to allocate the capital optimally among various correlated risky assets in order to achieve the highest overall return under a defined risk level. This type of portfolio management has become one of the most important elements in practical investment management. The main goal of asset allocation is to develop a long-term risk and return expectation curve for the portfolio and to establish an operating point for each individual investor to balance between the expected return and risk based on his or her own preferences. Traditionally, the process of asset allocation is to identify fundamentally different core asset classes (stocks, bonds, real estate, commodity, etc.) and decide what portion of the capital to invest in each class in order to compose an overall balanced portfolio.

To model and analyze financial data, many mathematical and statistical methods have been applied for quantitative analysis, such as time series analysis, regression analysis, machine learning methods, and Monte Carlo simulations [1–2]. In the financial markets, there are two main traditional approaches for market analysis and stock selection. Fundamental analysis looks into economic factors such as financial statements and market competitiveness to make subjective judgments on the qualitative relationship between equity values and expected market returns, whereas technical analysis uses quantitative historical data of a security such as trading patterns and volume to predict its future price movement [3–8].

Alternative to these traditional approaches, modern quantitative analysis applies complex mathematical models to analyze portfolio risk and develop algorithm trading and arbitraging strategies [2]. This paper adopts the concept of the classical modern portfolio theory (MPT) [9–11], which provides a foundation for explicit risk-reward trade-off analysis. In MPT, a portfolio is consisted of a set of correlated assets each with its own expected return (defined as annual average percentage return) and risk (defined as equity return standard deviation on an annual basis). The goal of this "mean-variance" (MV) approach is to allocate equity among the assets optimally so that the expected portfolio return can be maximized given a defined risk level or vice versa (the overall portfolio risk can be minimized given a desired return level). MPT develops a set of optimal asset allocation policies by optimizing the allocation

among available assets subject to risk constraints. These policies form an "efficient frontier" which allows the decision maker to select his/her own operating point on the curve to trade off between risk and return. When an asset is selected with a certain allocation, it implies that specific trading actions for the underlined asset would be taken.

Traditionally MPT-based performance prediction is made based on some idealized dynamic model for the volatility of the underlying asset [11–12], which does not work well in real world. Another approach developed originally by Kelly [13] shows that the optimal long run asset allocation strategy can be obtained by maximizing the expected logarithm of the portfolio value over each time step. This strategy has been proven in different ways [14–15] and has been successfully applied in financial markets [16–17]. An explicit connection between Kelly's criterion and the information theory has also been discussed in [18].

In real world financial market, characteristics of asset returns change rapidly over time such that an investor needs to develop a dynamic asset allocation/rebalancing strategy adaptive to the market environment [5, 19]. In portfolio rebalancing, when an asset is selected with a certain allocation, it implies that specific trading actions for the underlined asset would be taken. Typically, equity-trading strategies are simple buy or sell actions, which are often used for short-term trading.

To account for the complex dynamics of modern market behavior, this paper presents an approach for performance prediction and evaluation by incorporating historical market data in simulated trading. Specifically, the equity index futures and options (S&P 500) data from 1990 to 2016 are used to test the allocation strategy and trading decisions. In addition, instead of defining risk using volatility as in MPT, a different risk metric based on probability of ruin or "draw down" is computed to derive the corresponding efficient frontier.

In this paper, we advocate the development of optimal asset allocation strategies using Kelly's formula that was derived based on mutual information of a binary symmetric channel in communication theory. For clear exposition, we start with a simple trading scenario where the entire equity is allocated between two assets: the risk-free asset (money market or fixed income) and a risky equity asset (S&P index futures and options). In fact, the trading choices examined here are intentionally simplified so that we can clearly illustrate the analytical performance prediction with tradeoffs between risk and return.

The goal is to identify the optimal fraction of the equity to be allocated for trading in order to achieve the highest long-term return given a risk constraint. The results based on extended simulated trading with index options are compared with the analytical performance prediction. In the trading process, in order to obtain a more realistic options price, we develop an analytical model based on "implied volatility"[1] and adjust the option price accordingly. The results have been validated against the historical market data and proved to be reasonably accurate.

The preliminary version of this paper was presented in the 19th International Conference on Information Fusion [20]. We have thoroughly re-organized and revised the original paper with additional contributions. In particular, we develop a dynamic asset allocation strategy for portfolio with multiple correlated assets. Similar to Kelly's approach for single asset, the goal is to maximize the long-run portfolio growth rate over many investment cycles. In addition, we conduct extensive out-of-sample simulations and show that the resulting strategy outperforms the traditional MV approach in expected return but with higher volatility. This strategy can be considered as an alternative approach for the investor to trade off between risk and reward.

The paper is organized as follows. Section 2 describes the optimal asset allocation methodologies and the implications on risk and return. Section 3 presents the selected trading strategies and the options pricing model. Section 4 summarizes the simulation results and the performance analysis for single asset allocation. Section 5 presents Kelly's approach as a multi-asset dynamic allocation strategy and compares its performance with a naïve strategy and MV tangency portfolio. Conclusions and future research are presented in Section 6.

2. OPTIMAL ASSET ALLOCATION

The goal of constructing an optimal portfolio is to maximize the investor's return with a given risk level. Consider a portfolio consisting of multiple assets. The log return of each asset in the portfolio over an investment period is defined as $\eta_i(k) = \log[A_{i,k}/A_{i,k-1}]$ where $A_{i,k}$ is the value of the asset i at time k. Assume that the single period log returns are independent and normally distributed. Then the return of the asset,

$$R_i(k) = (A_{i,k} - A_{i,k-1})/A_{i,k-1} = e^{\eta_i(k)} - 1 \qquad (1)$$

is log-normally distributed.

2.1. Mean-Variance Approach

In modern portfolio theory (MPT) [21–22], a portfolio consists of a set of correlated assets each with its own expected return (defined as annualized percentage return) and risk (defined as equity standard deviation on an annual basis). For example, let $\mathbf{R} = [R_1 \cdots R_N]^T$ be the return of N correlated assets, $E(R) = \boldsymbol{\mu} = [\mu_1 \cdots \mu_N]^T$ be their expected return over an investment period, and Σ be the covariance matrix of $\mathbf{R}$. Denote $\boldsymbol{\omega} = [\omega_1 \cdots \omega_N]^T$ as the asset weights in the portfolio such that $\sum_{i=1}^{N} \omega_i = \mathbf{1}^T\boldsymbol{\omega} = 1$. The expected portfolio return is therefore $\sum_{i=1}^{N} \omega_i\mu_i = \boldsymbol{\omega}^T\boldsymbol{\mu}$ and the corresponding variance is $\sigma_P^2 = \boldsymbol{\omega}^T\Sigma\boldsymbol{\omega}$.

[1]The volatility of the option implied by the market price with a theoretical option value model.

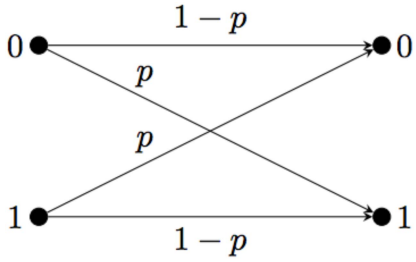


Fig. 1. A Binary Symmetric Channel

Typical investment goal is to allocate capitals among the assets optimally so that the expected portfolio return can be maximized given a defined risk level; or alternatively, the overall portfolio risk can be minimized given a desired return level. For example, if there is no risk-free asset, a typical asset allocation problem can be formulated as the following constrained optimization problem:

$$\text{Min}_{\omega}\, \boldsymbol{\omega}^T \Sigma \boldsymbol{\omega} \text{ subject to } \boldsymbol{\omega}^T \mathbf{1} = 1 \text{ and } \boldsymbol{\omega}^T \boldsymbol{\mu} = \mu_P \quad (2)$$

With this "mean-variance" (MV) approach, an efficient frontier can be constructed where for each return level a portfolio can be derived with minimum risk (variance).

2.2. Binary Symmetric Channel and the Kelly Criteria

In communication theory, a binary symmetric channel (BSC) is defined in Figure 1, where the binary input experiences a cross-over probability (probability of error) p to yield the binary output. In BSC, the channel capacity is defined by

$$C_{\text{BSC}} = 1 - H(p) = 1 + p\log(p) + (1-p)\log(1-p) \quad (3)$$

which is equivalent to the *maximum mutual information*.

The Kelly criterion was originally developed [13] based on the channel capacity concept in communication theory. Start with a single asset portfolio, the specific question addressed was how to allocate the asset optimally for an investment/betting opportunity in order to maximize the expected long-term equity growth rate. The only requirement is that the investment opportunity needs to have a positive expected return (i.e., with a winning edge).

Consider a specific investment (bet). Let p represent its winning probability, b represent the expected return per unit bet for a winning trade, f represent the fraction of the equity allocated to the investment (the remaining $1-f$ sits on the side line). The number of winning and losing trades over n bets is denoted by W and L respectively, with $W + L = n$. Apparently, W and L approach pn and $(1-p)n$ respectively when n is large. Let X_0 and X_n denote the initial and the final amount of the equity after n bets, where X_n/X_0 is called the terminal wealth ratio (TWR). Then the expected log growth rate of the equity per trade can be written as [20],

$$\begin{aligned} g(f) &= E\left\{\log\left[\frac{X_n}{X_0}\right]^{1/n}\right\} \\ &= E\left\{\frac{W}{n}\log(1+bf) + \frac{L}{n}\log(1-f)\right\} \end{aligned} \quad (4)$$

It can be easily shown that the optimal f that maximize $g(f)$ is,

$$f^* = \frac{bp-(1-p)}{b} \quad (5)$$

which is called the Kelly's formula. Specifically, when $b = 1$, $f^* = 2p - 1$, Equation (4) converts to (3), the BSC channel capacity or the *maximum mutual information* [14–15]. In other words, with the optimal fraction allocation based on the Kelly's formula (5), the expected log growth rate of the equity per trade with a winning probability p converges to the maximum mutual information of a binary symmetric channel.

2.3. Return and Risk Trade-off

Equation (4) shows that the log equity of a portfolio is expected to grow at a rate of $g(f)$ on a per-trade basis. For example, if $b = 1$, $p = 0.6$, then the optimal Kelly's fraction is $f^* = 0.2$ and the highest expected gross return per unit bet per trade is $bp - (1-p) = 0.2$. The corresponding expected growth rate of the equity is $R = e^{g(f^*)} - 1 = 0.02$ on a per-trade basis. In other words, with the optimal Kelly criterion, the equity is expected to grow on an average of 2% after each trade if 20% of the equity is allocated for each trade where the amount of potential gain or loss are the same for each trade and the winning probability of each trade is 0.6.

While the Kelly's formula provides an optimal allocation for each trade to maximize the long-term TWR, the potential risk could also be high. Traditionally, trading risk is assessed by the volatility (STD) of the equity return [9]. However, it has been recognized that a more appropriate measure of risk for an investment is draw down (DD) or probability of ruin. Draw down is defined as the percentage of the equity loss from a peak to a subsequent bottom within an investment cycle and the probability of ruin is defined as the probability of losing a certain percentage of the equity at the end of an investment cycle. While return and risk go against each other, it is obviously desirable for an investment strategy to achieve a high return with a limited draw down or low probability of ruin.

It has been shown that with Kelly's investment strategy, the probability of ruin at the terminal stage (having a draw down of $d = 1 - \bar{d}$) after an extensive trading period can be approximated by [15]

$$P\{(X_n/X_0) \le \bar{d}\} = \bar{d}^{(2/c)-1} \quad (6)$$

where the allocation for each trade is $f = cf^*$ with $0 < c < 2$. For example, with $f^* = 0.5$, $c = 1$, and $\bar{d} = 0.33$,

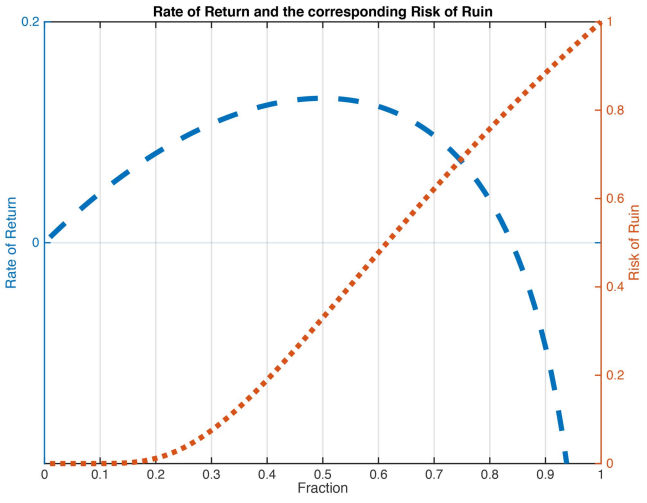


Fig. 2. Risk and Return Trade-off as a function of Investment Fraction

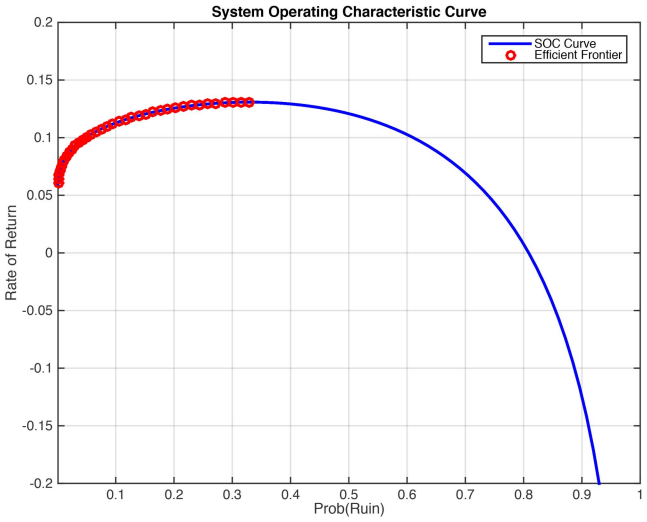


Fig. 3. System Operating Curve (SOC) for Return vs. Risk

$P\{(X_n/X_0) \leq 0.33\} = 0.33$ when $n \gg 1$. Namely, there is a 33% chance that the terminal equity after a large number of trades is less than 33% (a draw down of more than 67%) of the initial equity if 50% (optimal Kelly fraction) of the equity was allocated for each trade. On the other hand, if $c = 0.5$, $f = 0.5f^* = 0.25$ (half of Kelly), then the probability of ruin reduces to $P\{(X_n/X_0) \leq 0.33\} = 0.33^3 \approx 0.036$.

For example, with $p = 0.75$, $b = 1$, and $f^* = 0.5$, Figure 2 compares the normalized rate of return to the risk of ruin over the range of fraction f from 0 to 1 for equity allocation in each trade [20]. As can be seen from the figure, when the allocation follows the Kelly's suggestion (namely, $f = 0.5$), the rate of return is at maximum (0.13) while the probability of ruin (with $\bar{d} = 0.33$), is about 0.33. When half of Kelly (0.25) is applied, the rate of return lowers to 0.095 while the risk of ruin reduces to under 0.04. This also suggests that a systematic approach can be developed for an investment strategy where a system operating curve (SOC) can be established to predict risk and reward performance at different operating points and ultimately allow an investor to choose a specific point to fit his/her own risk preference.

To illustrate, Figure 3 shows the corresponding SOC curve derived from Figure 2 [20]. Each point on the curve represents an operating point with a specific fraction of equity being allocated for each trade, where the peak of the curve corresponds to the suggested optimal Kelly's fraction (50%). The left portion of the curve represents the operating points where higher return can be achieved by taking a higher risk. They can be considered as the "investment efficient frontier" where investors could pick and choose an operating point based on their own preference. It can be seen from the figure that beyond the efficient frontier, taking higher risk will negatively reduce the expected rate of return. This is due to the "over aggressiveness" by investing more than the optimal Kelly's fraction on each trade and is clearly not desirable.

3. TRADING STRATEGIES WITH SINGLE ASSET ALLOCATION

Having explained the basic principle of Kelly's formula, this section demonstrates how it can be applied for optimal allocation with a single asset in practice where a number of statistical characteristics of the trading asset need to be acquired first. Specifically, the winning probability, the average gain of winning trades, and the average loss of the losing trades need to be estimated.

3.1. Trading S&P Futures and Options

S&P futures and their options are selected as the asset for the portfolio. They are traded in many financial markets, including the Chicago Mercantile Exchange (CME) [23] and the CME electronic GLOBEX platform [24]. They are one of the most liquid equity index products traded in the world. The two most commonly traded options are the plain vanilla put and call options.[2] Both S&P futures and the corresponding options are extremely liquid and popular.

One could long or short the futures contracts or the option contracts depending on the goals of his/her trading strategies. By writing (selling) the put options when the market is expected to go higher would result in the options expiring worthlessly and therefore the seller could keep the collected premium. Similarly, the seller could keep the premium collected by writing the call options if the market does not move up beyond the strike price. However, while the potential loss of buying

[2]A put option gives the owner of the option the right, but not the obligation, to sell an asset (the underlying) at a specific price (the strike), by a pre-determined date (the expiration or maturity date) to the seller (or "writer") of the option. A call option gives the buyer of the option the right, but not the obligation, to buy an agreed quantity of the underlying from the seller of the option before the expiration date at a given strike price.

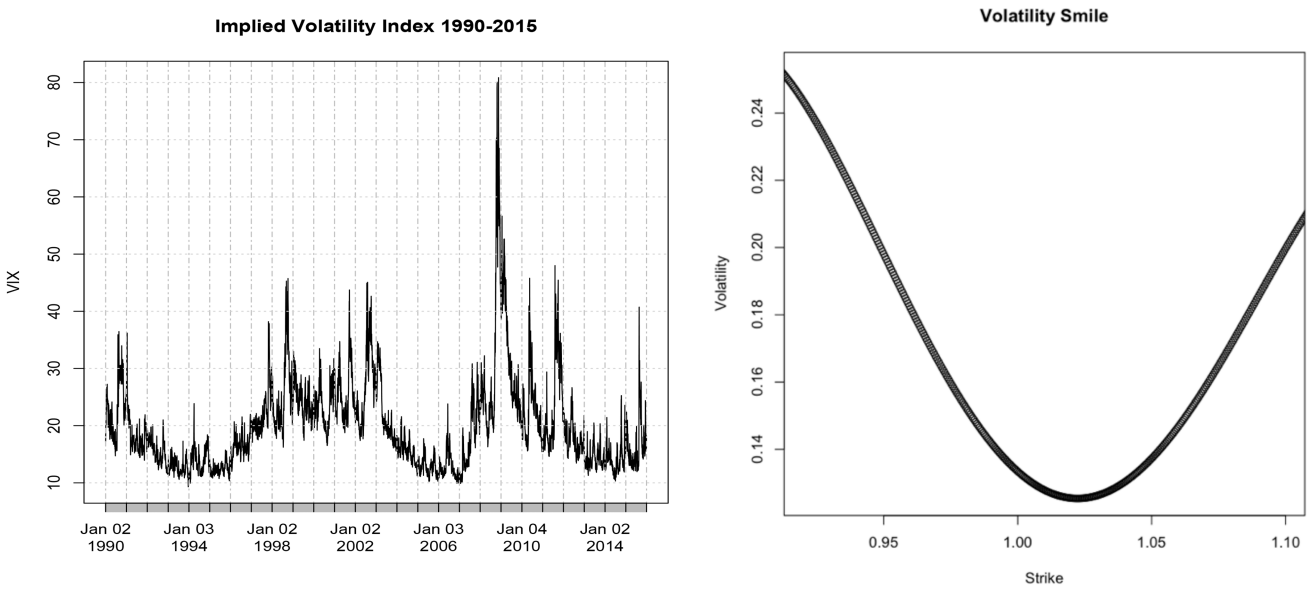


Fig. 4. Implied Volatility Index and the Implied Volatility Smile

options is limited by the premium paid, shorting options could be very risky because the loss is only limited by the market actions. For example, shorting a call option while the market continues moving up could result in a severe loss.

3.2. Options Pricing Model

To derive the fair option price, a common practice is to assume that the underlying asset follows a certain dynamic model such as geometric Brownian motion (GBM) with constant drift and volatility, described by the following stochastic differential equation:

$$dS = \mu S\,dt + \sigma S\,dW \tag{7}$$

where S is the asset price, μ is the drift parameter, σ is the volatility, and W is a Wiener process or Brownian motion. With the assumed model, a closed-form options pricing model has been developed as a function of the current asset price, the option strike price, the time to maturity, and the asset volatility [12, 25]. This popular Back-Scholes-Merton (BSM) option pricing model developed in 1973 had revolutionized the derivative industry for the last several decades.

As mentioned above, an important assumption behind the derivation of the BSM pricing model is that the price of the underlying asset follows a GBM model with constant drift and volatility. However, since the stock market crash of October 1987, the volatility of stock index options implied by the market prices has been observed to be "skewed" in the sense that the volatility became a function of strike and expiration instead of remaining a constant. This phenomenon, referred to as the "volatility smile," has since spread to other markets [26]. Because the original BSM model can no longer account for the smile, investors have to use more complex models to value and hedge their options. In this paper, for the purpose of evaluating the trading performance, we will emulate the option prices subject to the smile phenomenon by utilizing the historical implied volatility index (VIX) data and approximate the volatility smile with a quadratic function of moneyness[3] [25]. Figure 4 shows the historical VIX data from 1990 to 2015 and the corresponding volatility smile based on normalized strike price used in the simulated trading.

3.3. Trading Process

We employ a simple trading strategy, called "strangle," by simultaneously writing both out-of-money[4] (OTM) weekly put and call options. With an expiration cycle of 4 weeks, the options are written repeatedly on a weekly basis. The trading equity is allocated over the 4 weeks period where at the end of each week, a portion of the options expires and a new set of options is initiated/written. We use historical end-of-the-day S&P settlement prices and the options pricing model (Section 3.2) to emulate the filled-prices of the transactions. We assume no transaction cost and no slippage. Note that since the strategy does not produce substantial amount of trading as will be clearly described in the next Section, this assumption does not affect the validity of results.

4. TEST AND SIMULATION

In order to estimate the optimal Kelly's fraction, we first apply the "strangle" strategy with various parameters such as different out of the money (OTM) strikes and maturity dates to test the performance. Specifically, for each set of parameters, the winning probability, the average gain of winning trades, and the average loss of

[3]Moneyness is the relative position of the current price of an underlying asset with respect to the strike price of derivative.

[4]The strike of a call option is above the market price or the strike of a put option is below the market price of the underlying asset.

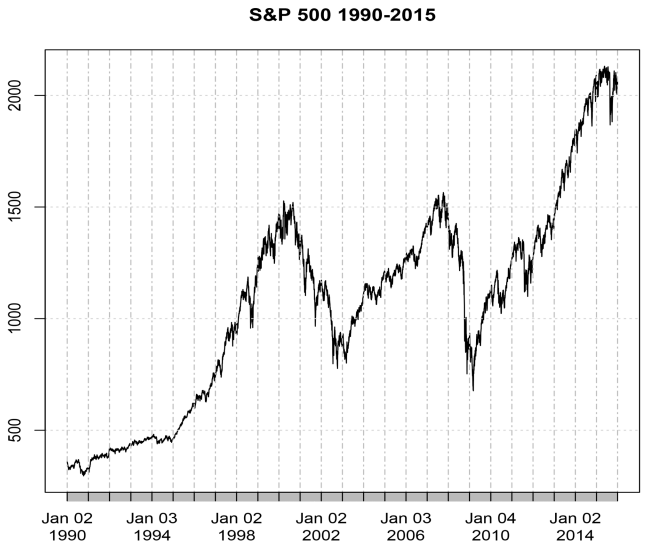


Fig. 5. S&P Index (1990–2015)

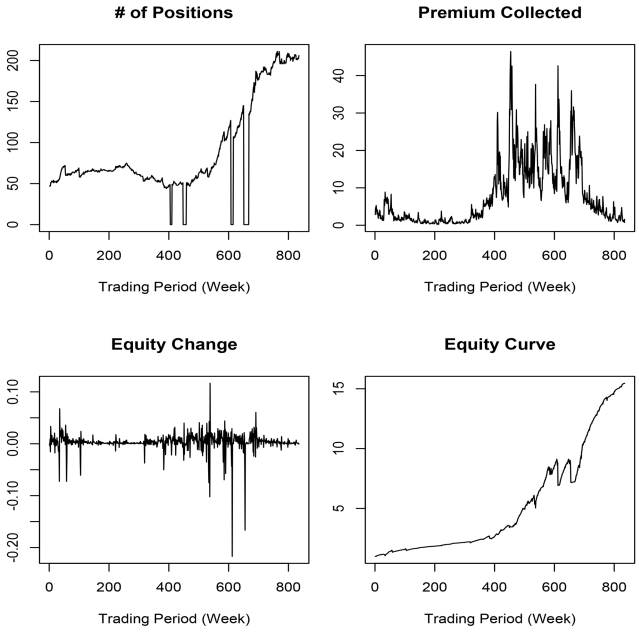


Fig. 6. Options Writing with 6% OTM Strangle (1990–2005)

the losing trades are computed. The results are then used to obtain the optimal Kelly's fraction based on Eqn. (5). The resulting fraction is then applied to allocate equity in the simulated trading process.

4.1. Options Writing Strategy

With the strangle strategy, we simultaneously short the OTM S&P put and call options regardless of the market conditions. We will keep the option positions open until expiration before repeating the same process again in the next trading cycle. To compare the performance, we vary the strike prices of the options from at-the-money (ATM) to 6% OTM with a 1% increment.

Note that the options could expire OTM, and therefore become worthless. In that case, the premium collected by the seller becomes the profit and the positions will be closed automatically by the exchange. On the other hand, if the options expire in the money (ITM), the options will have to be settled in cash in the sense that the sellers have to pay the market price at the expiration time to "buy" back the options they sold. In that case, if the market price deviates more than the premium collected, the seller will incur a loss.

4.2. Simulated Trading

Figure 5 shows the historical S&P data from 1990 to 2015. Since there are only limited historical options prices with specific strikes and expirations available in the public domain, we simulate the options filled-prices based on the model described earlier. Specifically, options prices are obtained by utilizing the BSM model given the S&P price, risk-free interest rate, volatility, and an expiration time of 4 trading weeks after writing the options. The S&P prices are based on historical data and served as the ATM strike prices. Risk-free interest is based on historical 3-month LIBOR data [27] and volatility is based on the historical implied volatility index (VIX).

However, as mentioned earlier, it is well known that true volatility is not a constant but a function of strike and expiration (volatility smile and surface, see Figure 4) [28]. To obtain a more realistic options price, we develop a smile model and adjust the option price accordingly as described in Section 3.2. The results have been validated against the available market data and proved to be reasonably accurate.

4.3. Performance Results

To estimate the Kelly parameters, we use 16 years (1990–2005) of historical data to test the weekly strangle options writing strategy. To be conservative, a 30% margin[5] is assumed to be required for each option contract. In addition, a VIX threshold of 35 is set to avoid a potential catastrophic loss.[6] In other words, all position will be closed when the VIX goes beyond the threshold and new positions will not be written until VIX moves below the threshold. To test the performance, the strike prices are varied from 0% ATM to 6% OTM. For example, Figure 6 shows that with 6% strangle writing, the option strategy produces fairly smooth equity curve with some minor drawdowns. Note that the top left panel of Figure 6 shows that the number of positions drop to zero in several occasions [20]. This is due to the VIX based closing criterion mentioned above.

The detailed resulting performances are summarized in Figure 7. For example, with a 6% OTM strike, the rate of winning is around 80% and the draw down is about 24%. The average amount of winning[7] is 2.2

[5]Margin is the amount of capital needed to initiate and maintain an option position. Typically, for S&P options, the margin requirement could be as low as 10% of the underlying asset value.

[6]The historical average of VIX is below 20.

[7]Note that for e-mini S&P futures market, each point corresponds to $50.

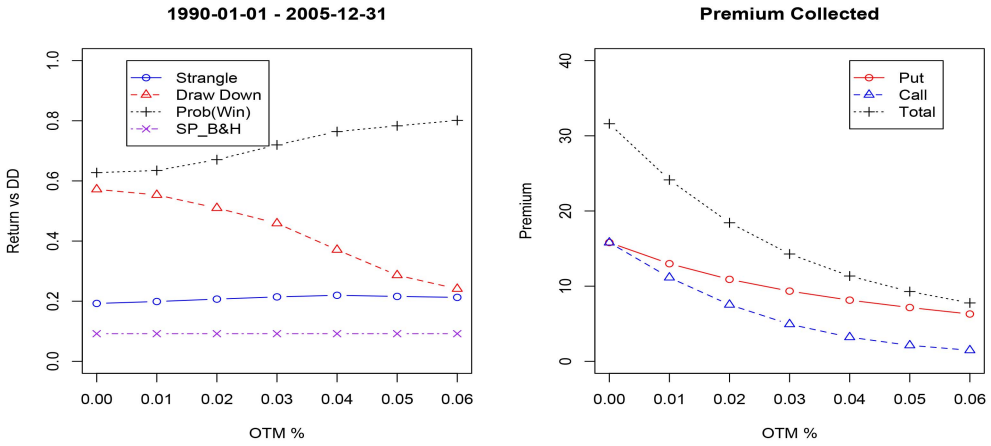


Fig. 7. Performance Summary—Strangle Writing (1990–2005)

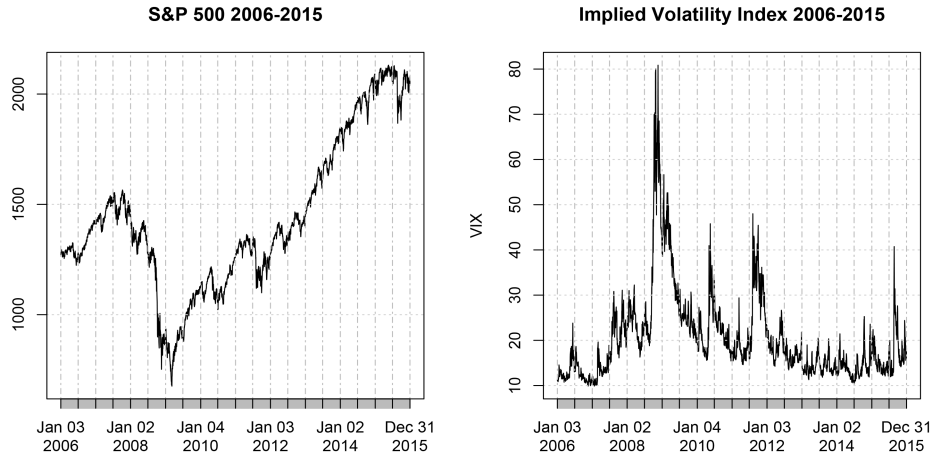


Fig. 8. S&P and VIX—2006–2015

per winning trade and −4.2 for a losing trade. This corresponds to a Kelly's fraction of

$$f^* = \frac{bp-(1-p)}{b} = \frac{\frac{2.2}{4.2}(0.8)-0.2}{2.2/4.2} \approx 0.418. \quad (8)$$

The results indicate that, based on the historical performance, the Kelly's formula recommends an allocation of about 42% of the equity for each trade in order to achieve the highest possible long-term gain. Note that Figure 7 also shows that the average S&P annual return (Buy and hold) during the same period was less than 9%.

We apply the Kelly criterion to test the S&P data from 2006 to 2015. Figure 8 shows the corresponding S&P and VIX data over the 10-year period. Note that a 42% Kelly also implies that at most 42% of the equity can be lost in a single trade. In order to ensure that, a stop loss needs to be in place to determine the total number of option positions that could be written. For example, with an initial capital of $1M, a 42% Kelly and a $4k stop loss per contract, the maximum number of positions is $1M*0.42/$4k = 105.

Figure 9 shows the trading performance with a 6% strangle and the optimal Kelly's fraction. The option strategy produces an average of 40+% annual return and a draw down of around 40%. The detailed performances are shown in Figures 10–11 and also summarized in Table 1. In the table, two sets of performance results, one with the optimal Kelly and the other with a $\frac{1}{2}$ Kelly, are given for comparison. It can be seen from the table that, with the optimal Kelly, the strangle strategy generally produces much higher rate of return than the naïve buy-and-hold (B&H) policy, at the expense of a higher risk (DD) [20].

For example, Table 1 shows that over the 10-year period, the B&H strategy produces an average annual return of 5.73% with a DD of 56.24%, while a 2% OTM strangle produces a 26.39% annual return with a DD of 74.56%. On the other hand, with a 5% OTM strangle, the annual return reach the highest value of 44.82% with a draw down of 50.02%. This relatively "conservative" strategy[8] produces much better return than the naïve B&H strategy while with a smaller drawdown.

[8]A strangle selling strategy with higher OTM strikes is more conservative than the one with lower OTM strikes in the sense that it is expected to achieve a lower rate of return with a smaller DD.

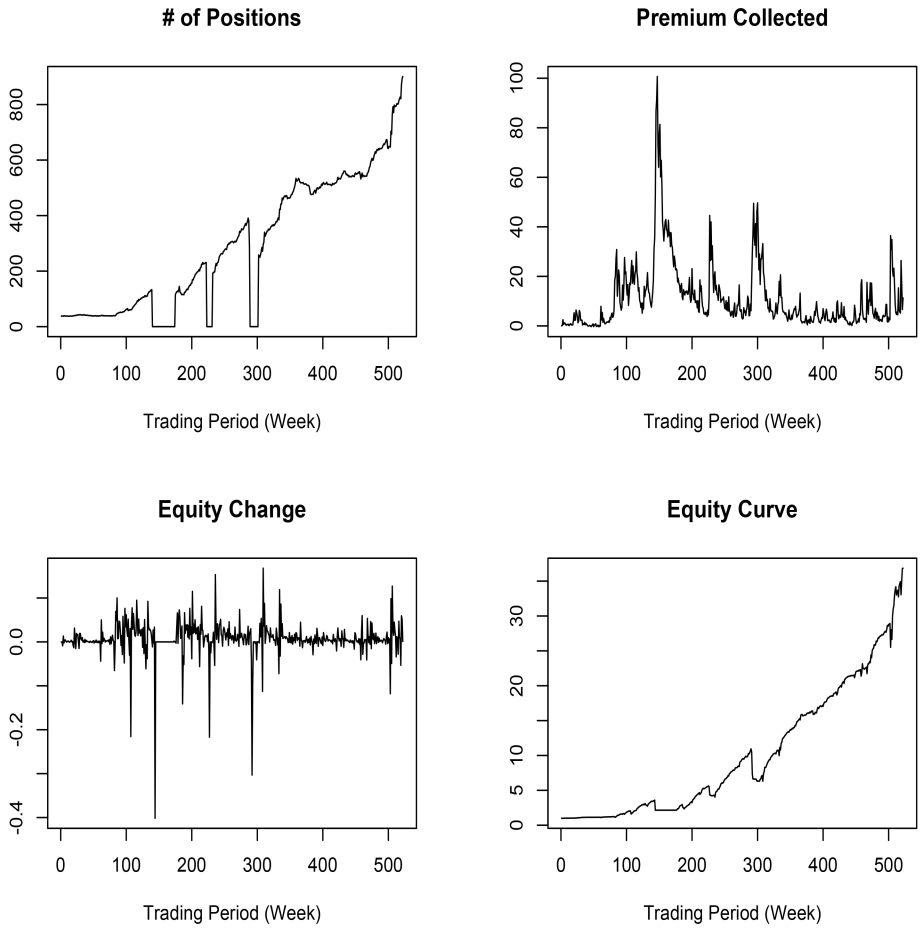


Fig. 9. Options Writing Performance with 6% OTM Strangle and Optimal Kelly Fraction (2006–2015)

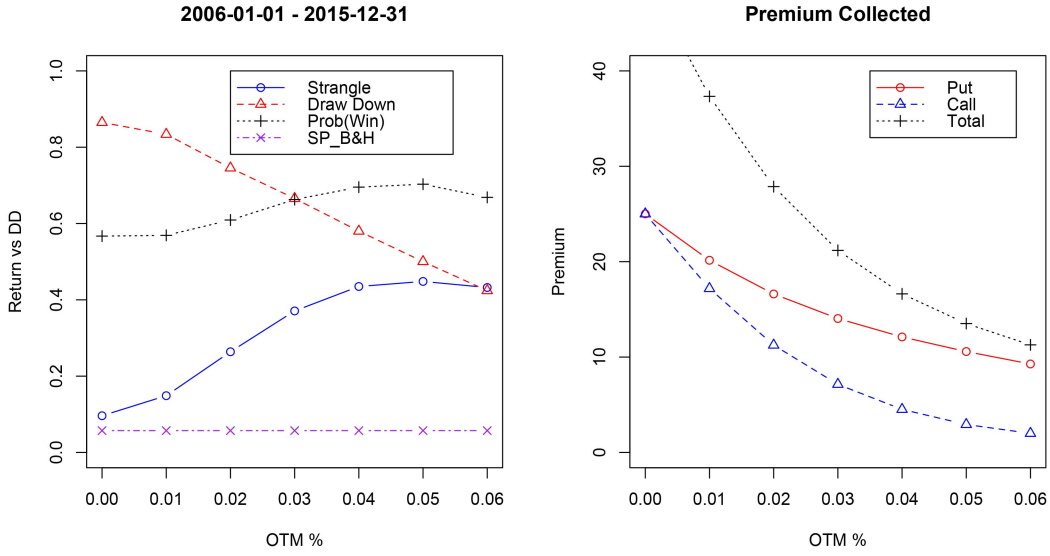


Fig. 10. Performance Summary—Strangle Writing with Optimal Kelly Fraction (2006–2015)

With $\frac{1}{2}$ Kelly, both annual rate of return and maximum drawdown are much lower due to lower leverage as shown in Figure 11 and Table 1. For example, with a 4% strangle, the annual rate of return reduces to 23.28% while the draw down also drops to 30.75%. A 6% strangle reduces the DD to 22.48% and an average annual return of 21.22%. It is clear from the table that by selecting different leveraged options writing policies, the performance can be adapted to fit individual investor's risk aptitude.

Figure 12 shows the trade-off between risk and return with different OTM strangles and different invest-

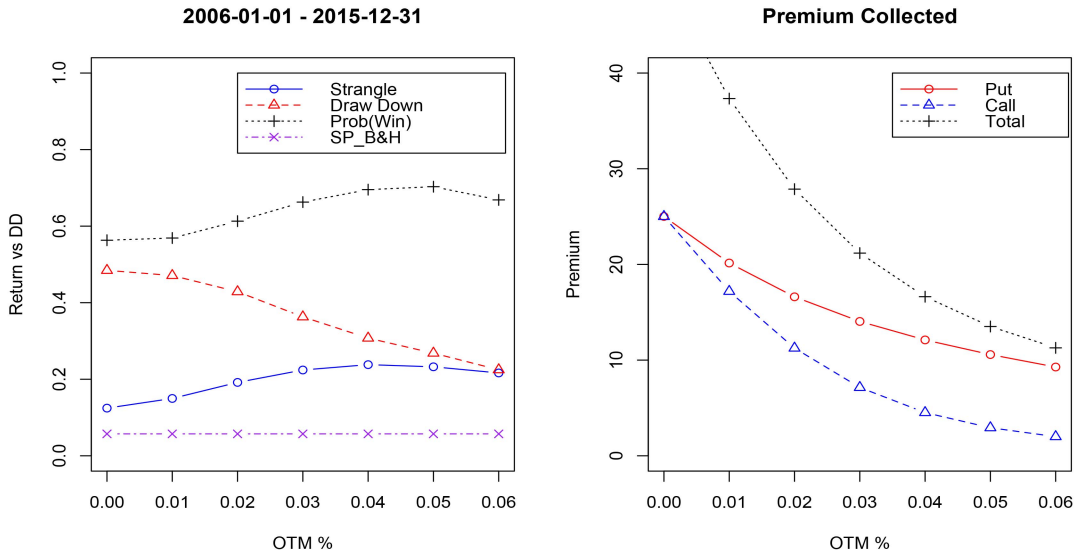


Fig. 11. Performance Summary—Strangle Writing with $\frac{1}{2}$ Kelly fraction (2006–2015)

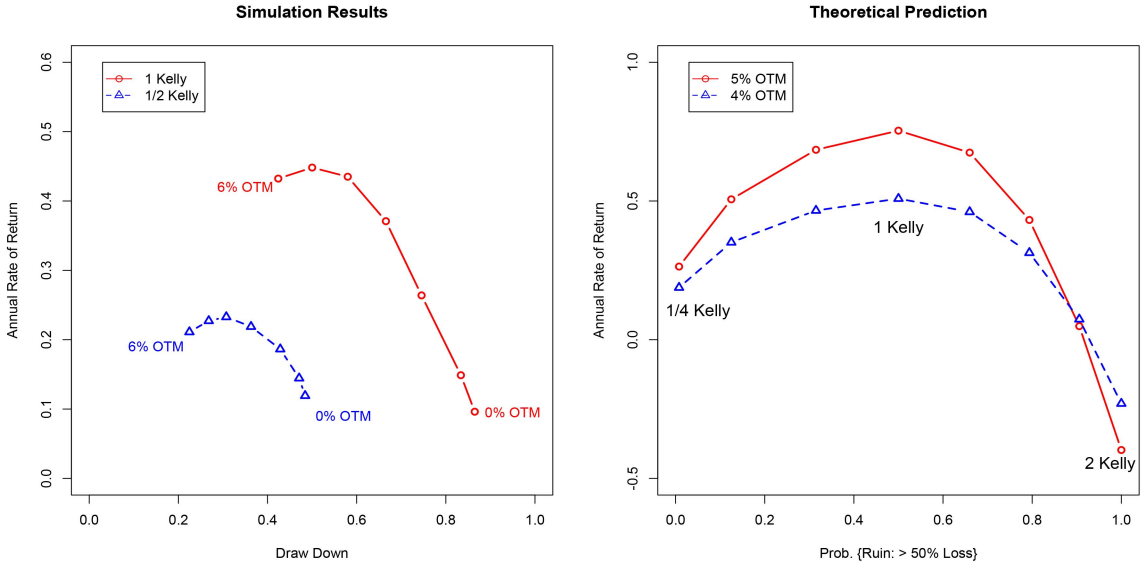


Fig. 12. Performance Summary vs. Theoretical Predictions

TABLE 1.
Performance Comparison with optimal Kelly fraction

2006–2015	Annual RR 1 Kelly	Max DD 1 Kelly	Annual RR $\frac{1}{2}$ Kelly	Max DD $\frac{1}{2}$ Kelly
Buy–Hold	5.73%	56.24%	5.73%	56.24%
Strangle 0%	9.62%	86.49%	11.92%	48.44%
Strangle 1%	14.88%	83.38%	14.45%	47.10%
Strangle 2%	26.39%	74.56%	18.64%	42.86%
Strangle 3%	37.09%	66.56%	21.88%	36.30%
Strangle 4%	43.50%	58.00%	23.28%	30.75%
Strangle 5%	44.82%	50.02%	22.72%	26.82%
Strangle 6%	43.23%	42.41%	21.11%	22.48%

ment fractions [20]. The left panel of Figure 12 shows that, for the simulated trading during the 10-year period, the highest rate of return could be achieved with around 4–5% OTM strangles regardless of the choice of Kelly's fractions. The right panel shows the theoretical prediction using Equations (4) and (6) based on the probability of winning and average gain and loss per trade obtained from the simulation results for 4% and 5% OTM strangle options writing. As expected, the "efficient frontier" peak at the optimal Kelly's fraction. With the SOC curves given in Figure 12, an operating point can be chosen to satisfy almost any desired risk aptitude, should that be defined as drawdown or probability of ruin. For example, an aggressive investor might decide to employ a higher Kelly leverage ratio and a higher OTM strike price with an expectation of better return and an understanding of the accompanying higher risk as indicated by the predictions.

5. MULTI-ASSET ALLOCATION

With the "mean-variance" (MV) approach described in Section 2.1, an efficient frontier can be constructed for multi-asset allocation where for each return level a

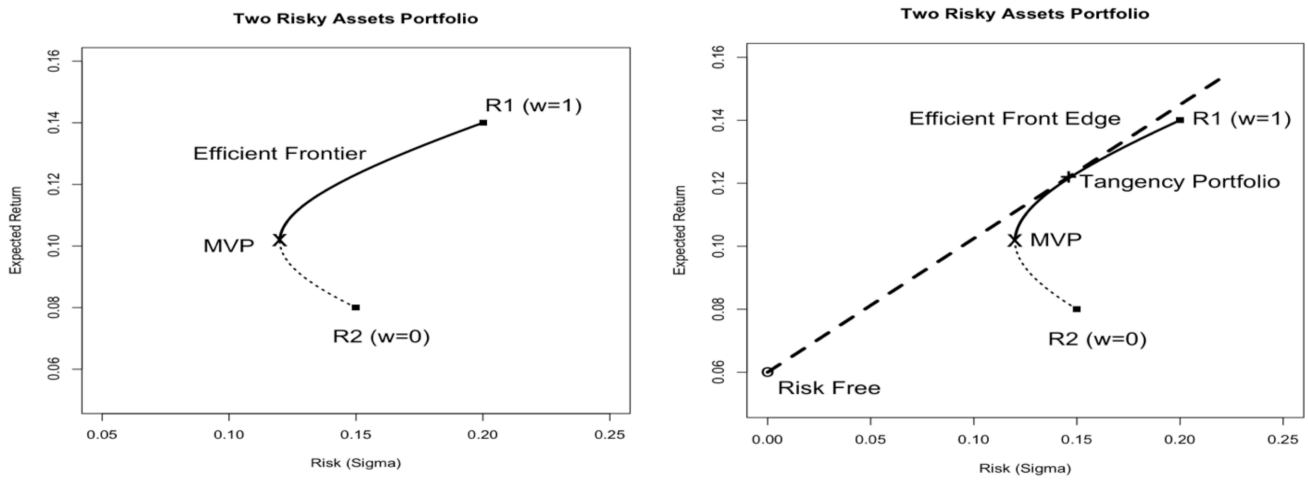


Fig. 13. (a) Efficient Frontier for a Two Risky Asset Portfolio; (b) Efficient Front Edge and Tangency Portfolio

portfolio can be derived with minimum risk (variance). For example, with two risky assets, $\boldsymbol{\mu} = [0.14, 0.08]^T$ and $\Sigma = \text{diag}[0.2^2, 0.15^2]$, Figure 13 (a) shows the "efficient frontier" portfolios on a mean-STD (standard deviation) chart. Each optimal portfolio consists of a combination of the two assets where at the top right end of the frontier, the entire 100% of the capital is allocated to asset 1 while at the bottom, all capital is allocated to asset 2. Any portfolio below the return level of MVP (minimum variance portfolio) is not considered efficient due to its lower expected return. Therefore, an efficient frontier is constructed from MVP to the top right of the curve.

With the efficient frontier, an investor could choose a portfolio on the curve depending on his/her own risk aptitude. For example, a conservative investor may choose a portfolio close to MVP while an aggressive investor may choose a portfolio close to R_1.[9] Note that when risk-free asset[10] is available, an "efficient front edge" can be constructed by connecting the risk-free asset and the tangency portfolio on the mean-variance chart[11] (see Figure 13(b) with risk-free rate $r_f = 0.06$). The MV tangency portfolio (MVTP) can be shown to maximize the risk-adjusted return (Sharpe ratio[12]) and is a desirable choice of optimal portfolio on the frontier[13] [9].

5.1. Kelly's Approach

As in (2), we consider a portfolio consisting of a set of correlated assets with weights $\boldsymbol{\omega} = [\omega_1 \cdots \omega_N]^T$ such

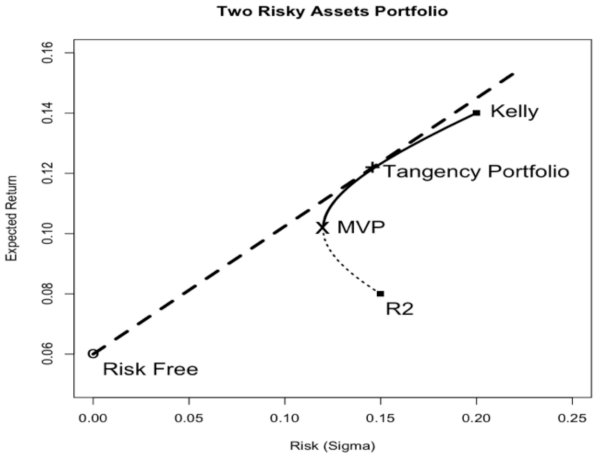


Fig. 14. Tangency Portfolio and the Kelly's Portfolio

that $\sum_{i=1}^{N} \omega_i = \mathbf{1}^T \boldsymbol{\omega} - 1$. Then the portfolio value for the following investment period becomes,

$$P(k) = P(k-1)\left(1 + \sum_{i=1}^{N} \omega_i R_i(k)\right) = P(k-1)(1 + R_P(k)) \tag{9}$$

where $P(k)$ is the portfolio value at time k and $R_P(k)$ is the portfolio return.

According to Kelly, in order to maximize the investment growth rate in the long run, it is equivalent to maximize the logarithm of the equity after each time step [13, 29–30]. Therefore, to construct Kelly's portfolio, with no short selling and no leverage, it is necessary to solve the following optimization problem,

$$\text{Max}_{\omega} E\left[\ln\left(1 + \sum_{i=1}^{N} \omega_i R_i(k)\right)\right] \text{ subject to } \boldsymbol{\omega}^T \mathbf{1} = 1; \omega_i \geq 0 \tag{10}$$

For example, the Kelly's portfolio (KP) for the two risky assets example given in Figure 13 turns out to be the one with 100% allocation on asset #1 as shown in Figure 14.

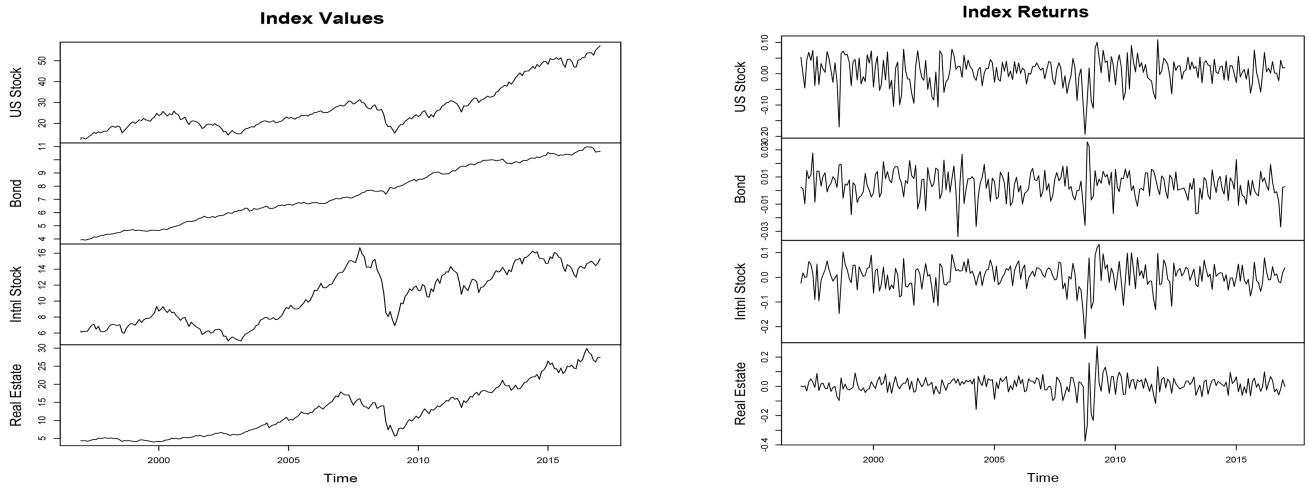
[9] If short-selling or borrowing/leverage are allowed, an investor could choose a portfolio beyond R_1 where higher expected return together with higher risk can be achieved.

[10] Cash or Treasuries with interest rate r_f and with little or no risk.

[11] It's also called the capital market line (CML).

[12] Sharpe ratio is the risk-adjusted return defined as $(\mu_P - r_f)/\sigma_P$

[13] When the portfolio includes all assets in the market, the tangency portfolio converges to the market portfolio by the equilibrium argument [11].



(a) (b)

Fig. 15. (a) The Four Core Asset Index Funds: 1996–2016; (b) Core Asset Returns: 1996–2016

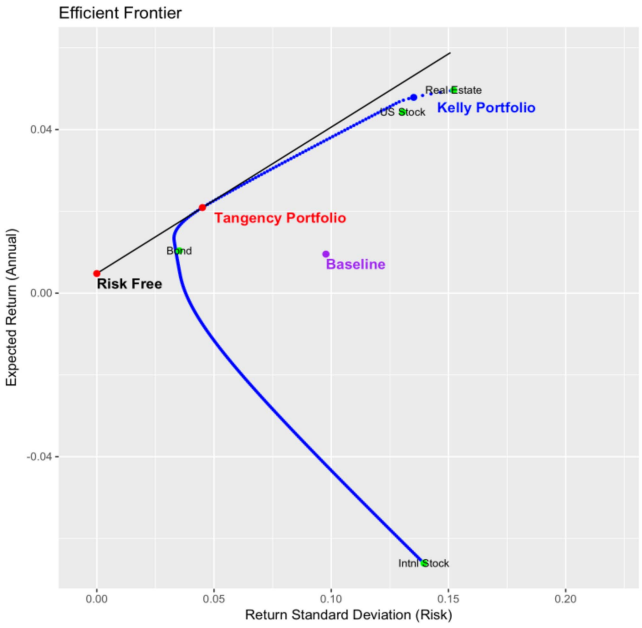


Fig. 16. Efficient Frontier and the Kelly's Portfolio

5.2. Test and Simulation

For evaluation purpose, we compose an artificial portfolio with a few "core assets" selected from the Vanguard index-based mutual fund family. Specifically, four core asset classes, including the US total stock market index (VTSMX), total bond index (VBMFX), international stock index (VGTSX), and the real estate index (VGSIX), are selected to construct the portfolio. The 20-year historical prices obtained from finance.yahoo.com and their log returns of the four assets from 1996 to 2016 are shown in Figure 15.

As mentioned earlier, the goal of constructing an optimal portfolio is to maximize the investor's return or minimizing the risk. Under a given capital constraint, portfolios are constructed and dynamically rebalanced by allocating the capital over the four core assets using different strategies. With no shorting and no leverage assumptions, the allocation fraction of each asset is subject to $\boldsymbol{\omega}^T\mathbf{1} = 1$ and $\omega_i \geq 0$.

The three strategies to be compared include MV tangency portfolio (MVTP), Kelly's portfolio (KP), and the portfolio based on a Naïve strategy. The Naïve strategy is a simple allocation scheme served as the baseline for comparison, in which the portfolio is simply rebalanced uniformly among all the core assets at the beginning of each investment period. The historical data of the four core assets are used to train the model. Specifically, a sliding window of 18 months of data is used to estimate the asset returns, volatilities, and the correlations between the assets. Based on the results, optimal portfolios under each strategy will be formed and rebalanced accordingly on a monthly basis from 2000 to 2016. During the test period (2000–2016), a total of 204 months is available for portfolio rebalancing and performance evaluation. For example, Figure 16 shows a snap shot of the efficiency frontier and the corresponding locations of the three strategies for Nov. 2016. The history of the dynamic allocation fractions of the four core assets in the portfolio based on MVTP and KP are shown in Figure 17. As can be seen, KP tends to take a more extreme allocation than that of the MVTP.

5.3. Performance Results

With the three strategies, the portfolio is dynamically rebalanced monthly during the investment period from 2000 to 2016. In the process, we assume no transaction cost and no slippage for the rebalancing trades. The resulting portfolio equity curves for the three strategies are shown in Figure 18 and the overall performances are summarized in Table 2. The Naïve portfolio has the lowest annualized return (5.33%) with highest volatility (12.80%) and worst drawdown (49.94%) due to its simplicity and the inability to deal with the 2008

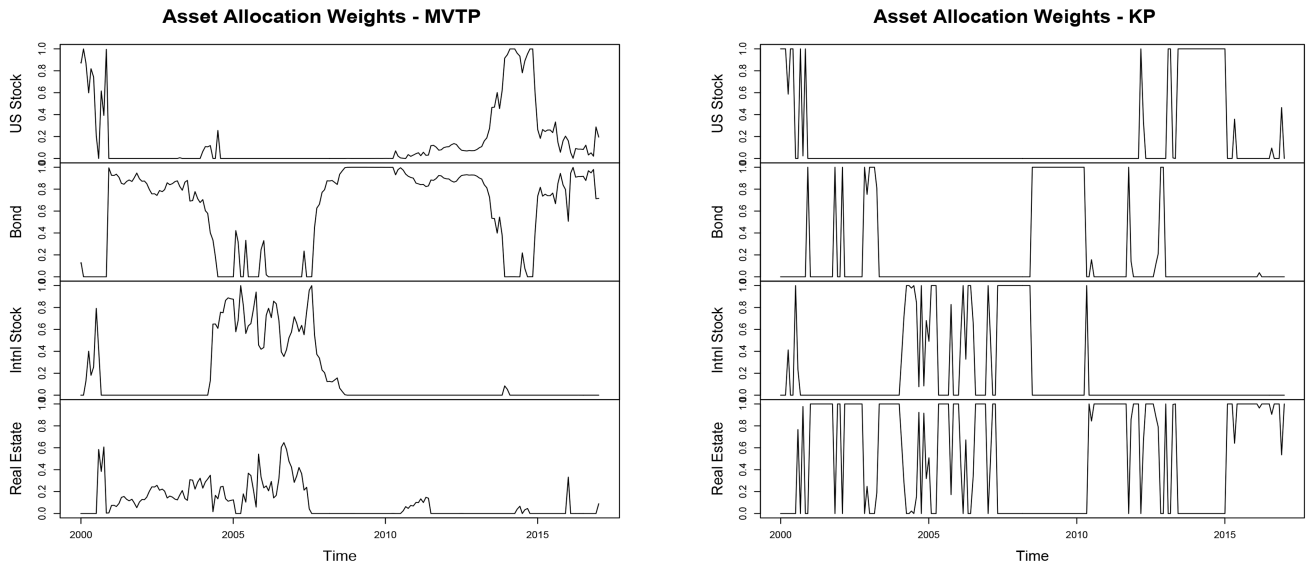


Fig. 17. (a) Asset Allocation Fractions—MVTP; (b) Asset Allocation Fractions—KP

TABLE 2.
Performance Summary: 2000–2016

2000–2016	MVTP	Baseline	KP
Annual Return	7.44%	5.33%	**9.90%**
Annual Risk	**6.65%**	12.80%	12.53%
Sharpe Ratio	**1.044**	0.377	0.750
Drawdown	**14.78%**	49.94%	23.27%

credit crisis. As expected, Kelly's portfolio (KP) is an aggressive strategy and it produces the highest annualized return (9.90%) while suffers a noticeable drawdown (23.27%). Not surprisingly, the MVTP strategy produces the highest Sharpe ratio of 1.044. This is expected by the nature of the tangency portfolio. In addition, it's necessary to point out that the MVTP is able to weather the 2008 credit storm with a fairly small portfolio volatility (6.65%) and a manageable drawdown (14.78%). Note that MVTP and KP represent two complementary strategies that allow an investor to make a tradeoff between risk and return according to his/her own preferences.

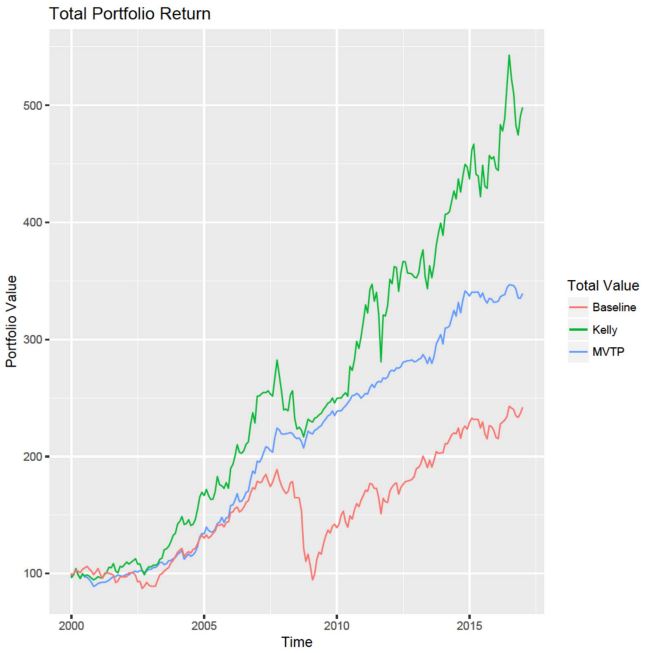


Fig. 18. Equity Curves Comparison

6. SUMMARY AND CONCLUSION

An optimal asset allocation strategy to support investment and trading decisions is developed. First, a simple yet practical trading scenario where the entire equity is allocated between a risk-free asset and a risky asset is considered. The goal is to identify the optimal fraction of the equity to be allocated for trading in order to achieve the highest long-term return with a limited risk. The allocation is based on Kelly criterion derived from the concept of mutual information in binary symmetric communication channels. The resulting model is applied to dynamically allocate equity for writing S&P futures options. Several trading strategies are implemented based on the decision makers' risk aptitude.

Similar to the classical portfolio theory, a system operating curve is developed for each trading strategy where each operating point on the curve representing an expected trade-off between risk of ruin and return. An investor can choose any operating point to satisfy a desired risk and return aptitude. An extended simulation was conducted for performance prediction and evaluation by incorporating historical market data of S&P index futures and options. The results of the simulated trading using these strategies over a 10-year period significantly outperform the buy-and-hold strategy. They

are also consistent with the analytical performance predictions.

The single asset allocation strategy is then extended to portfolio rebalancing with multiple correlated assets. As in the single asset case, the goal is to maximize the long-run portfolio growth rate over many investment cycles. We compare three strategies including MV tangency portfolio (MVTP), Kelly's portfolio (KP), and the Naïve strategy. Through extensive out-of-sample simulations, we show that the resulting KP strategy outperforms the traditional MVTP approach in annual return but with higher volatility. As expected, KP can be considered as an alternative approach for investor to trade off between risk and reward.

While the preliminary results shown in this paper are promising, one of the critical future step is to develop and integrate a dynamic model [31–32] into the allocation strategies so that we would be able to apply the expected future returns of the chosen assets into the optimization process. Another potential future research direction is to integrate the quantitative data with the qualitative information by utilizing the data fusion and machine learning technologies.

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Jiayang Yu for his input and programming assistance on multi-asset allocation.

REFERENCES

[1] S. Martin
Macroeconomic Variables and Stock Market: US Review
MPRA Paper No. 39094, May 2012. http://mpra.ub.uni-muenchen.de/39094/.

[2] X. P. S. Zhang and F. Wang
Signal Processing for Finance, Economics, and Marketing: Concepts, framework, and big data applications
IEEE Signal Processing Magazine, vol. 34, no. 3, pp. 14–35, May 2017.

[3] Y. Zuo and E. Kita
Stock Price Forecast using Bayesian Network
Expert Systems with Applications, Elsvier, Vol. 39, pp. 6729–6737, 2012.

[4] O. Jangmin, J. W. Lee, S. B. Lark, and B. T. Zhang
Stock trading by Modeling Price Trend with Dynamic Bayesian Networks
IDEAL, pp. 794–799, 2004.

[5] J. Geweke and G. Amisano
Comparing and Evaluating Bayesian Predictive Distributions of Asset Returns
European Central Bank Eurosystem, Working paper No. 969, 2008.

[6] N. G. Polson and B. V. Tew
Bayesian Portfolio Selection: An Empirical Analysis of the S&P 500 Index 1970–1996
ASA Journal of Business and Economic Statistics, Vol. 18, No. 2, 2000.

[7] R. Demirer, R. R. Mau, and C. Shenoy
Bayesian Networks: A Decision tool to Improve Portfolio Risk Analysis
Journal of Applied Finance (Winter 2006), 106–119.

[8] C. Shenoy and P. P. Shenoy
Bayesian Network Models of Portfolio Risk and Return
Computational Finance 1999, pp. 87–106, The MIT Press, Cambridge, MA.

[9] J. Hull
Options, Futures, and other Derivatives, 9th edition
Pearson, 2015.

[10] I. Karatzas and S. Shreve
Methods of Mathematical Finance
Springer-Verlag New York, Incorporated, 1998.

[11] C. Cao and J. Hiang
Determinants of S&P Index Option Returns
Review Derivative Research, 10:1–38, 2007.

[12] F. Black and M. Scholes
The pricing of options and corporate liabilities
Journal of Political Economy 81, 637–653, 1973.

[13] J. Kelly
A New Interpretation of Information Rate
Bell System Technical Journal 35: 917–926, 1956.

[14] E. Thorp
Fortune's Formula: The Game of Blackjack
American Mathematical Society, Jan, 1961.

[15] E. Thorp
The Kelly Criterion in Blackjack, Sports Betting, and the Stock Market
http://www.eecs.harvard.edu/cs286r/courses/fall12/papers/Thorpe_KellyCriterion2007.pdf.

[16] E. Thorp
A Man for All Markets: From Las Vegas to Wall Street, How I beat the Dealer and the Market
Random House, 2017.

[17] E. Platen and. Heath
A Benchmark Approach to Quantitative Finance
Springer, 2006.

[18] T. M. Cover and J. A. Thomas
Elements of Information Theory, 2nd edition
Wiley, 2006.

[19] Y. Feng
Dynamic Asset Allocation; A Bayesian Approach
Ph.D. Dissertation, City University of New York, 2014.

[20] K. C. Chang and Z. Tian
Optimal Asset Allocation with Mutual Information
Proc. 19th International Conference on Information Fusion, July 5–8, 2016, Heidelberg, Germany.

[21] H. Markowitz
Portfolio Selection
The Journal of Finance, Vol. 7, p. 77–91, 1952.

[22] E. J. Elton, M. J. Gruber, S. J. Brown, and W. N. Goetzmann
Modern Portfolio Theory and investment Analysis, 9th edition
Wiley, 2014.

[23] CME
http://www.cmegroup.com/trading/equity-index/us-index/sandp-500_contract_specifications.html.

[24] Globex
http://www.cmegroup.com/trading/equity-index/us-index/e-mini-sandp500.html.

[25] R. C. Merton
Theory of Rational Option Pricing
Bell Journal of Economics and Management Science (The RAND Corporation) 4 (1): 141–183, 1973.

[26] D. Backus, S. Foresi, and L. Wu
Accounting for Bias in Black-Scholes
http://faculty.baruch.cuny.edu/lwu/papers/bias.pdf.

[27] Federal Reserve Economic Data, St. Louis FRED
http://research.stlouisfed.org/fred2/.

[28] J. D. Coval and T. Shumway
Expected Option Returns
The Journal of Finance, Vol. LVI, No. 3, 2001.

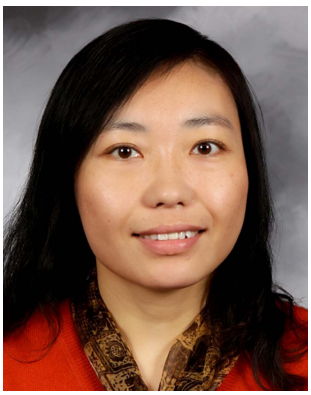
[29] P. Laureti, M. Medo, and Y. Zhang
Analysis of Kelly-optimal Portfolios
Quantitative Finance, Vol. 10, No. 7, p. 689–697, Aug., 2010.

[30] V. Nekrasov
Kelly criterion for multivariate portfolio: a model-free approach
Elsevier, SSRN, 2014. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2259133.

[31] K. Triantafyllopoulos
Time-varying vector autoregressive models with stochastic volatility
Journal of Applied Statistics, Vol. 38, No. 2, p, 369–382, Feb., 2011.

[32] M. West and J. Harrison
Bayesian Forecasting and Dynamic Modes
Springer, 2nd edition, 1997.

Kuo-Chu Chang received the M.S. and Ph.D. degrees in Electrical Engineering from the University of Connecticut in 1983 and 1986 respectively. From 1983 to 1992, he was a senior research scientist in Advanced Decision Systems (ADS) division, Booz-Allen & Hamilton, Mountain View, California. In 1992, he joined the Systems Engineering and Operations Research department, George Mason University where he is currently a professor. His research interests include estimation theory, multisensor data fusion, Bayesian inference, and financial engineering. He is particularly interested in applying unconventional techniques in the conventional decision and control systems. He has more than 30 years of industrial and academic experience and published more than two hundred papers in the areas of multitarget tracking, distributed sensor fusion, Bayesian Networks technologies, and financial engineering. He was an associate editor on Tracking/Navigation Systems from 1993 to 1996 and on Large Scale Systems from 1996 to 2006 for *IEEE Transactions on Aerospace and Electronic Systems*. He was also an associate editor of *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, from 2002 to 2007. He was the Technical Program Co-chair of the 2009 International Conference on Information Fusion, Seattle, USA.



Zhi Tian is a Professor in the Electrical and Computer Engineering Department of George Mason University, since 2015. Prior to that, she was on the faculty of Michigan Technological University from 2000 to 2014. Her research interests lie in estimation and detection theory, with applications in statistical signal processing, wireless communications, and target tracking and inference. She is an IEEE Fellow. She serves as an IEEE Distinguished Lecturer for both the IEEE Communications Society and the IEEE Vehicular Technology Society. She served as Associate Editor for *IEEE Transactions on Wireless Communications* and *IEEE Transactions on Signal Processing*.

Semi-Blind Secure Watermarking based on integration of AES and ECC in DCT Domain

VINEET MEHAN

Copyright protection and integrity of digital images have become one of the vital issues in crucial watermark applications like Cheque Truncation System (CTS), Patient record management system and e-document verification etc. This paper illustrates an integrated watermarking and encryption technique to safeguard copyright of images and to offer security to the watermarked image contents. Watermarking technique based on combination of Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) in Discrete Cosine Transform (DCT) is proposed in this work. 25 sets of watermark are classified for embedding owner details with a size variation of 256–3328 bits. Watermark sequence and the secret keys are the prime requisite in the semi-blind approach for the extraction purpose. Peak Signal to Noise Ratio (PSNR), Structural similarity index measure (SSIM), Correlation Coefficient (CC), Net Pixel Change Rate (NPCR) and Entropy are specified in the objective function to identify noise, structural match, association, variation and imperceptibility factors. The experimental results display that the projected watermarking scheme offers better quantitative parameter outcomes in comparison with previous related techniques.

Manuscript received July 5, 2017; revised January 26, 2018; released for publication January 30, 2018.

Refereeing of this contribution was handled by Alexander Toet.

Author's address: Maharaja Agrasen Institute of Technology, Maharaja Agrasen University, Baddi 174103, H.P., India (E-mail: mehanvineet@gmail.com).

1557-6418/18/$17.00

1. INTRODUCTION

Unauthorized distribution and protection of intellectual digital property raised the need of Watermarking techniques. Watermarking has emerged as a prominent practice in the last decade. Digital data subversion has generated a number of concerns around digital authentication, reliability and copyright defence.

Unrestricted and easy transmission of information is also one of its greatest weaknesses, leading to the copying and outright theft of information, particularly images. Increase in use of digital images brings about the necessity for individuals to safeguard their digital assets. Given the motivation to protect intellectual property by ownership definition and security concerns; a watermarking with AES and ECC for digital images has been suggested as a form of secure watermarking scheme for images.

An expert crafts a digital image with due exertions along with a price. When illegitimate imitation of the image is found on the web, then the proprietorship correlated with the image is to be determined. Due to this delinquent, a practice called watermarking was announced to defend the copyright of digital images with its creative holder. The system of implanting data into digital image is labelled as digital watermarking [1]. Data to be injected into the image is called a watermark. Inserted watermark can be mined in future for the tenacity of proof of identity and verification [2]. Amendment triggered by entrenching the watermark is controlled to preserve visual resemblance amongst the host and the watermarked image [3]. The watermarking scheme can be represented symbolically by

$$I_w = E(I_o, W) \tag{1}$$

(1) where I_o, W and I_w denote the original image, the watermark containing the owner information, and the watermarked image, respectively. For watermark recognition, a perceiving function P is used. This operation is represented by

$$W' = P(I_w, I_0) \tag{2}$$

The extracted watermark sequence W' is then compared with the original W using a correlation measure θ given as

$$\theta(W, W') = \begin{cases} 1, & \text{if } t > \gamma \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where t is the value of the correlation and y' is a positive threshold. One bit watermarking is aimed to identify the existence or the lack of the watermark in the discernable object. Multiple bit watermarking includes a message (M) with n-bit long stream

$$M = \{0,1\}^n \tag{4}$$

such that ($m = m_1, m_2, \ldots m_n$, with $n = |m|$)

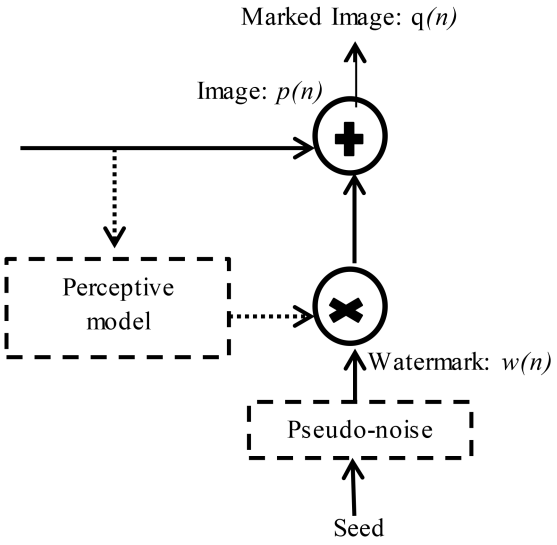


Fig. 1. Perceptible Watermarking Model

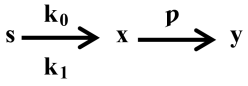


Fig. 2. Robust watermark detection

TABLE 1.
Comparative Analysis of DFT, DWT and DCT.

S. No.	Parameter	DFT	DWT	DCT
1.	Computational Complexity	High	High	Low
2.	Coefficients	Real and Imaginary	Real and Imaginary	Real
3.	Energy Compaction Property	Low	Moderate	High
4.	Block Artifacts	More	Less	Less
5.	Periodicity	More Discontinuous	Discontinuous	Less Discontinuous

Watermarking is categorized into subsequent two classes as per visual insight [4]: Perceptible watermarking and Imperceptible watermarking. Perceptible watermarks work is of assertion of proprietorship and origin [5]. Figure 1 shows the perceptible watermarking model. Observable watermarks decrease the value of an image for an offender devoid of dropping its value for genuine legal tenacities [6]. Imperceptible watermarks are also termed as invisible watermarks [7], as in this the watermarks are not apparent on the image. Watermarks are implanted in the digital image such that visible modification amongst the cover and watermarked image is not perceived [8].

Imperceptible watermarking is categorized as: Fragile watermarking and Robust watermarking. Fragile watermarking is castoff for image certification [9] to attest that acknowledged image was not altered in the course of communication. Even a minor alteration of the image, eliminates the implanted watermark. Fragile watermarking turn into semi-fragile watermarking if a definite boundary is fixed for amendment [10]. Robust watermarking is castoff for safeguarding copyright [11]. In robust watermarking, the inserted evidence is not aloof when the image is altered. Even an enormous extent of alteration does not eradicate the watermark that has been implanted [12].

Figure 2 shows robust watermark detection where s is a vector signal such that $s = (s_1, s_2, \ldots, s_n) \in S^n$ of n-dimensional multimedia host signal; k is an integer from an index set $K = \{1, 2, \ldots, k\}$ where K is total number of messages; x is an authenticated signal such that $x \in S^n$ without hosting perceptible visual distortion; p is a probability density function; and y is the channel output.

Watermarking is classified into two groups [13] depending upon the processing realm : Spatial domain watermarking and Frequency domain watermarking Spatial domain watermarking changes the content of the image pixels unswervingly based on the watermark that has to be implanted [14]. The key benefit of this system is reduced computational complexity and less time [15]. Frequency domain system transforms an image from spatial domain to frequency domain. Watermark is injected into the frequency coefficients. Inverse transform is then smeared to transmute it back into spatial domain. Frequency domain practice is more robust than spatial domain system. Commonly used frequency domain transforms are Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) [16–22]. DCT has been widely used for watermarking applications among all the transforms due to low computational complexity, less block artifacts and high energy compaction property as shown in Table 1.

2. LITERATURE SURVEY

Potdar et al. in 2005 [23] recommended three diverse types of watermarking on the basis of mining prerequisite of the watermark. These include: Non-blind watermarking, semi-blind watermarking and blind watermarking [24]. In non-blind watermarking original image is essential for the abstraction of the watermark. In semi-blind watermarking, only the watermark signal is vital for the removal of the watermark. In blind watermarking no other evidence is required apart from the watermarked image. In our proposed work an efficient semi-blind watermarking scheme is perceived by retrieving the watermarks from the watermarked image. In semi-blind approach only the watermark sequence and the secret keys are needed for the extraction purpose. As in most of the watermark applications original image is not available to the detector, thus the approach exhibits to be more advantageous than the non-blind approach. A semi blind watermarking notation is given as

$$D_d : \bar{I} \times I \times W \square \acute{I} \times \acute{W} \cup \{\perp\} \text{ and}$$
$$X_x : \bar{I} \times I \times \acute{W} \square \acute{M} \times \acute{K} \cup \{\perp\} \quad (5)$$

where D is detection function; X is extraction function and W is a watermark.

A combination of robust and fragile watermarking scheme is designed by Zhang et al. in 2008 [25]. In the robust process watermark is encrypted using AES. DCT is applied to the blue component of the image for embedding watermark. In the fragile process red component of the image is hashed using SHA-256 and then encrypted using ECC key and finally embedded using LSB technique. Our paper used AES with 256 bit key using frequency coefficients rather than in spatial domain. Key generated in our approach is using ECDHP which is immune to attacks and can be used for copyright protection, image integrity certification and identity authentication.

A multipurpose image watermarking with public key cryptography is proposed by Ding et al. in 2008 [26]. A blend of copyright protection is done with content authentication using error correcting codes. In our proposed approach watermarks are implanted into separate DCT coefficients as per image block size. To build up security, the watermarking process makes use of the ECC, ECDHP and AES instead of RSA algorithm as is used in this paper.

3. WATERMARKING APPLICATIONS OF PROPOSED MODEL

Watermarking finds enormous interesting applications in the field of multimedia, image processing and e-commerce etc. Some of the key applications associated with the proposed work are given as:

3.1. Cheque Truncation System

Cheque Truncation System (CTS) is a practice of averting physical crusade of cheque by switching it with a digital image, with an intention for secure and quicker clearance [27]. Watermarking can be applied in the domain of cheque truncation where the cover image is a scanned cheque image. Watermarks to be implanted into the image may encompass user and cheque details. Embedded watermark can be detached later for the purpose of credentials and validation to be exploited for making transactions.

Progression in technology leads to development of novel algorithms and standards by substituting with previous security standards. Standards must take account of aspects like authentication and dependability with the sharing of images in CTS for making transactions. The projected method applies new principles and processes to CTS which tends to be highly consistent and targets at achieving the standardized practice. Watermarking methodology and secure algorithms assistance to offer data reliability, security, and certification solutions to CTS. Watermarking has been proposed as a standard system to solve the anomalies concomitant with CTS. Comparative Analysis of Reserve Bank of India (RBI) based CTS with our proposed approach is shown in Table 2.

TABLE 2.
Comparative Analysis of RBI based CTS [26] with our proposed approach.

Parameter	RBI-CTS	Proposed CTS
Key Generation	DH	ECDHP
Asymmetric Encryption	RSA	ECC
Symmetric Encryption	Triple DES	AES
Image Specification	Gray Scale	Color Image

The proposed effort will corroborate advantageous for the CTS systems being activated in developing countries and will also aid the developed countries to weigh up their prevailing CTS procedures.

3.2. Copyright Protection and Owner Identification of Digital Images

Digital watermarking system allows an individual to add copyright notices and other verification messages to image signals. Such a message is a group of bits describing information pertaining to the owner of the image. The messages can be easily detached by cropping the image part that has the identification. Digital watermarking helps to overcome this problem by embedding the watermark in the form of bits that forms an integral part of the content. In the case of dispute over ownership of the host data, embedded watermark can be used as a proof to identify the true owner of the host data. Image selling portals like imagesbazaar.com carry over one million digital images of Indian visuals. Images at this portal cost substantially depending upon the theme and the style. Proposed technique helps in securing the digital image present online by inserting copyright details.

3.3. Patient Record Management System

Digital watermarking is useful in the e-health environment for tele-consulation and tele-diagnosis purpose [28] Medical images encompass diagnostic information which can be used for timely detection of the diseases. It is useful to safeguard patient data, content certification and medical image reliability. Images are watermarked to prove the integrity by confirming that the image was not altered by illicit person [30]. Watermarking is also applied to determine the authenticity by confirming that the image belongs to the right patient and exact source.

The proposed approach can play an effective role in the management of patient's record. Using this technique vital information related to patient like name, patient id, disease name and patient's photo can be embedded in the medical image. This will prevent the error of mismatching records of patients.

3.4. Certification of Electronic Passport

Certification is a substantial staple for documents, such as electronic passports. Fortification of validity in

passport raised the necessity for the implementation of electronic passport [31]. Electronic Passport is alike to the regular passport with addition of a slight integrated circuit to store digital image [32]. The proposed method permits secure and imperceptible storing of passport details which may include passport number, name of passport owner and other important passport credentials within a digital image. Any variation done to the stored image will result in authentication failure which can be easily identified using the proposed approach.

Usual exercise of programmed passport authorization contests the image existent in the chip with the appearance of the passport holder [33]. The scheme deportment limits when the modifications are not perceived in the image. Prevailing method does not observe the swapping of the passport image with an alternative image. The foremost facet of this verification method is to introduce an orientation between passport's particulars and implanted image insides. Application of digital signature tools legalizes the precision of the evidence retained in the image. It defends passport's genuineness opposing to fraud and security crevices.

The exploration effort proposed by this research work can be used for automatic verification mechanism of passport to be used for immigration clearance system installed at airports. The proposed scheme can also be applied to other important certification documents which include driving licence, identification cards, institute certificates, university degrees and official government documents.

4. SECURE WATERMARKING COMPONENTS

Secure watermarking integrates ECC, ECDHP and AES properties to solve key distribution problem and security concerns for watermarking.

4.1. ECC based Encoding

Elliptic curve cryptography is an asymmetric key cryptosystem which relies on the computational hard discrete logarithm of an elliptic curve [34]. ECC techniques do not perform encryption and decryption of actual data rather they encrypt and decrypt points on the curve. Encoding translates a message into points defined by the elliptic curve, while decoding translates the points back to the original message [35].

ECC operations use multiplication operations instead of exponentiation operations. This makes ECC much faster than other public key cryptosystem like RSA. The security level specified by RSA can be delivered by reduced key size of ECC. For example, the 1024 bit security strength of a RSA can be obtained by only 163 bit security strength of ECC [36]. In the proposed work ECC's small key size, high security and reduced computational complexity characteristics are integrated with digital watermarking for improved ownership protection.

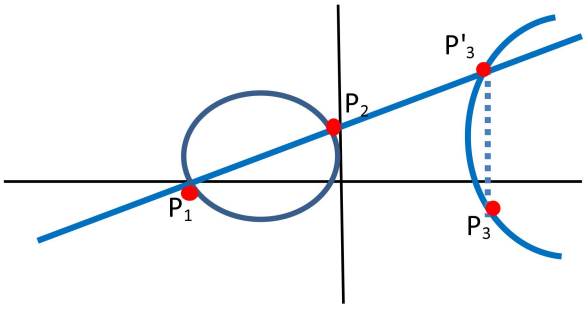


Fig. 3. Adding points such that $P_1 \neq P_2$

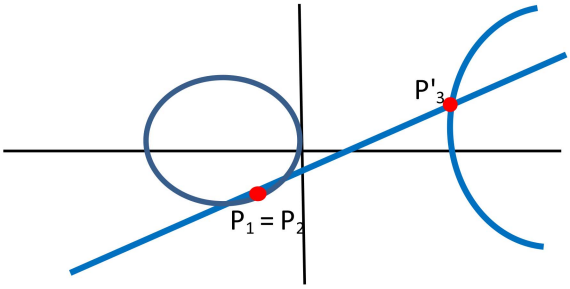


Fig. 4. Adding points such that $P_1 = P_2$

Elliptic curves are customarily signified using Weierstrass [37–39] devising in the most common form. An elliptic curve E_c on a prime field F_p is specified as

$$E_c(F_p) : y^2 = x^3 + ax + b \qquad (p > 3) \qquad (6)$$

where $a, b \in F_p$ and $\Delta = -16(4a^3 + 27b^2) \neq 0$. Different choice of a and b gives different elliptic curves. A true condition of Discriminant (Δ) forms Group Law [40–44]. There can be three cases in this situation.

Case 1: To add two separate points P_1 and P_2 such that $P_1 \neq P_2$. For an equation $y^2 = x^3 - x$ the elliptic curve is shown in Figure 3.

Step 1. Join the two points i.e. P_1 and P_2 on an elliptic curve.

Step 2. The line will also intersect the elliptic curve at P_3'.

Step 3. Reflect the line to get point P_3.

Case 2: To add two points P_1 and P_2 such that $P_1 = P_2$. For the same equation $y^2 = x^3 - x$ the elliptic curve is shown in Figure 4.

Step 1. Find the tangent line to pint P_1 on an elliptic curve.

Step 2. Find the second point of intersection i.e. P_3'

Step 3. Reflect P_3 to get point P_3'.

Case 3: In case of parallel lines it is assumed that the line from P_1 to P_2 will intersect the curve at ∞. In this case the elliptic curve is shown in Figure 5.

In order to find the coordinates of third point using Group Law the line equation (7) is computed with

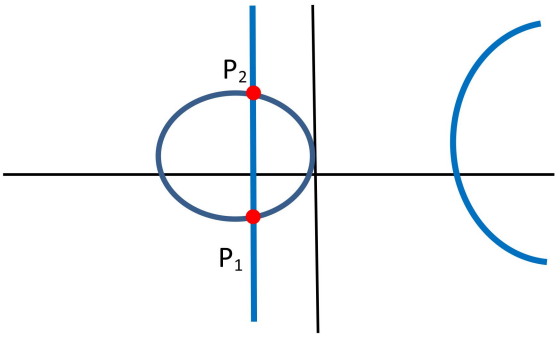


Fig. 5. Adding vertical lines

elliptic curve equation.

$$y = mx + b \tag{7}$$

Let $m = \lambda$ and $b = \beta$ be substituted in equation (7) such that

$$y = (\lambda x + \beta) \tag{8}$$

$$y^2 = (\lambda x + \beta)^2 \tag{9}$$

$$(\lambda x + \beta)^2 = x^3 + ax + b$$

$$\lambda^2 x^2 + \beta^2 + 2\lambda x\beta = x^3 + ax + b$$

$$x^3 - \lambda^2 x^2 - 2\lambda x\beta + ax + b - \beta^2 = 0 \tag{10}$$

For a cubic equation (5) let x_1, x_2 and x_3 are the three roots such that

$$x_1 + x_2 + x_3 = \lambda^2$$

$$x_3 = \lambda^2 - x_1 - x_2 \tag{11}$$

Using point slope form between points (x_1, y_1) and (x_3, y_3)

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

$$y_3 = y_1 + \lambda(x_2 - x_1) \tag{12}$$

Our contribution in this work is to apply logical non-linear ECC curve points to safeguard watermark insertion against active content identification attacks. The proposed algorithm performs selective encoding on the transform coefficients. Encoding converts owner details into points defined by the elliptic curve in order to be suitable for encryption. Decoding converts the points into the original message at the time of retrieval.

4.2. Key generation with ECDHP

Traditional digital rights management (DRM) schemes involve a twofold structure consisting of only owner and the buyer. With the ascendable rise in digital industry multi-level distributors and sub-distributors are needed to support and circulate the digital content [45]. A native distributor can identify the possibly unfamiliar marketplace to the owner and make strategies as per the requirement of the trade. But at the same time, a selfish distributor can pass on the digital content to other consumers without the consent of the owner. ECDHP solves this content packaging mechanism by generating key among all the owners, distributors and sub-distributors. ECDHP is a deviation smeared to Diffie Hellman technique through ECC [46]. The method allows members without any former consociate, to reciprocally generate a shared key above a susceptible network [47]. The content packaging system is handled without the need of a license granting authority [48]. The key is used in encrypting the credentials of all the persons involved in the chain. The owner then passes the watermarked content containing the embedded encrypted credentials to the next level for distribution. Key generation (K_G) by ECDHP prevents the illegal circulation of digital content among multiple owners, distributors and sub-distributors.

NIST based elliptic curves are challenging to solve as the discrete log problem is strong. Key created by the system can be castoff by cryptographic organizations to defend the legitimacy and cover up of the information [49]. Trustworthy heralds can tangibly distribute the secret key, but as the reckoning of key exchange upsurges, the power involved in the distribution of keys grows quickly. Programmed key establishing arrangement based on ECDHP assistances in the conservation of the cryptographic schemes applied in current dominions [50]. The procedure is appropriate to covenant with exclusivity, authorization, key agreement and accelerative concealment.

Key generation using ECDHP involves:

1. A_l and B_b agree publicly on elliptic curve (E_p) over a large finite field.
2. A_l and B_b each privately choose large random integer as secret key A_k and B_k.
3. Using elliptic curve point addition, A_l computes $(A_k G)$ on E_p and sends it to B_b.
4. Similarly, B_b computes $(B_k G)$ on E_p and sends it to A_l.
5. Both A_l and B_b can now compute the point $(A_k B_k G)$.
6. Shared secret key computed by both A_l and B_b is the same.

ECDHP forms efficient arithmetic with shorter key length. ECC provides enhanced security based on discrete logarithm problem. NIST prime curve is computationally efficient as it significantly reduces the total number of multiplies in an exponentiation. To protect a 256 bit symmetric key, RSA algorithm would require 15360 bit key size which is approximately 30 times greater than the size of elliptic curve with 521 bits.

4.3. Secure Watermark Encryption with AES in DCT Domain

AES is a symmetric block cipher algorithm. It uses iterated block cipher, supporting a static length block

TABLE 3.
AES algorithm parameters.

Algorithm	Key Length (words)	Block Size (words)	Number of Rounds
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

TABLE 4.
AES with ECC prime fields.

Symmetric Length	Algorithm	Prime Field	Binary Field
80	SKIPJACK	$\|p\| = 192$	$m = 163$
112	Triple-DES	$\|p\| = 224$	$m = 233$
128	AES Small	$\|p\| = 256$	$m = 283$
192	AES Med.	$\|p\| = 384$	$m = 409$
256	AES Large	$\|p\| = 521$	$m = 571$

of 128 bits [51, 52]. The AES algorithm primarily comprises of three phases: round change, turns and key expand. Each round conversion includes non-linear layer, linear mixture layer and add round key layer. AES algorithm properties are depicted in Table 3. Three key sizes of 128 bits, 192 bits and 256 bits specify different number of repetitions of transformation rounds.

Watermark security is safeguarded by encrypting owner details by means of AES with 256 bits key. Encrypted owner details are generated in multiple of 128 bits as per the block size specification of AES. Encrypted watermark is then implanted into the digital image. AES algorithm delivers watermark security as only a legitimate owner can retrieve and decrypt the inserted stuffing. AES is the preeminent recognized symmetric algorithm for encrypting information, but it suffers from the delinquent of key distribution [53–56]. The key distribution concern is elucidated using ECC by using ECDHP. A hybrid encryption algorithm of AES and ECDHP ensures the content security in digital watermarking. AES provide fast computing speed and encrypts lengthy data while ECDHP handles the key management issues. The projected scheme inhibits the confidentiality of owner data by conjoining encryption with watermarking. Table 4 gives the sizes of the various underlying fields. $\|p\|$ is the length of the binary expansion of the integer p.

Encrypted details are embedded in digital image using the DCT methodology. Only an authenticated user with secret key can retrieve the inserted watermarks from the specified positions within the watermarked image. DCT transforms the image from spatial domain to transform domain [57]. 2-D DCT of an $N \times N$ real signal matrix $f(x,y)$ $(x,y = 0,1,2,\ldots,N-1)$ is defined as

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \times \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{13}$$

$$\alpha(u)\alpha(v) = \begin{cases} \sqrt{\dfrac{1}{N}} & u = 0,\ v = 0 \\ \sqrt{\dfrac{2}{N}} & u \neq 0,\ v \neq 0 \end{cases} \tag{14}$$

where

$C(u,v)$: DCT coefficient at frequency (u,v)

$f(x,y)$: Original image pixel at location (x,y)

$x,y = 0,1,2,\ldots,N-1$

$u,v = 0,1,2,\ldots,N-1$

$\alpha(u)$ and $\alpha(v)$ are the scale factors needed to make DCT orthogonal

2-D inverse DCT of $N \times N$ image matrix is defined by the equation (10) [58] as shown below:

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)\cos\left[\frac{\pi(2x+1)u}{2N}\right] \times \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{15}$$

Matrix and image data define the necessary coefficients for implanting the watermark content. Mid frequency coefficients at $[2,0]$ position are altered within each 4×4 quantised block. Separability [59] and Symmetry [60] characteristics of DCT are exploited to create a 4-point DCT in matrix form. If two 1D elementary functions are same, the transform is said to be symmetric.

$$A_{k,l}(m,n) = a_k(m)b_l(n) = a_k(m)a_l(n) \tag{16}$$

This expression allows a notation in terms of the analysis matrix A associated with the 1D Transform

$$Y = A * XA^H \tag{17}$$

where,

$$X = \begin{pmatrix} x(0,0) & \cdots & x(0,n) & \cdots & x(0,N-1) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x(m,0) & \cdots & x(m,n) & \cdots & x(m,N-1) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x(M-1,0) & \cdots & x(M-1,n) & \cdots & x(M-1,N-1) \end{pmatrix} \tag{18}$$

and

$$Y = \begin{pmatrix} y(0,0) & \cdots & y(0,n) & \cdots & y(0,N-1) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y(m,0) & \cdots & y(m,n) & \cdots & y(m,N-1) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y(M-1,0) & \cdots & y(M-1,n) & \cdots & y(M-1,N-1) \end{pmatrix} \tag{19}$$

TABLE 5.
Owner watermarks embedded in Q1.

Label	W. Content	W. Size (in bits)
W1	OID1	256
W2	W1 + ON1	384
W3	W2 + OHN1	512
W4	W3 + OSE1	640
W5	W4 + OC1	768
W6	W5 + OST1	896
W7	W6 + OMN1	1024
W8	W7 + OEM1	1152
W9	W8 + OID2	1280
W10	W9 + ON2	1408
W11	W10 + OHN2	1536
W12	W11 + OSE2	1664
W13	W12 + OC2	1792
W14	W13 + OST2	1920
W15	W14 + OMN2	2048
W16	W15 + OEM2	2176
W17	W16 + OID3	2304
W18	W17 + ON3	2432
W19	W18 + OHN3	2560
W20	W19 + OSE3	2688
W21	W20 + OC3	2816
W22	W21 + OST3	2944
W23	W22 + OCY3	3072
W24	W23 + OMN3	3200
W25	W24 + OEM3	3328

2-D DCT, configuration and disintegration are separable processes, consequently resizing of images can be consummate by applying 1-D operations successively in horizontal and vertical directions. Our resizing method with 1-D sequence includes a factor of S/T where S and T are relatively large prime numbers greater than 1. A total of U successive N-point DCT blocks are prerequisite. Each DCT block is zero padded to a size of SN and then decomposed into SN-point DCT blocks. Consequently, T disintegrated N-point DCT blocks are collected into a single TN-point DCT block. Each composed TN-point DCT block is then trimmed to a size of N.

Integration of AES with RSA to safeguard the watermark confidentiality consequences in intake of large key size grounded on integer factorization. Computationally proficient key exchange contrivance built on ECC can substitute this security constraint with smaller key size. AES is the finest recognized symmetric key cryptographic algorithm for encrypting information. It guarantees comprehensive safekeeping of the watermark by smearing block cipher methodology with fixed length blocks of 128 bits. For improved safety a concentrated key length of 256 bits is engendered by employing ECDHP. Energy compaction scrutiny of DCT produces the essential transform coefficients. Using this characteristic a reduced fraction of coefficients is attained with big magnitude. Quantizing auxiliary coefficients root for analogous re-construction. Re-watermarking model uses the sequential inclusion approach by providing insertion flexibility to the owner of image.

5. PROPOSED MODEL, EXPERIMENTAL RESULTS AND DISCUSSION

5.1. Experimental Environment

A total of one hundred test images are taken for embedding watermarks. Color images of .jpeg format form the test images. Four different image dimensions are identified for embedding which includes: 512×512, 640×480, 800×600 and 1024×768. Each dimension includes 25 different images. Owner watermarks embedded are are shown in Table 5. Size of watermark varies from 256–3328. Owner details include Owner Identification Number (OID), Owner Name (ON), Owner House Number (OHN), Owner Sector (OSE), Owner City (OC), Owner State (OST), Owner Mobile Number (OMN), Owner E-mail (OEM) and Owner Country (OCY). Multiple owner details are generated by assigning the owner details sequentially. Encrypted owner details are generated in multiple of 128 bits as per the block size specification of AES.

Watermark embedding is achieved by combining different owner details. In re-watermarking multiple watermarks are implanted in a sequential manner. Re-watermarking model uses the successive insertion method to deliver flexibility by defining the number of watermarks to be inserted in the image.

5.2. Proposed Model

For embedding and extracting watermark content (E_c) from cover Image (I), various operations performed are depicted in Figure 6 and Figure 7. Mid Frequency Coefficients (M_{FC}) are quantized using DCT in Block Size (B_s) of 4×4. Partial IDCT and second DCT are applied for block determination (B_D). The RGB color space is converted to YUV color space for each 4×4 block using equations (20).

$$\begin{aligned} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ U &= 0.596 \times R - 0.275 \times G - 0.321 \times B \\ V &= 0.212 \times R - 0.523 \times G - 0.311 \times B \end{aligned} \tag{20}$$

5.3. Experimental Results

Quantitative parameters are analyzed for identifying the effectiveness of the proposed approach. The parameters include Peak Signal to Noise Ratio (PSNR), Structure Similarity Index (SSIM), Correlation Coefficient (CC), Entropy (E), Embedding Processing Time (EPT) and Retrieval Processing Time (RPT). The statistical analysis data SAD containing minimum (MN), maximum (MX) and mean (ME) values are evaluated for each parameter.

Quantitative parameters are mathematically defined image quality measures which play a vital role depending upon the image processing applications they are applied in. The quality measures are independent of the perceptual conditions and specific observers. PSNR is

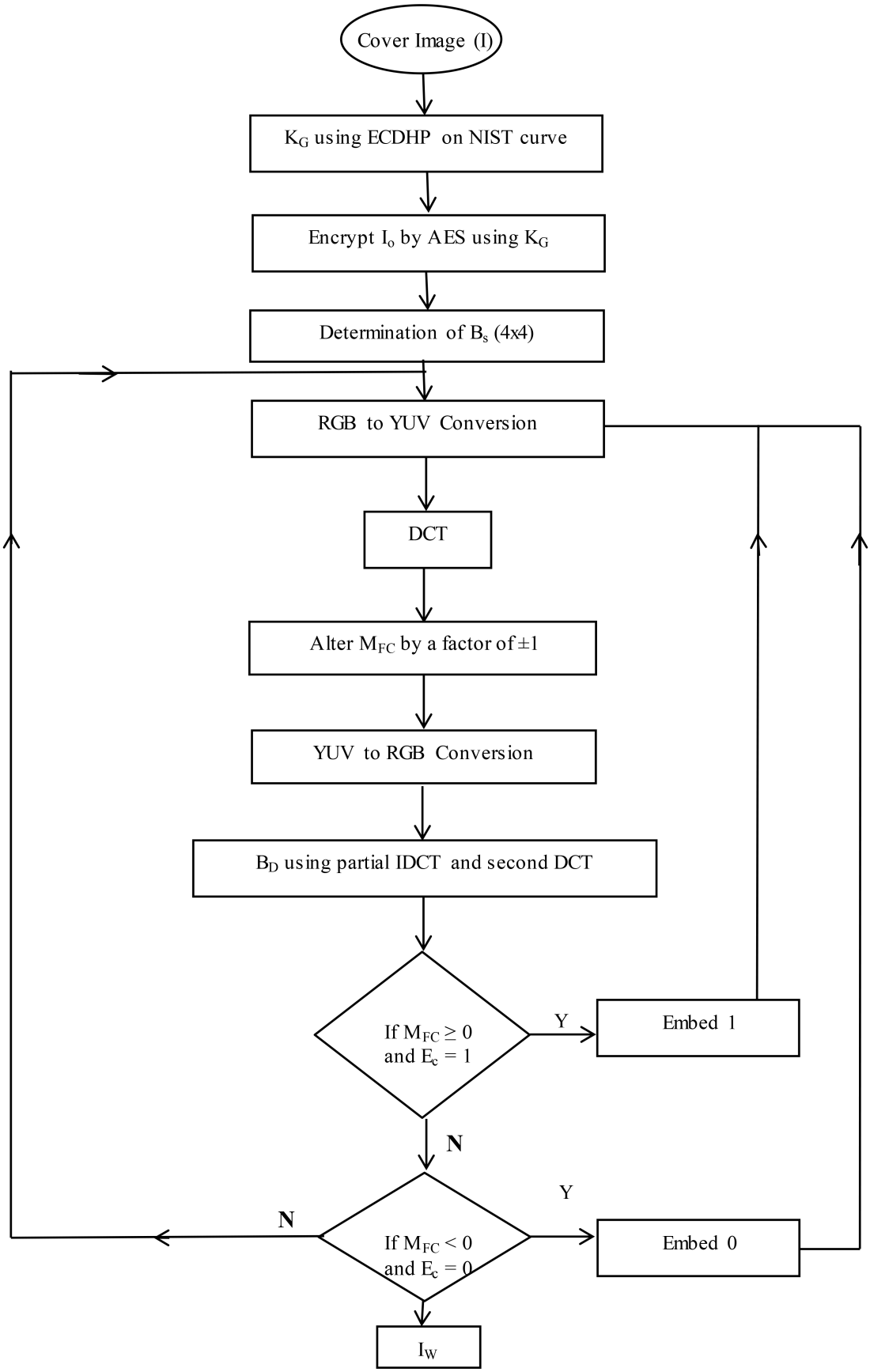


Fig. 6. Watermark Embedding Algorithm

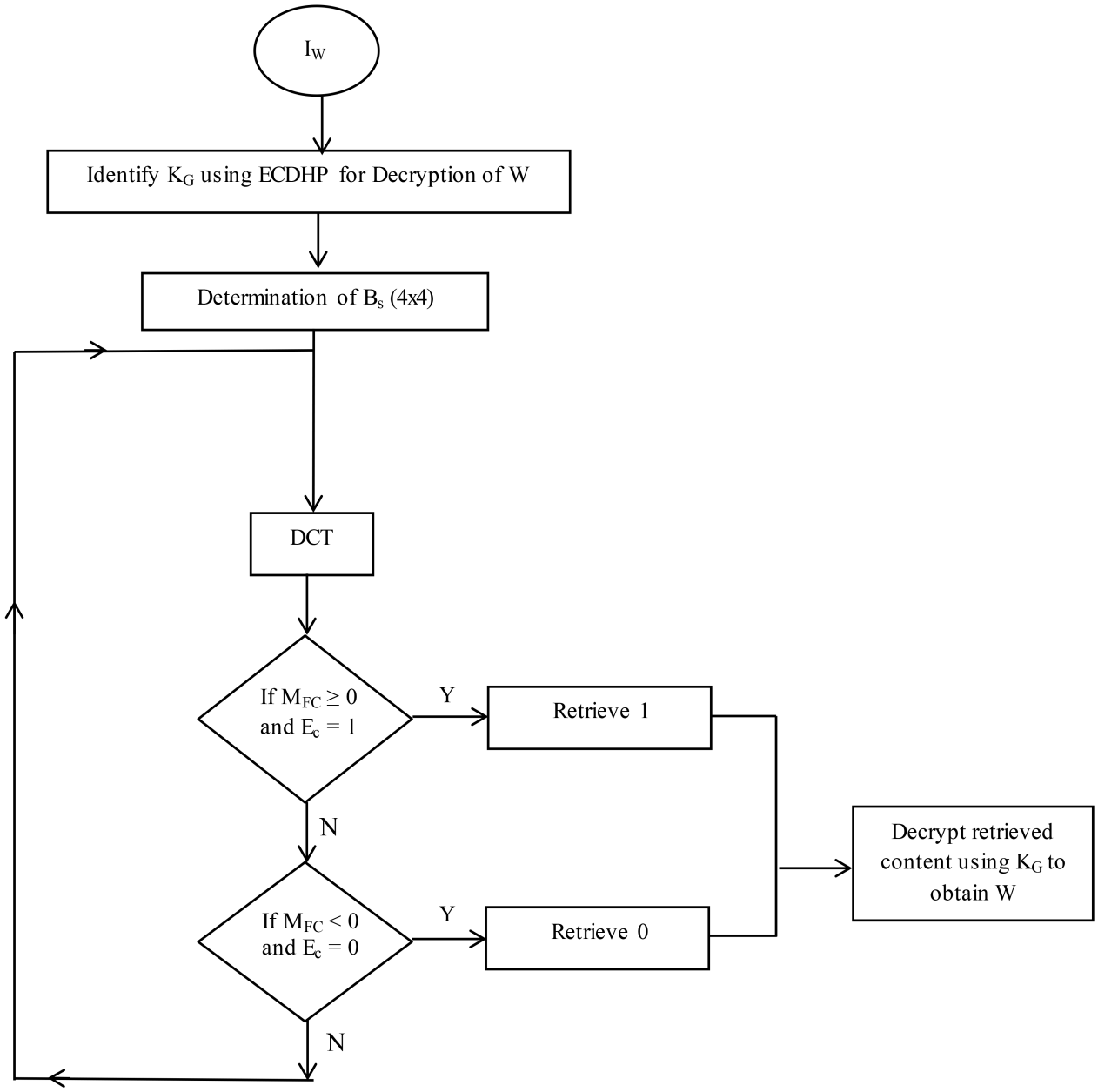


Fig. 7. Watermark Retrieval Algorithm

created on pixel difference based measure. In this original and watermarked images are compared in terms of undistorted reference signal and error signal. SSIM on the other hand is based on Human Visual System measure. This measure is closely related to the perception of human eye in terms of luminance, contrast and comparative structure of two images. In CC correlation of pixels is used as a measure of the image quality measure. Entropy is used to predict the image coding quality for different embedding rates. It measures the disorganized occurrence of watermarked pixels in each row and column and to increase the image visibility.

1) PSNR PSNR is a commonly used measure for determining the quality of images. PSNR computes the peak signal to noise ratio between two images. The ratio factor is used for quality determination among cover image and watermarked image. PSNR for image is calculated in decibels (dB) using the equation [61] (21).

$$\text{PSNR} = 10\log_{10}\frac{(2^N - 1)^2}{\text{MSE}} \tag{21}$$

N is the maximum bit size for a pixel, MSE is Mean Square Error.

PSNR is calculated for all image dimensions with varying watermark size. High values of PSNR obtained imply that the generated image contains less noise. Inverse relation exists between MSE and PSNR such that a lower value of MSE results in high PSNR whereas a higher value of MSE results in low PSNR.

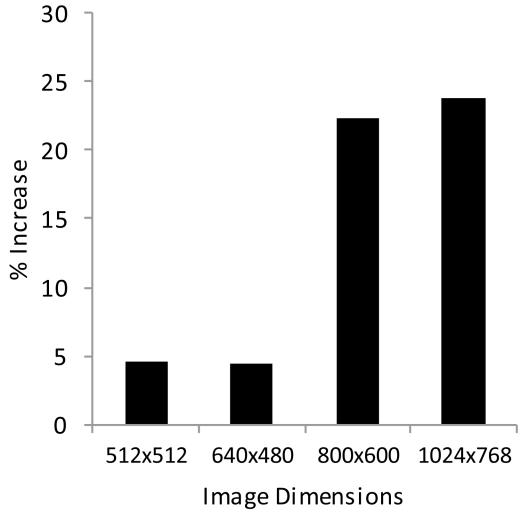


Fig. 8. % Increase in PSNR for different image dimensions.

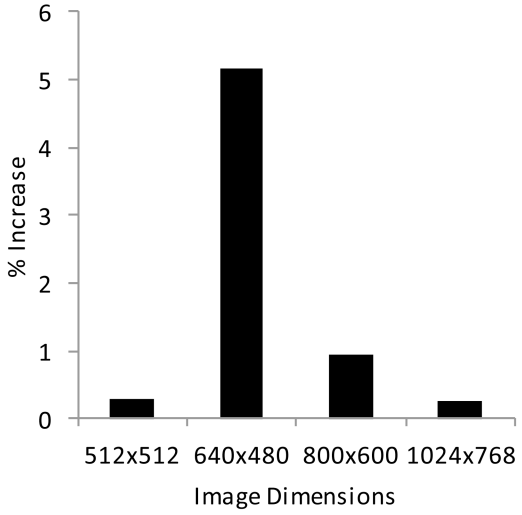


Fig. 9. % Increase in SSIM for different image dimensions.

TABLE 6.
Comparative analysis of PSNR obtained with previous approaches.

Image Dimensions	Proposed Approach	Previous Approach	% Increase
512×512	50.69	48.5 [62]	4.52
640×480	51.28	49.09 [63]	4.46
800×600	53.17	43.48 [64]	22.29
1024×768	55.22	44.6 [65]	23.81

TABLE 7.
Comparative analysis of SSIM obtained with previous approaches.

Image Dimensions	Proposed Approach	Previous Approach	% Increase
512×512	0.99894	.99600 [67]	.30
640×480	0.99908	.99250 [68]	5.17
800×600	0.99932	.99000 [69]	.94
1024×768	0.99957	.99710 [70]	.25

Average PSNR results obtained are: 50.69 for 512×512 images, 51.28 for 640×480 images, 53.17 for 800×600 images and 55.22 for 1024×768. The results ascertain creation of good quality watermarked images. It is also observed that with increasing image dimensions PSNR is also getting increased. Comparative analysis of PSNR obtained using our proposed approach with other approaches identified from literature is shown in Table 6.

M_X PSNR % increase of 23.81 is observed for 1024×768 images while MN PSNR % increase of 4.46 is observed for 640×480 images. The results obtained using the proposed approach delivers a PSNR higher than the existing techniques, thereby displaying a significant improvement. % increase in PSNR for different image dimensions is shown in Figure 8.

2) SSIM SSIM calculates the similarity among two images. It is based on the notion of HVS that measure the variation of structure between the original and the watermarked image. It matches luminance, contrast and structure among two images. Maximum value of 1 is attained if the two images are completely alike. SSIM is defined by the equation [9] (22).

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \qquad (22)$$

where, x, y are the image pixel positions; μ_x, μ_y are the mean values w.r.t. x and y; σ_x, σ_y are the standard deviation values w.r.t. x and y; C_1 and C_2 are the stability constants. SSIM is calculated for all image dimensions varying watermark size. Comparative analysis of SSIM obtained using proposed approach with previous approaches identified from literature is shown in Table 7.

Structural data present in an image have strong inter-pixel dependencies among spatial content. It lies in the range of -1 and 1. These dependencies carry significant evidence about the structure of the objects in the image. M_X SSIM % increase of 5.17 is observed for 640×480 images while M_N SSIM % increase of .25 is observed for 1024×768 images. Experimental results state an improvement of SSIM index in comparison to the previous approaches. If two images are alike by SSIM then perceptual quality of watermarked image is considered to be of good quality. % Increase in SSIM for different image dimensions is shown in Figure 9.

3) CC CC parameter identifies the association among two images. A positive correlation creates a CC value close to $+1$ while a negative correlation creates a CC value close to -1. The CC between original image and watermarked image computes image deformation at

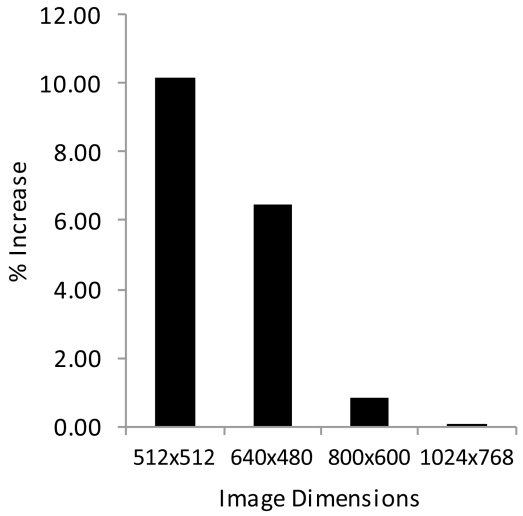


Fig. 10. % Increase in CC for different image dimensions.

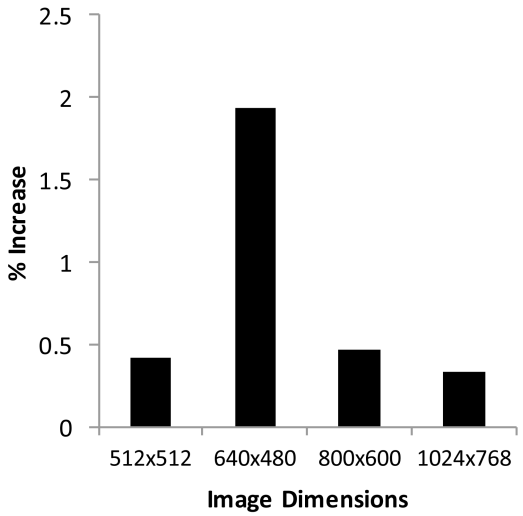


Fig. 11. % Increase in NPCR for different image dimensions.

TABLE 8.
Comparative analysis of CC obtained with previous approaches.

Image Dimensions	Proposed Approach	Previous Approach	% Increase
512×512	.99969	0.9074 [72]	10.17
640×480	.99978	0.9389 [71]	6.48
800×600	.99985	0.99166 [73]	0.83
1024×768	.99985	0.9992 [74]	0.07

pixels level. CC is calculated by the equation [71] (23).

$$C_{ab} = \frac{\frac{1}{r*c}\sum\sum(A_{i,j} - \bar{A})(B_{i,j} - \bar{B})}{\sqrt{\frac{1}{r*c}\sum\sum(A_{i,j} - \bar{A})^2}\sqrt{\frac{1}{r*c}\sum\sum(B_{i,j} - \bar{B})^2}} \tag{23}$$

$A_{i,j}$ and $B_{i,j}$ are the pixels in the ith row and jth column of images A and B; $\bar{A}$ is the mean of A while $\bar{B}$ is mean of B; r and c are the width and height of an image. CC is measured for all image dimensions with varying watermark size. Comparative analysis of CC obtained with previous approaches is shown in Table 8.

The closer CC value is to one, the better it is. Our approach generates a high positive CC which reveals a strong association among host image and watermarked image. M_X CC % increase of 10.17 is observed for 512×512 images while M_N CC % increase of .07 is observed for 1024×768 images. Experimental results state an improvement of CC in comparison to the previous approaches. % Increase in CC for different image dimensions is shown in Figure 10.

4) NPCR NPCR determines the total number of pixels altered between original image (I) and watermarked image (I'). It calculates the percentage of dissimilar pixel quantities between images. NPCR is calculated by equation [75] (24)

$$\text{NPCR} = \frac{\sum_{i=1}^{m}\sum_{i=1}^{n} p_{i,j}}{m*n} * 100\% \tag{24}$$

TABLE 9.
% Increase in NPCR for different image dimensions.

Image Dimensions	% Increase
512×512	0.42
640×480	1.94
800×600	0.47
1024×768	0.33

where,

$$p_{i,j} = 0, \text{ if } I_{i,j} = I'_{i,j}$$
$$1, \text{ if } I_{i,j} \neq I'_{i,j}$$

m, n are the width and height of the image; $p_{i,j}$ is an array of same size as I and I'. NPCR is evaluated for all image dimensions with varying watermark size. Average NPCR results obtained are: .11282 for 512×512 images, 0.10254 for 640×480 images, 0.07732 for 800×600 images and 0.05803 for 1024×768. Comparative ratio proportion reveals % increase in NPCR obtained using proposed approach with previous image encryption approaches [76–79]. % Increase in NPCR for different image dimensions is shown in Table 9.

NPCR parameter is used commonly in image encryption. The parameter identifies the number of pixels change rate between two ciphered images. For good NPCR encrypted image the change rate should be close to 100. For the first time NPCR parameter is explored in the field of watermarking. Since watermarking aims at prevention of image distortion between original and the watermarked image, so a good NPCR watermarked image will give value close to 0. Our average NPCR outcome for different image dimensions reveals a good assessment. M_X NPCR % increase of 1.94 is observed for 640×480 images while M_N NPCR % increase of .09 is observed for 1024×768 images. % Increase in NPCR for different image dimensions is shown in Figure 11.

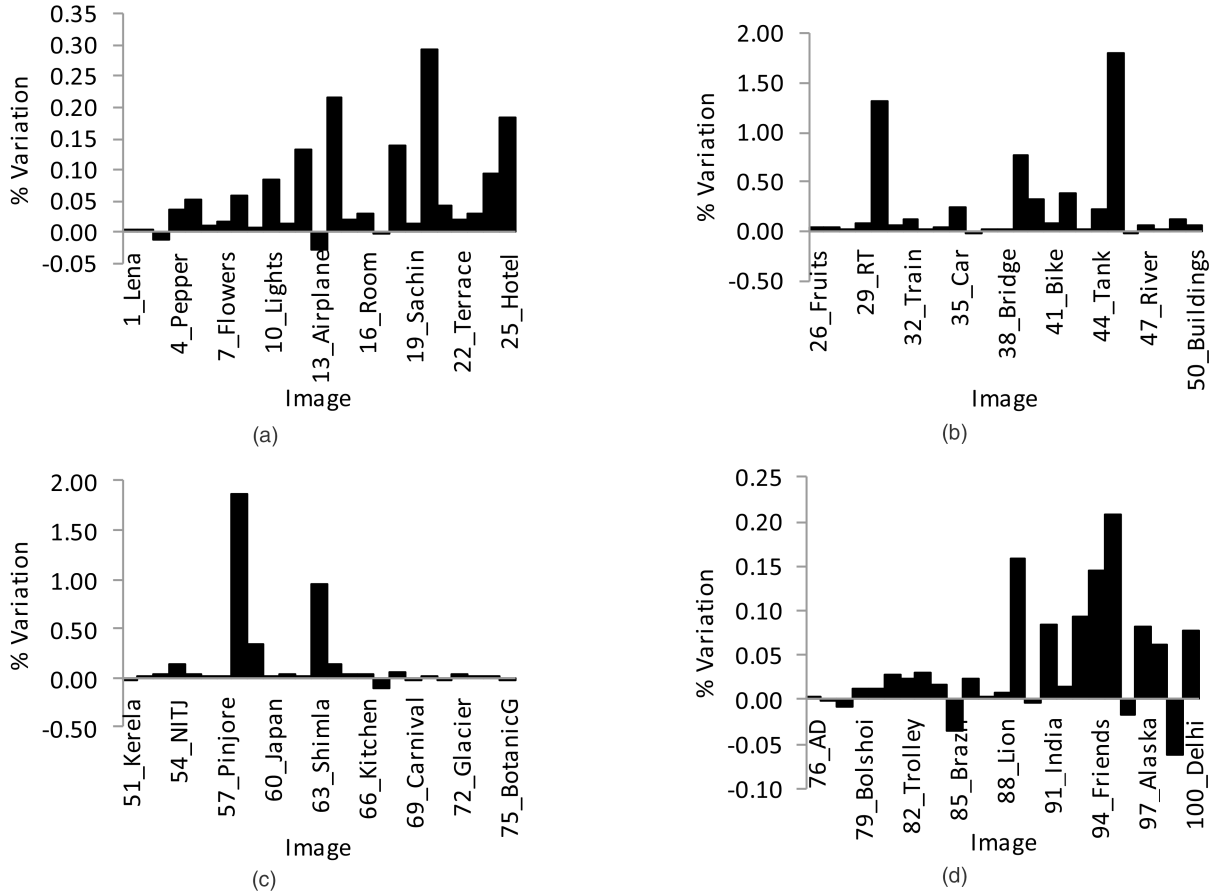


Fig. 12. % Variation in Entropy between original images and complete watermarked images for dimensions. (a) 512×512; (b) 640×480; (c) 800×600; (d) 1024×768.

5) Entropy Entropy is a statistical measure of uncertainty defined by the equation [80] (25)

$$E_G = \sum_{x \in G}^{n} p(x) \log \left(\frac{1}{p(x)} \right) \tag{25}$$

G is the data raised from a particular domain and $p(x)$ is the probability of sample in the group G. Entropy parameter ascertains existence of watermarks' imperceptibility. A watermarked image having high entropy has less perceivable distortion to human eye than an image with low entropy.

Entropy is estimated for all image dimensions with varying watermark size. Average Entropy results obtained are: 7.3847 for 512×512 images, 7.3653 for 640×480 images, 7.4225 for 800×600 images and 7.4733 for 1024×768 images. % Variation in Entropy between original images and complete watermarked images for all image dimensions are shown in Figure 12. Results show that the watermarks embedded in the image are highly imperceptible as the entropy values obtained are slightly more than the original image entropy. A higher disorder implies that more information can be embedded in the image without being perceived.

6) EPT and RPT EPT is the total computational time taken by the proposed watermarking scheme. It is measured in milliseconds (ms). The processing time achieved using the proposed approach depicts a small embedding time complexity as shown in Figure 13.

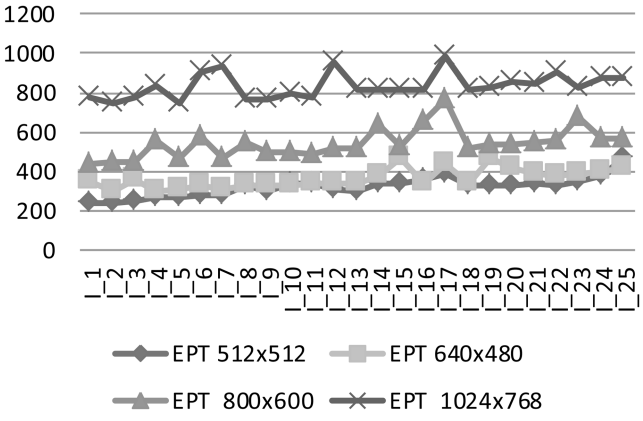


Fig. 13. % EPT for various image dimensions.

RPT is the total computational time taken by the proposed watermarking scheme. It is measured in milliseconds (ms). The processing time achieved using the proposed approach depicts a small retrieving time complexity as shown in Figure 14.

7) Robustness In order to test the robustness of the proposed approach, various attacks are launched against

TABLE 10.
NC for various attacks.

Attack	Previous Approach 512×512	512×512	640×480	800×600	1024×768
Salt and Pepper Noise	.9805	.9857	.9887	.9896	.9912
Gaussian Noise	.9800	.9859	.9873	.9894	.9921
Cropping	.9177	.9265	.9289	.9345	.9412
JPEG Compression	.9898	.9917	.9939	.9947	.9986
Median Filtering	.9112	.9225	.9312	.9319	.9418

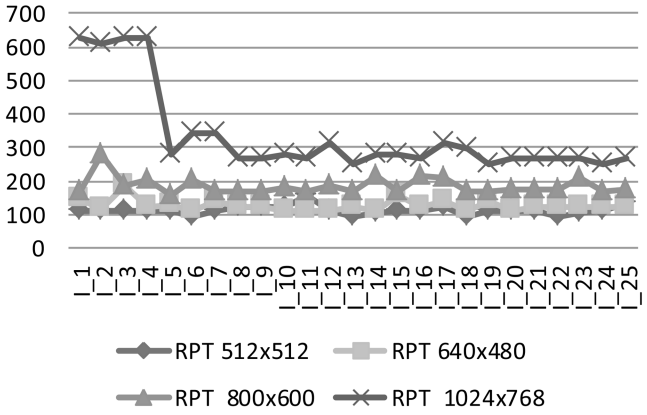


Fig. 14. % RPT for various image dimensions.

the watermarked image. These attacks include Salt and Pepper Noise, Gaussian Noise, Cropping, JPEG Compression and Median Filtering. Normalized Correlation (NC) performance of proposed algorithm against attacks on watermark embedded for a set of all image dimensions is shown in Table 10.

6. CONCLUSIONS

This paper proposes the problem of semi-blind and secure digital watermarking for authentic cation of images. It generates solution by providing confidentiality, integrity, and authenticity to the watermarked image. The objective is achieved by integrating digital watermarking & cryptography together in order to insert the secret information to gain a high level of privacy & efficiency. AES encrypted watermark with a secret key bundle makes it very challenging for the attacker to identify and hinder the watermark saved inside host image. ECDHP generates secret key based on discrete logarithm for solving key distribution problem among multiple owners and distributers. Insertion and removal of secure watermark using DCT domain provides low computational complexity and fast speed. The performances of the secure watermarking technique are compared on the basis of PSNR, SSIM, CC, NPCR and Entropy values. Quantitative analysis of image quality parameters reveals effectiveness of the proposed approach.

REFERENCES

[1] M. H. Pi, C. H. Li, and H. Li
A novel fractal image watermarking
IEEE Transactions on Multimedia, Vol. 8, No. 3, 2006, pp. 488–499.

[2] Y. Hu and B. Jeon
Reversible Visible Watermarking and Lossless Recovery of Original Images
IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 11, 2006, pp. 1423–1429.

[3] D. M. Thodi and J. J. Rodriguez
Expansion Embedding Techniques for Reversible Watermarking
IEEE Transactions on Image Processing, Vol. 16, No. 3, 2007, pp. 721–730.

[4] P. Y. Lin, J. S. Lee, and C.C. Chang
Dual Digital Watermarking for Internet Media Based on Hybrid Strategies
IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19, No. 8, 2009, pp. 1169–1177.

[5] T. Y. Liu and W. H. Tsai
Generic Lossless Visible Watermarking—A New Approach
IEEE Transactions on Image Processing, Vol. 19, No. 5, 2010, pp. 1224–1235.

[6] L. Kocarev, Z. Galias, and S. Lian
Intelligent Computing Based on Chaos
Springer, 2009.

[7] A. K. Parthasarathy and S. Kak
An Improved Method of Content Based Image Watermarking
IEEE Transactions on Broadcasting, Vol. 53, No. 2, 2007, pp. 468–479,

[8] L. O. M. Kobayashi, S. S. Furuie, and P. S. L. M. Barreto
Providing Integrity and Authenticity in DICOM Images: A Novel Approach
IEEE Transactions on Information Technology in Biomedicine, Vol. 13, No. 4, 2009, pp. 582–589.

[9] C. Chin-Chen and H. Chou
A New Public-Key Oblivious Fragile Watermarking for Image Authentication Using Discrete Cosine Transform
In Second International Conference on Future Generation Communication and Networking Symposia, 2008, pp. 11–14.

[10] C. Fei, R. H. Kwong, D. Kundur, and F. Chuhong
A Hypothesis Testing Approach to Semifragile Watermark-Based Authentication
IEEE Transactions on Information Forensics and Security, Vol. 4, No. 2, 2009, pp. 179–192.

[11] K. C. Liu and C. H. Chou
Robust and transparent watermarking scheme for colour images
IET Image Processing, Vol. 3, No. 4, 2009, pp. 228–242.

[12] T. Jen-Sheng, H. Win-Bin, and K. Yau-Hwang
On the Selection of Optimal Feature Region Set for Robust Digital Image Watermarking
IEEE Transactions on Image Processing, Vol. 20, No. 3, 2011, pp. 735–743.

[13] P. S. Huang, C. S. Chiang, C. P. Chang, and T. M. Tu
Robust spatial watermarking technique for colour images via direct saturation adjustment
IEE Proceedings—Vision, Image and Signal Processing, Vol. 152, No. 5, 2005, pp. 561–574.

[14] S. K. Singh, S. Kumar, M. Srivastava, A. Chandra, and S. Srivastava
Wavelet Based Robust Digital Watermarking Technique Using Reverse Additive Algorithm (RAA)
In Third UKSim European Symposium on Computer Modeling and Simulation, 2009, pp. 241–244.

[15] S. P. Mohanty, E. Kougianos, and N. Ranganathan
VLSI architecture and chip for combined invisible robust and fragile watermarking
IET Computers & Digital Techniques, Vol. 1, No. 5, 2007, pp. 600–611.

[16] G. Karakonstantis, N. Banerjee, and K. Roy
Process-Variation Resilient and Voltage-Scalable DCT Architecture for Robust Low-Power Computing
IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 18, No. 10, 2010, pp. 1461–1470.

[17] W. C. Chu
DCT-based image watermarking using subsampling
IEEE Transactions on Multimedia, Vol. 5, No. 1, 2003, pp. 34–38.

[18] X. Jun and W. Ying
Multiple Watermarking Based on Spread Transform
In 8th International Conference on Signal Processing, 2006, pp. 1–4.

[19] Z. Dong, W. Sha, and Z. Jiying
RST Invariant Image Watermarking Algorithm with Mathematical Modeling and Analysis of the Watermarking Processes
IEEE Transactions on Image Processing, Vol. 18, No. 5, 2009, pp. 1055–1068.

[20] Z. Ying, X. Jun, and W. Ying
Side informed image watermarking algorithm with high security
In IEEE Youth Conference on Information, Computing and Telecommunication, 2009, pp. 395–398.

[21] Y. Tan, L. Tang, Z. Gao, P. Sun, X. Yang, and Y. Li
A Rotation Resistant Image Watermarking Algorithm via Circle
In Eighth International Conference on Computational Intelligence and Security (CIS), 2012, pp. 461–463.

[22] P. Viswanathan, and P. V. Krishna
A Joint FED Watermarking System using Spatial Fusion for Verifying the Security Issues of Teleradiology
IEEE Journal of Biomedical and Health Informatics, Vol. 18, No. 3, pp. 753–764, 2014.

[23] V. M. Potdar, H. Song, and E. Chang
A survey of digital image watermarking techniques
In 3rd IEEE International Conference on Industrial Informatics, 2005, pp. 709–716.

[24] T. Stutz, F. Autrusseau, and A. Uhl
Non-Blind Structure-Preserving Substitution Watermarking of H.264/CAVLC Inter-Frames
IEEE Transactions on Multimedia, Vol. 16, No.5, pp. 1337–1349, 2014.

[25] L. Zhang, F. Qian, Y. Gao, and Y. Zhu
A New Integration Scheme of Robust and Fragile for Secured Digital Watermarking
International Colloquium on Computing, Communication, Control, and Management, 2008, pp. 312–316.

[26] Y.-W. Ding, Z. Lin, and L. Wang
A Multipurpose Public-Key Cryptosystem Based Image Watermarking,
In 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008, pp. 1–4.

[27] RBI
FAQ on Cheque Truncation Project in the National Capital Region
Department of Payment and Settlement Systems, 2010.

[28] RBI
Payment and Settlement Systems and Information Technology
Reserve Bank of India Annual Report 12-13, 2013. pp. 128–139

[29] R. Eswaraiah, and E. Sreenivasa Reddy
Robust medical image watermarking technique for accurate detection of tampers inside region of interest and recovering original region of interest
IET Image Processing, Vol. 9, No. 8, 2015, pp. 615–625.

[30] X. Li, X. Sun and L. Quansheng
Image Integrity Authentication Scheme Based on Fixed Point Theory
IEEE Transactions on Image Processing, Vol. 24, No. 2, 2015, pp. 632–635

[31] A. B. Jeng and C. Lo-Yi
How to enhance the security of e-Passport
In International Conference on Machine Learning and Cybernetics, 2009, pp. 2922–2926.

[32] S. Kundra, A. Dureja, and R. Bhatnagar
The study of recent technologies used in E-passport system
In IEEE Global Humanitarian Technology Conference—South Asia Satellite (GHTC-SAS), 2014, pp. 141–146.

[33] M. Q. Saeed, A. Masood, and F. Kausar
Securing ePassport system: A proposed Anti-Cloning and Anti-Skimming Protocol
In 17th International Conference on Software, Telecommunications & Computer Networks, 2009, pp. 90–94.

[34] K. D. Akdemir, D. Karakoyunlu, and B. Sunar
Non-linear error detection for elliptic curve cryptosystems
IET Information Security, Vol. 6, No. 1, 2012, pp. 28–40.

[35] L. Jyu-Yuan and H. Chih-Tsun
Energy-Adaptive Dual-Field Processor for High-Performance Elliptic Curve Cryptographic Applications
IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 19, No. 8, 2011, pp. 1512–1517.

[36] D. M. Schinianakis, A. P. Fournaris, H. E. Michail, A. P. Kakarountas, and T. Stouraitis
An RNS Implementation of an Elliptic Curve Point Multiplier
IEEE Transactions on Circuits and Systems, Vol. 56, No. 6, pp. 2009, 1202–1213.

[37] NIST
Digital Signature Standard
In Federal Information Processing Standards Publications FIPS PUB 186-2, 2000, pp. 1–73.

[38] NIST
Mathematical routines for the NIST prime elliptic curves
In National Security Agency, 2010, pp. 1–43.

[39] NIST
Recommended Elliptic Curves for Federal Government
Use http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf, Date of Accession 19-11-2013.

[40] K. Ananyi, H. Alrimeih, and D. Rakhmatov
Flexible Hardware Processor for Elliptic Curve Cryptography Over NIST Prime Fields
IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 17, No. 8, 2009, pp. 1099–1112.

[41] O. S. Althobaiti and H. A. Aboalsamh
An enhanced Elliptic Curve Cryptography for biometric
In 7th International Conference on Computing and Convergence Technology (ICCCT), 2012, pp. 1048–1055.

[42] L. Tawalbeh, M. Mowafi, and W. Aljoby
Use of elliptic curve cryptography for multimedia encryption
IET Information Security, Vol. 7, No. 2, 2013, pp. 67–74.

[43] A. Cilardo, L. Coppolino, N. Mazzocca, and L. Romano
Elliptic Curve Cryptography Engineering
Proceedings of the IEEE, Vol. 94, No. 2, 2006, pp. 395–406.

[44] M. Amara and A. Siad
Elliptic Curve Cryptography and its applications
In 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA), 2011, pp. 247–250.

[45] A. Sachan, S. Emmanuel, A. Das, and M. S. Kankanhalli
Privacy Preserving Multiparty Multilevel DRM Architecture
In 6th IEEE Consumer Communications and Networking Conference, 2009, pp. 1–5.

[46] D. Mishra and S. Mukhopadhyay
Privacy preserving hierarchical content distribution in multiparty multilevel DRM
In World Congress on Information and Communication Technologies (WICT), 2012, pp. 525–530.

[47] L. Harn, W. J. Hsin, and M. Mehta
Authenticated Diffie-Hellman key agreement protocol using a single cryptographic assumption
IEE Proceedings—Communications, Vol. 152, No. 4, 2005, pp. 404–410.

[48] G. P. Biswas
Diffie-Hellman technique: extended to multiple two-party keys and one multi-party key
IET Information Security, Vol. 2, No. 1, 2008, pp. 12–18.

[49] M. Abid and H. Afifi
Secure E-Passport Protocol Using Elliptic Curve Diffie-Hellman Key Agreement Protocol
In Fourth International Conference on Information Assurance and Security, 2008, pp. 99–102.

[50] J. R. Vacca
Computer and Information Security
1st ed.: Morgan Kaufmann Publishers, 2009.

[51] X. Li, J. Chen, D. Qin, and W. Wan
Research and realization based on hybrid encryption algorithm of improved AES and ECC
In International Conference on Audio Language and Image Processing (ICALIP), 2010, pp. 396–400.

[52] NIST
Advanced Encryption Standard
In Federal Information Processing Standards, 2001, pp. 1–47.

[53] W. E. Burr
Selecting the Advanced Encryption Standard
IEEE Security & Privacy, Vol. 1, No. 2, 2003, pp. 43–52.

[54] M. Mozaffari-Kermani and A. Reyhani-Masoleh
Efficient and High-Performance Parallel Hardware Architectures for the AES-GCM
IEEE Transactions on Computers, Vol. 61, No. 8, 2012, pp. 1165–1178.

[55] R. Banu and T. Vladimirova
Fault-Tolerant Encryption for Space Applications
IEEE Transactions on Aerospace and Electronic Systems, Vol. 45, No. 1, 2009, pp. 266–279.

[56] L. Bin and B. M. Baas
Parallel AES Encryption Engines for Many-Core Processor Arrays
IEEE Transactions on Computers, Vol. 62, No. 3, pp. 2013, 536–547.

[57] R. Setchi, I. Jordanov, R. J. Howlett, and L. C. Jain
Knowledge-Based and Intelligent Information and Engineering Systems
Springer, 2010.

[58] S. P. Noolu and M. S. Baghini
Comments on An Analog 2-D DCT Processor
IEEE Transactions on Circuits and Systems for Video Technology, Vol. 20, No. 8, 2010, pp. 1162–1163.

[59] T. Dutoit and F. Marques
Applied Signal Processing
Springer, 2009.

[60] E. L. Tan, W. S. Gan, and S. K. Mitra
Fast arbitrary resizing of images in the discrete cosine transform domain
IET Image Processing, Vol. 5, No. 1, 2011, pp. 73–86.

[61] E. Walia and A. Suneja
Fragile and blind watermarking technique based on Weber's law for medical image authentication
Computer Vision, Vol. 7, No. 1, 2013, pp. 9–19.

[62] J. Hammerle-Uhl, C. Koidl, and A. Uhl
Multiple blind re-watermarking with quantisation-based embedding
In 18th IEEE International Conference on Image Processing (ICIP), 2011, pp. 265–268.

[63] R. Rosenbaum, G. Fuchs, and H. Schumann
Region-wise meta-data in JPEG2000-encoded imagery
In 5th International Conference on Visual Information Engineering, 2008, pp. 741–746.

[64] E. Halici and A. A. Alatan
Watermarking for depth image based rendering
In IEEE International Conference on Image Processing (ICIP), 2009, pp. 4217–4220.

[65] T. Hui Li, L. Zhengguo, T. Yih Han, S. Rahardja, and Y. Chuohuo
A Perceptually Relevant MSE-Based Image Quality Metric
IEEE Transactions on Image Processing, Vol. 22, No. 11, 2013, pp. 4447–4459.

[66] T. Hui Li, L. Zhengguo, T. Yih Han, S. Rahardja, and Y. Chuohuo
A Perceptually Relevant MSE-Based Image Quality Metric
IEEE Transactions on Image Processing, Vol. 22, No. 11, 2013, pp. 4447–4459.

[67] A. Kunhu and H. Al-Ahmad
Multi watermarking algorithm based on DCT and hash functions for color satellite images
In 9th International Conference on Innovations in Information Technology (IIT), 2013, pp. 30–35.

[68] H. Heechul and S. Kwanghoon
Automatic illumination and color compensation using mean shift and sigma filter
IEEE Transactions on Consumer Electronics, Vol. 55, No. 3, pp. 2009, 978–986.

[69] R. F. Lopes, C. D. M. Regis, W. T. A. Lopes, and M. S. de Alencar
AdaptVoD—An Adaptive Video-on-Demand Platform for Mobile Devices
In 5th FTRA International Conference on Multimedia and Ubiquitous Engineering (MUE), 2011, pp. 257–262.

[70] D. V. S. X. De Silva, W. A. C. Fernando, S. T. Worrall, and A. M. Kondoz
A novel depth map quality metric and its usage in depth map coding
In 3DTV Conference: The True Vision—Capture, Transmission and Display of 3D Video (3DTV-CON), 2011, pp. 1–4.

[71] S. Philipp, K. Stephan, E. Wolfgang, and T. Niels
Semi-automatic registration of videos for improved watermark detection
In Proceedings of the first annual ACM SIGMM conference on Multimedia systems Phoenix, Arizona, USA: ACM, 2010, pp. 23–34.

[72] H. A. Al-Otum and A. O. Al-Taba'a
Color image copyright ownership protection based on a multi-spectral selective pixel-wise watermarking technique
In 3rd International Symposium on Communications, Control and Signal Processing (ISCCSP), 2008, pp. 544–549.

[73] L. Cheng-Liang and C. Yi-Shiang
The application of intelligent system to digital image forensics
In International Conference on Machine Learning and Cybernetics, 2009, pp. 2991–2998.

[74] S. Kwong, H. Yuan, J. Liu, and J. Sun
A Novel Distortion Model and Lagrangian Multiplier for Depth Maps Coding
IEEE Transactions on Circuits and Systems for Video Technology, Vol. 25, No. 99, 2014, pp. 443–451.

[75] A. Umamageswari and G. R. Suresh
Security in medical image communication with arnold's cat map method and reversible watermarking
In International Conference on Circuits, Power and Computing Technologies (ICCPCT), 2013, pp. 1116–1121.

[76] M. S. El-Mahallawy, E. A. Hagras, A. Z. Eldin, and M. W. Fakhr
Robust Blind and Secure Biometric Watermarking Based on Partial Multi-Map Chaotic Encryption
In 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2011, pp. 1–5.

[77] K. P. Narendra, P. Vinod, and K. S. Krishan
Diffusion-substitution based gray image encryption scheme
Digital Signal Processing, Vol. 23, No. 3, 2013, pp. 894–901.

[78] N. K. Pareek, V. Patidar, and K. K. Sud
Substitution-diffusion based Image Cipher
International Journal of Network Security & Its Applications, Vol. 3, No. 2, 2011, pp. 149–160.

[79] S. M. Seyedzadeh and Y. Hashemi
Image encryption algorithm based on Choquet Fuzzy Integral with self-adaptive pseudo-random number generator
In 11th International Conference on Intelligent Systems Design and Applications (ISDA), 2011, pp. 642–647.

[80] Y. Qiu, Z. Yana, Y. Cheng, and L. Wei
Information Entropy Used in Digital Watermarking
In Symposium on Photonics and Optoelectronics (SOPO), 2012, pp. 1–4.

Vineet Mehan received the B.Tech. degree in Information Technology from Kurukshetra University in 2003. He received the M.E. degree in Computer Science and Engineering from NITTTR, Panjab University in 2008. He completed Ph.D. degree in Computer Science and Engineering from Dr. B.R. Ambedkar National Institute of Technology, Jalandhar in 2016. His research interests include Image Processing, Watermarking and Cryptographic Algorithms.

JAIF

Journal of Advances in Information Fusion

A semi-annual archival publication of the
International Society of Information Fusion

Volumes 1–12 Index

A

Adurthi, N., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Ala-Luhtala, J., *see* Raitoharju, M., *JAIF*, **11**, 1 (June 2016), 3–14.
Alford, M., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Ali-Löytty, S., *see* Raitoharju, M., *JAIF*, **11**, 1 (June 2016), 3–14.
Alsun, M., Lecornu, L., Solaiman, B., Possibilistic Medical Knowledge Representation Model, *JAIF*, **7**, 2 (December 2012), 101–113.
An, W., Singh, S., Pattipati, K. R., Kleinman, D. L., and Gokhale, S. S., Dynamic Scheduling of Multiple Hidden Markov Model-Based Sensors, *JAIF*, **3**, 1 (July 2008), 33–49.
Anderson, S. L., *see* Stone, L. D., *JAIF*, **10**, 1 (June 2015), 3–12.
Andler, S. F., *see* Karlsson, A., *JAIF*, **6**, 2 (December 2011), 150–166.
Aravinthan, A., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
Areta, J., Bar-Shalom, Y., and Pattipati, K. R., T2T and M2T Association with Combined Hypotheses, *JAIF*, **4**, 1 (July 2009), 40–51.
Areta, J., Bar-Shalom, Y., and Rothrock, R., Misassociation Probability in M2TA and T2TA, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Areta, J., Bar-Shalom, Y., Levedahl, M., and Pattipati, K. R., Hierarchical Track Association and Fusion for a Networked Surveillance System, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Arnborg, S., *see* Brynielsson, J., *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Arnborg, S., Robust Bayesianism: Relation to Evidence Theory *JAIF*, **1**, 1 (July 2006), 75–90.
Aughenbaugh, J. M. and La Cour, B. R., Measurement-Guided Likelihood Sampling for Grid-Based Bayesian Tracking *JAIF*, **5**, 2 (Dec. 2010), 108–127.
Avasarala, V., Mullen, T., and Hall, D., A Market-based Approach to Sensor Management, *JAIF*, **4**, 1 (July 2009), 52–71.

B

Bab-Hadiashar, A., *see* Hoseinnezhad, R., *JAIF*, **1**, 1 (July 2006), 52–62.
Balasingam, B., *see* Choi, S., *JAIF*, **8**, 2 (December 2013), 143–155.
Balasingam, B., *see* Pasupuleti, D., *JAIF*, **12**, 1 (June 2017), 41–57.
Bar-Shalom, Y., *see* Areta, J., *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Bar-Shalom, Y., *see* Areta, J., *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Bar-Shalom, Y., *see* Areta, J., *JAIF*, **4**, 1 (July 2009), 40–51.
Bar-Shalom, Y., *see* Belfadel, D., *JAIF*, **9**, 2 (December 2014), 59–74.
Bar-Shalom, Y., *see* Belfadel, D., *JAIF*, **10**, 2 (December 2015), 101–112.
Bar-Shalom, Y., *see* Belfadel, D., *JAIF*, **12**, 1 (June 2017), 58–72.
Bar-Shalom, Y., *see* Bordonaro, S., *JAIF*, **12**, 2 (December 2017), 228–242.
Bar-Shalom, Y. and Chen, H., Covariance Reconstruction for Track Fusion with Legacy Track Sources *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Bar-Shalom, Y. and Chen, H., Multisensor Track-to-Track Association for Tracks with Dependent Errors *JAIF*, **1**, 1 (July 2006), 3–14.
Bar-Shalom, Y. and Chen, H., Track-to-Track Association Using Attributes *JAIF*, **2**, 1 (July 2007), 49–59.
Bar-Shalom, Y., *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.
Bar-Shalom, Y., *see* Dou, W., *JAIF*, **10**, 2 (December 2015), 163–182.
Bar-Shalom, Y., *see* Huang, H. A. J., *JAIF*, **12**, 1 (June 2017), 110–124.
Bar-Shalom, Y., *see* Osborne, R. W., III, *JAIF*, **7**, 1 (June 2012), 3–15.
Bar-Shalom, Y., *see* Osborne, R. W., III, *JAIF*, **9**, 2 (December 2014), 75–89.
Bar-Shalom, Y., *see* Osborne, III, R. W., *JAIF*, **10**, 2 (December 2015), 199–210.
Bar-Shalom, Y., *see* Ravindra, V. C., *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Bar-Shalom, Y., *see* Rodningsby, A., *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Bar-Shalom, Y., *see* Romeo, K., *JAIF*, **10**, 2 (December 2015), 113–124.
Bar-Shalom, Y., *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.
Bar-Shalom, Y., *see* Tian, X., *JAIF*, **4**, 2 (Dec. 2009), 146–164.
Bar-Shalom, Y., *see* Tian, X., *JAIF*, **5**, 1 (July 2010), 3–17.
Bar-Shalom, Y., *see* Tian, X., *JAIF*, **5**, 2 (Dec. 2010), 128–138.
Bar-Shalom, Y., *see* Yang, R., *JAIF*, **12**, 1 (June 2017), 3–19.
Bar-Shalom, Y., *see* Yuan, T., *JAIF*, **6**, 2 (December 2011), 131–149.
Bar-Shalom, Y., *see* Zhang, S., *JAIF*, **6**, 1 (June 2011), 3–23.
Bar-Shalom, Y., *see* Zhang, S., *JAIF*, **9**, 1 (July 2014), 38–46.
Baum, M., *see* Bordonaro, S., *JAIF*, **12**, 2 (December 2017), 228–242.
Baum, M., *see* Faion, F., *JAIF*, **10**, 1 (June 2015), 13–30.
Baum, M., *see* Granström, K., *JAIF*, **12**, 2 (December 2017), 139–174.
Baum, M., *see* Pasupuleti, D., *JAIF*, **12**, 1 (June 2017), 41–57.
Belaroussi, R., Prevost, L., and Milgram, M., Algorithms Fusion for Face Localization, *JAIF*, **1**, 1 (July 2006), 35–51.
Belfadel, D., Osborne, R. W., III, Bar-Shalom, Y., Bias Estimation and Observability for Optical Sensor Measurements with Targets of Opportunity, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., Osborne, R. W., III, Bar-Shalom, Y., Bias Estimation for Moving Optical Sensor Measurements with Targets of Opportunity, *JAIF*, **10**, 2 (December 2015), 101–112.
Belfadel, D., Osborne, III, R. W., Bar-Shalom, Y., Pattipati, K., Space Based Sensor Bias Estimation in the Presence of Data Association Uncertainty, *JAIF*, **12**, 1 (June 2017), 58–72.
Benaskeur, A. R., Rhéaume, F., and Paradis, S., Target Engageability Improvement through Adaptive Tracking, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Biermann, J., Hörling, P., Snidaro, L., Experiences and Challenges in Automated Support for Intelligence in Asymmetric Operations, *JAIF*, **8**, 2 (December 2013), 101–118.
Blair, W. D. and Miceli, P. A., Performance Prediction of Multisensor Tracking Systems for Single Maneuvering Targets *JAIF*, **7**, 1 (June 2012), 28–45.
Blasch, E., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.
Blasch, E., *see* Duník, J., *JAIF*, **11**, 1 (June 2016), 91–109.
Blasch, E., Kadar, I., Salerno, J., Kokar, M. M., Das, S., Powell, G. M., Corkill, D. D., and Ruspini, E. H., Issues and Challenges in Situation Assessment (Level 2 Fusion), *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Blasch, E., *see* Kahler, B., *JAIF*, **6**, 2 (December 2011), 101–118.
Blasch, E., *see* Yang, C., *JAIF*, **3**, 1 (July 2008), 14–32.
Blasch, E., *see* Zheng, Y., *JAIF*, **9**, 2 (December 2014), 124–135.
Bloem, E. A., *see* Blom, H. A. P., *JAIF*, **1**, 1 (July 2006), 15–34.
Blom, H. A. P. and Bloem, E. A., Joint Particle Filtering of Multiple Maneuvering Targets From Unassociated Measurements *JAIF*, **1**, 1 (July 2006), 15–34.
Bordonaro, S., Willett, P., Bar-Shalom, Y., Baum, M., Luginbuhl, T., Extended Object Tracking with Exploitation of Range Rate Measurements, *JAIF*, **12**, 2 (December 2017), 228–242.
Bossé, É., *see* Valin, P., *JAIF*, **5**, 1 (July 2010), 32–40.
Braca, P., *see* Vivone, G., *JAIF*, **12**, 2 (December 2017), 189–210.
Brynielsson, J. and Arnborg, S., An Information Fusion Game Component *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Bubalo, A., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Bukal, M., *see* Marković, I., *JAIF*, **11**, 2 (December 2016), 157–172.
Bursik, M. I., *see* Rogova, G. L., *JAIF*, **3**, 2 (Dec. 2008), 118–128.

C

Campbell, M., *see* Wyffels, K., *JAIF*, **12**, 2 (December 2017), 211–227.
Carthel, C., *see* Coraluppi, S., *JAIF*, **5**, 1 (July 2010), 18–31.
Carthel, C., *see* Coraluppi, S., *JAIF*, **6**, 1 (June 2011), 57–67.
Carthel, C., *see* Coraluppi, S., *JAIF*, **7**, 2 (December 2012), 153–164.
Ćesić, J., *see* Marković, I., *JAIF*, **11**, 2 (December 2016), 157–172.
Chakravorty, R. and Challa, S., Augmented State Integrated Probabilistic Data Association Smoothing for Automatic Track Initiation in Clutter *JAIF*, **1**, 1 (July 2006), 63–74.
Carvalho, R. N. and Chang, K., A Fusion Analysis and Evaluation Tool for Multi-Sensor Classification Systems *JAIF*, **7**, 2 (December 2012), 141–152.
Challa, S., *see* Chakravorty, R., *JAIF*, **1**, 1 (July 2006), 63–74.
Chang, K., *see* Carvalho, R. N., *JAIF*, **7**, 2 (December 2012), 141–152.
Chang, K. C. and Hill, J. P., Level I and Level II Target Valuations for Sensor Management *JAIF*, **1**, 2 (Dec. 2006), 95–107.
Chang, K. C., *see* Mori, S., *JAIF*, **12**, 1 (June 2017), 95–109.
Chen, G., Shen, D., Kwan, C., Cruz, J. B., Jr., Kruger, M., and Blasch, E., Game Theoretic Approach to Threat Prediction and Situation Awareness, *JAIF*, **2**, 1 (July 2007), 35–48.
Chanussot, J., *see* Vivone, G., *JAIF*, **12**, 2 (December 2017), 189–210.
Chen, H., *see* Bar-Shalom, Y., *JAIF*, **1**, 1 (July 2006), 3–14.
Chen, H., *see* Bar-Shalom, Y., *JAIF*, **2**, 1 (July 2007), 49–59.
Chen, H., *see* Bar-Shalom, Y., *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Choi, S., Balasingam, B., Willett, P., Testing Under Communication Constraints, *JAIF*, **8**, 2 (December 2013), 143–155.
Choi, S., Willett, P., Zhou, S., The PMHT for Passive Radar in a DAB/DVB Network, *JAIF*, **9**, 1 (July 2014), 27–37.
Chong, C-Y., *see* Mori, S., *JAIF*, **12**, 1 (June 2017), 95–109.
Commeau, G., *see* Pasupuleti, D., *JAIF*, **12**, 1 (June 2017), 41–57.
Coraluppi, S. and Carthel, C., Multi-Stage Multiple-Hypothesis Tracking *JAIF*, **6**, 1 (June 2011), 57–67.
Coraluppi, S. and Carthel, C., Modified Scoring in Multiple-Hypothesis Tracking *JAIF*, **7**, 2 (December 2012), 153–164.
Coraluppi, S., *see* Erdinc, O., *JAIF*, **2**, 1 (July 2007), 22–34.
Coraluppi, S., Guerriero, M., Willett, P., and Carthel, C., Fuse-before-Track in Large Sensor Networks, *JAIF*, **5**, 1 (July 2010), 18–31.
Corkill, D. D., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Cornacchia, M., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Crassidis, J. L., *see* George, J., *JAIF*, **6**, 1 (June 2011), 39–56.
Crouse, D. F., Guerriero, M., and Willett, P., A Critical Look at the PMHT, *JAIF*, **4**, 2 (Dec. 2009), 93–116.

Crouse, D. F., Osborne, R. W., III, Pattipati, K., Willett, P., Bar-Shalom, Y., Efficient 2D Sensor Location Estimation using Targets of Opportunity, *JAIF*, **8**, 1 (July 2013), 73–89.
Crouse, D. F., Simulating Aerial Targets in 3D Accounting for the Earth's Curvature *JAIF*, **10**, 1 (June 2015), 31–57.
Crouse, D. F., Zheng, L., Willett, P., Corrections to "A Critical Look at the PMHT", *JAIF*, **7**, 1 (June 2012), 97–98.
Cruz, J. B., Jr., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.

D

Damarla, T., *see* Ravindra, V. C., *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Darling, J. E. and Demars, K. Ju., Uncertainty Propagation of Correlated Quaternion and Euclidean States Using the Gauss-Bingham Density *JAIF*, **11**, 2 (December 2016), 186–205.
Das, S., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Davey, S. J., Detecting a Small Boat using Histogram PMHT *JAIF*, **6**, 2 (December 2011), 167–186.
Degen, C., Govaers, F., Koch, W., Tracking Targets with Multiple Measurements per Scan Using the Generalized PHD Filter, *JAIF*, **10**, 2 (December 2015), 125–142.
Demars, K. Ju., *see* Darling, J. E., *JAIF*, **11**, 2 (December 2016), 186–205.
Dezert, J., *see* Martin, A., *JAIF*, **3**, 2 (Dec. 2008), 67–89.
Djiknavorian, P., *see* Valin, P., *JAIF*, **5**, 1 (July 2010), 32–40.
Dorigo, F., *see* Pasupuleti, D., *JAIF*, **12**, 1 (June 2017), 41–57.
Dou, W., Bar-Shalom, Y., Willett, P., Bistatic Measurement Fusion from Multistatic Configurations for Air Collision Warning, *JAIF*, **10**, 2 (December 2015), 163–182.
Duník, J., Straka, O., Šimandl, M., Blasch, E., Sigma-Point Set Rotation for Derivative-Free Filters in Target Tracking Applications, *JAIF*, **11**, 1 (June 2016), 91–109.

E

Eilbert, J., *see* Schrag, R. C., *JAIF*, **2**, 2 (Dec. 2007), 77–98.
Erdinc, O., Willett, P., and Coraluppi, S., Multistatic Sensor Placement: A Tracking Approach, *JAIF*, **2**, 1 (July 2007), 22–34.

F

Fahrny, J., *see* Pasupuleti, D., *JAIF*, **12**, 1 (June 2017), 41–57.
Faion, F., Zea, A., Baum, M., Hanebeck, U. D., Symmetries in Bayesian Extended Object Tracking, *JAIF*, **10**, 1 (June 2015), 13–30.
Falkman, G., *see* Johansson, F., *JAIF*, **6**, 2 (December 2011), 187–199.
Ferry, J. P., Exact Association Probability for Data with Bias and Features *JAIF*, **5**, 1 (July 2010), 41–67.
Flamant, J., *see* Le Bihan, N., *JAIF*, **11**, 2 (December 2016), 173–185.
Foo, P. H. and Ng, G. W., High-level Information Fusion: An Overview *JAIF*, **8**, 1 (July 2013), 33–72.
Foo, P. H., Ng, G. W., Ng, K. H., and Yang, R., Application of Intent Inference for Air Defense and Conformance Monitoring, *JAIF*, **4**, 1 (July 2009), 3–26.
Fosbury, A. M., *see* George, J., *JAIF*, **6**, 1 (June 2011), 39–56.

G

George, J., Crassidis, J. L., Singh, T., Fosbury, A. M., Anomaly Detection using Context-Aided Target Tracking, *JAIF*, **6**, 1 (June 2011), 39–56.
George, J. and Kaplan, L. M., Shooter Localization using a Wireless Sensor Network of Soldier-Worn Gunfire Detection Systems *JAIF*, **8**, 1 (July 2013), 15–32.
George, J., *see* Osborne, R. W., III, *JAIF*, **9**, 2 (December 2014), 75–89.
Georgescu, R., Willett, P., Marano, S., Matta, V., Predetection Fusion in Large Sensor Networks with Unknown Target Locations, *JAIF*, **7**, 1 (June 2012), 61–77.
Gilitschenski, I., *see* Kurz, G., *JAIF*, **9**, 2 (December 2014), 90–105.
Gilitschenski, I., *see* Kurz, G., *JAIF*, **11**, 2 (December 2016), 138–156.
Glattetre, J., *see* Rodningsby, A., *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Goger, P., *see* Schrag, R. C., *JAIF*, **2**, 2 (Dec. 2007), 77–98.
Goker, A., *see* Kaliciak, L., *JAIF*, **10**, 2 (December 2015), 183–198.
Gokhale, S. S., *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.
Govaers, F., *see* Degen, C., *JAIF*, **10**, 2 (December 2015), 125–142.
Granström, K., Baum, M., Reuter, S., Extended Object Tracking: Introduction, Overview, and Applications, *JAIF*, **12**, 2 (December 2017), 139–174.
Granström, K., *see* Vivone, G., *JAIF*, **12**, 2 (December 2017), 189–210.
Guerriero, M., *see* Coraluppi, S., *JAIF*, **5**, 1 (July 2010), 18–31.
Guerriero, M., *see* Crouse, D. F., *JAIF*, **4**, 2 (Dec. 2009), 93–116.
Guitouni, A., *see* Jabeur, K., *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Gustafsson, F., *see* Roth, M., *JAIF*, **11**, 1 (June 2016), 47–70.

H

Habtemariam, B. K., Aravinthan, A., Tharmarasa, R., Punithakumar, K., Lang, T., Kirubarajan, T., Distributed Tracking with a PHD Filter using Efficient Measurement Encoding, *JAIF*, **7**, 2 (December 2012), 114–130.
Hall, D., *see* Avasarala, V., *JAIF*, **4**, 1 (July 2009), 52–71.
Hallingstad, O., *see* Rodningsby, A., *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Hamp, Q. and Reindl, L., Association Performance Enhancement Through Classification *JAIF*, **7**, 2 (December 2012), 131–140.
Han, X., Yu, F., Levchuck, G., Pattipati, K., Tu, F., A Probabilistic Computational Model for Identifying Organizational Structures from Uncertain Activity Data, *JAIF*, **7**, 1 (June 2012), 78–96.
Hanebeck, U. D., *see* Faion, F., *JAIF*, **10**, 1 (June 2015), 13–30.
Hanebeck, U. D., *see* Kurz, G., *JAIF*, **9**, 1 (July 2014), 13–26.
Hanebeck, U. D., *see* Kurz, G., *JAIF*, **9**, 2 (December 2014), 90–105.
Hanebeck, U. D., *see* Kurz, G., *JAIF*, **11**, 2 (December 2016), 138–156.
Hanebeck, U. D., *see* Pfaff, F., *JAIF*, **11**, 2 (December 2016), 206–226.
Hanebeck, U. D., *see* Steinbring, J., *JAIF*, **9**, 2 (December 2014), 106–123.
Hanebeck, U. D., *see* Steinbring, J., *JAIF*, **11**, 1 (June 2016), 71–90.
Hanson-Hedgecock, S., *see* Rogova, G. L., *JAIF*, **3**, 2 (Dec. 2008), 118–128.
Hartikainen, J., *see* Särkkä, S., *JAIF*, **11**, 1 (June 2016), 31–46.
Hendeby, G., *see* Roth, M., *JAIF*, **11**, 1 (June 2016), 47–70.
Hill, J. P., *see* Chang, K. C., *JAIF*, **1**, 2 (Dec. 2006), 95–107.
Holender, M., *see* Papageorgiou, D. J., *JAIF*, **6**, 2 (December 2011), 77–100.
Holender, M., *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.
Hörling, P., *see* Biermann, J., *JAIF*, **8**, 2 (December 2013), 101–118.
Hoseinnezhad, R. and Bab-Hadiashar, A., Fusion of Redundant Information in Brake-By-Wire Systems Using a Fuzzy Voter *JAIF*, **1**, 1 (July 2006), 52–62.
Howell, C. and Moyer, S., Establishment of Human Performance Baseline for Image Fusion Algorithms in the LWIR and SWIR Spectra *JAIF*, **8**, 2 (December 2013), 133–142.
Huang, H. A. J., Bar-Shalom, Y., Yang, R., Ng, G. W., Tracking a Maneuvering Target Using Two Heterogeneous Passive Sensors on a Single Stationary Platform with IMM Estimation, *JAIF*, **12**, 1 (June 2017), 110–124.

J

Jabeur, K. and Guitouni, A., A Generalized Framework for Multi-Criteria Classifiers with Automated Learning: Application on FLIR Ship Imagery *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Jauffret, C., Pillon, D., Pignol, A-C., Leg-by-leg Bearings-Only Target Motion Analysis Without Observer Maneuver, *JAIF*, **6**, 1 (June 2011), 24–38.
Jentschel, H-J., *see* Junghans, M., *JAIF*, **3**, 1 (July 2008), 50–62.
Jilkov, V. P. and Li, X. R., On Fusion of Multiple Objectives for UAV Search & Track Path Optimization *JAIF*, **4**, 1 (July 2009), 27–39.
Jilkov, V. P. and Wu, J., Efficient GPU-Accelerated Implementation of Particle and Particle Flow Filters for Target Tracking *JAIF*, **10**, 1 (June 2015), 73–88.
Johansson, F. and Falkman, G., Real-time Allocation of Firing Units To Hostile Targets *JAIF*, **6**, 2 (December 2011), 187–199.
Johansson, R., *see* Karlsson, A., *JAIF*, **6**, 2 (December 2011), 150–166.
Julier, S., *see* Kurz, G., *JAIF*, **9**, 2 (December 2014), 90–105.
Jones, E., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Junghans, M. and Jentschel, H-J., Correction of Selection Bias in Traffic Data by Bayesian Network Data Fusion *JAIF*, **3**, 1 (July 2008), 50–62.

K

Kadar, I., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Kahler, B. and Blasch, E., Decision-Level Fusion Performance Improvement From Enhanced HRR Radar Clutter Suppression *JAIF*, **6**, 2 (December 2011), 101–118.
Kaliciak, L., Myrhaug, H., Goker, A., Song, D., Early Fusion and Query Modification in Their Dual Late Fusion Forms, *JAIF*, **10**, 2 (December 2015), 183–198.
Kaplan, L. M., *see* George, J., *JAIF*, **8**, 1 (July 2013), 15–32.
Kaplan, L., *see* Osborne, R. W., III, *JAIF*, **9**, 2 (December 2014), 75–89.
Karlsson, A., Johansson, R., Andler, S. F., Characterization and Empirical Evaluation of Bayesian and Credal Combination Operators, *JAIF*, **6**, 2 (December 2011), 150–166.
Khan, M. A., Ulmke, M., Koch, W., Analysis of Log-Homotopy Based Particle Flow Filters, *JAIF*, **12**, 1 (June 2017), 73–94.
Kirubarajan, T., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
Kirubarajan, T., *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.
Kleinman, D. L., *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.
Koch, W., *see* Degen, C., *JAIF*, **10**, 2 (December 2015), 125–142.
Koch, W., *see* Khan, M. A., *JAIF*, **12**, 1 (June 2017), 73–94.
Kokar, M. M., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.

Kokkala, J., Solin, A., Särkkä, S., Sigma-Point Filtering and Smoothing Based Parameter Estimation in Nonlinear Dynamic Systems, *JAIF*, **11**, 1 (June 2016), 15–30.
Kroschel, K., *see* Schlosser, M. S., *JAIF*, **2**, 2 (Dec. 2007), 65–76.
Kruger, M., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.
Kurz, G., Gilitschenski, I., Julier, S., Hanebeck, U. D., Recursive Bingham Filter for Directional Estimation Involving 180 Degree Symmetry, *JAIF*, **9**, 2 (December 2014), 90–105.
Kurz, G., Gilitschenski, I., Siegwart, R. Y., Hanebeck, U. D., Methods for Deterministic Approximation of Circular Densities, *JAIF*, **11**, 2 (December 2016), 138–156.
Kurz, G. and Hanebeck, U. D., Dynamic Surface Reconstruction by Recursive Fusion of Depth and Position Measurements *JAIF*, **9**, 1 (July 2014), 13–26.
Kurz, G., *see* Pfaff, F., *JAIF*, **11**, 2 (December 2016), 206–226.
Kwan, C., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.

L

Le Bihan, N., Flamant, J., Manton, J. H., Density Estimation on the Rotation Group Using Diffusive Wavelets, *JAIF*, **11**, 2 (December 2016), 173–185.
La Cour, B. R., *see* Aughenbaugh, J. M., *JAIF*, **5**, 2 (Dec. 2010), 108–127.
Lang, T., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
Lecornu, L., *see* Alsun, M., *JAIF*, **7**, 2 (December 2012), 101–113.
Levchuck, G., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.
Levedahl, M., *see* Areta, J., *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Li, X. R., *see* Jilkov, V. P., *JAIF*, **4**, 1 (July 2009), 27–39.
Lintz, C., *see* Pasupuleti, D., *JAIF*, **12**, 1 (June 2017), 41–57.
Luginbuhl, T., *see* Bordonaro, S., *JAIF*, **12**, 2 (December 2017), 228–242.

M

Mannaru, P., *see* Pasupuleti, D., *JAIF*, **12**, 1 (June 2017), 41–57.
Manton, J. H., *see* Le Bihan, N., *JAIF*, **11**, 2 (December 2016), 173–185.
Manton, J. H., *see* Said, S., *JAIF*, **11**, 2 (December 2016), 227–249.
Marano, S., *see* Georgescu, R., *JAIF*, **7**, 1 (June 2012), 61–77.
Marković, I., Bukal, M., Ćesić, J., Petrović, I., Multitarget Tracking with the von Mises-Fisher Filter and Probabilistic Data Association, *JAIF*, **11**, 2 (December 2016), 157–172.
Martin, A., Osswald, C., Dezert, J., Smarandache, F., General Combination Rules for Qualitative and Quantitative Beliefs, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
Matta, V., *see* Georgescu, R., *JAIF*, **7**, 1 (June 2012), 61–77.
Miceli, P. A., *see* Blair, W. D., *JAIF*, **7**, 1 (June 2012), 28–45.
Milgram, M., *see* Belaroussi, R., *JAIF*, **1**, 1 (July 2006), 35–51.
Mori, S., Chang, K. C., Takahashi, H., Chong, C-Y., Applying Interacting Multiple Model to Financial Asset Allocation, *JAIF*, **12**, 1 (June 2017), 95–109.
Moyer, S., *see* Howell, C., *JAIF*, **8**, 2 (December 2013), 133–142.
Mullen, T., *see* Avasarala, V., *JAIF*, **4**, 1 (July 2009), 52–71.
Myrhaug, H., *see* Kaliciak, L., *JAIF*, **10**, 2 (December 2015), 183–198.

N

Nagi, R., *see* Sambhoos, K., *JAIF*, **3**, 2 (Dec. 2008), 90–106.
Nadarajah, N., *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.
Natale, A., *see* Vivone, G., *JAIF*, **12**, 2 (December 2017), 189–210.
Ng, G. W., *see* Foo, P. H., *JAIF*, **4**, 1 (July 2009), 3–26.
Ng, G. W., *see* Foo, P. H., *JAIF*, **8**, 1 (July 2013), 33–72.
Ng, G. W., *see* Huang, H. A. J., *JAIF*, **12**, 1 (June 2017), 110–124.
Ng, G. W., *see* Yang, R., *JAIF*, **12**, 1 (June 2017), 3–19.
Ng, K. H., *see* Foo, P. H., *JAIF*, **4**, 1 (July 2009), 3–26.
Nitzan, E., Routtenberg, T., Tabrikian, J., Stochastic Filtering Using Periodic Cost Functions, *JAIF*, **11**, 2 (December 2016), 123–137.

O

Oispuu, M., *see* Steffes, C., *JAIF*, **11**, 2 (December 2016), 250–261.
Osborne, R. W., III, Bar-Shalom, Y., George, J., Kaplan, L., Statistical Efficiency of Target Localization from Angle and Shockwave Measurements, *JAIF*, **9**, 2 (December 2014), 75–89.
Osborne, R. W., III, Bar-Shalom, Y., Willett, P., Fusion of Asynchronous Passive Measurements, *JAIF*, **10**, 2 (December 2015), 199–210.
Osborne, R. W., III, Bar-Shalom, Y., Willett, P., Track-to-Track Association with Augmented State, *JAIF*, **7**, 1 (June 2012), 3–15.
Osborne, R. W., III, *see* Belfadel, D., *JAIF*, **9**, 2 (December 2014), 59–74.
Osborne, R. W., III, *see* Belfadel, D., *JAIF*, **10**, 2 (December 2015), 101–112.
Osborne, III, R. W., *see* Belfadel, D., *JAIF*, **12**, 1 (June 2017), 58–72.
Osborne, R. W., III, *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.
Osswald, C., *see* Martin, A., *JAIF*, **3**, 2 (Dec. 2008), 67–89.

P

Panakkal, V. P. and Velmurugan, R., An improved measurement model for target tracking under measurement origin uncertainty *JAIF*, **10**, 2 (December 2015), 143–162.
Pander, M., *see* Steinbring, J., *JAIF*, **11**, 1 (June 2016), 71–90.
Papageorgiou, D. J. and Holender, M., Track-to-Track Association and Ambiguity Management in the Presence of Sensor Bias *JAIF*, **6**, 2 (December 2011), 77–100.
Paradis, S., *see* Benaskeur, A. R., *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Pasupuleti, D., Mannaru, P., Balasingam, B., Baum, M., Pattipati, K., Willett, P., Lintz, C., Commeau, G., Dorigo, F., Fahrny, J., Cognitive Video Streaming, *JAIF*, **12**, 1 (June 2017), 41–57.
Pattipati, K. R., *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.
Pattipati, K. R., *see* Areta, J., *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Pattipati, K. R., *see* Areta, J., *JAIF*, **4**, 1 (July 2009), 40–51.
Pattipati, K., *see* Belfadel, D., *JAIF*, **12**, 1 (June 2017), 58–72.
Pattipati, K., *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.
Pattipati, K., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.
Pattipati, K., *see* Pasupuleti, D., *JAIF*, **12**, 1 (June 2017), 41–57.
Pattipati, K. R., *see* Tian, X., *JAIF*, **5**, 1 (July 2010), 3–17.
Peel, L., Estimating Network Parameters for Selecting Community Detection Algorithms *JAIF*, **6**, 2 (December 2011), 119–130.
Petrović, I., *see* Marković, I., *JAIF*, **11**, 2 (December 2016), 157–172.
Pfaff, F., Kurz, G., Hanebeck, U. D., Multivariate Angular Filtering Using Fourier Series, *JAIF*, **11**, 2 (December 2016), 206–226.
Piché, R., *see* Raitoharju, M., *JAIF*, **11**, 1 (June 2016), 3–14.
Pignol, A-C., *see* Jauffret, C., *JAIF*, **6**, 1 (June 2011), 24–38.
Pillon, D., *see* Jauffret, C., *JAIF*, **6**, 1 (June 2011), 24–38.
Powell, G. M., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Prevost, L., *see* Belaroussi, R., *JAIF*, **1**, 1 (July 2006), 35–51.
Punithakumar, K., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.

R

Raitoharju, M., Piché, R., Ala-Luhtala, J., Ali-Löytty, S., Partitioned Update Kalman Filter, *JAIF*, **11**, 1 (June 2016), 3–14.
Ravindra, V. C., Bar-Shalom, Y., and Damarla, T., Feature-Aided Tracking of Ground Vehicles using Passive Acoustic Sensor Arrays, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Rhéaume, F., *see* Benaskeur, A. R., *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Reindl, L., *see* Hamp, Q., *JAIF*, **7**, 2 (December 2012), 131–140.
Reuter, J., *see* Schuster, M., *JAIF*, **12**, 2 (December 2017), 175–188.
Reuter, S., *see* Granström, K., *JAIF*, **12**, 2 (December 2017), 139–174.
Rickard, T., *see* Sambhoos, K., *JAIF*, **3**, 2 (Dec. 2008), 90–106.
Rickard, T., *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.
Rodningsby, A., Bar-Shalom, Y., Hallingstad, O., and Glattetre, J., Multitarget Multisensor Tracking in the Presence of Wakes, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Rogova, G. L., Bursik, M. I., and Hanson-Hedgecock, S., Intelligent System for Interpreting the Pattern of Volcanic Eruptions, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
Romeo, K., Bar-Shalom, Y., Willett, P., Fusion of Multipath Data with ML-PMHT for Very Low SNR Track Detection in an OTHR, *JAIF*, **10**, 2 (December 2015), 113–124.
Roth, M., Hendeby, G., Gustafsson, F., Nonlinear Kalman Filters Explained: A Tutorial on Moment Computations and Sigma Point Methods, *JAIF*, **11**, 1 (June 2016), 47–70.
Rothrock, R., *see* Areta, J., *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Routtenberg, T., *see* Eyal Nitzan, E., *JAIF*, **11**, 2 (December 2016), 123–137.
Ruspini, E. H., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.

S

Said, S. and Manton, J. H., Particle Filtering with Observations in a Manifold: A Proof of Convergence and Two Worked Examples *JAIF*, **11**, 2 (December 2016), 227–249.
Salerno, E., Adurthi, N., Singh, T., Singla, P., Bubalo, A., Cornacchia, M., Alford, M., Jones, E., Road Network Identification by means of the Hough Transform with Uncertainty Analysis, *JAIF*, **10**, 1 (June 2015), 58–72.
Salerno, J., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Särkkä, S., Hartikainen, J., Svensson, L., Sandblom, F., On the relation between Gaussian process quadratures and sigma-point methods, *JAIF*, **11**, 1 (June 2016), 31–46.
Särkkä, S., *see* Kokkala, J., *JAIF*, **11**, 1 (June 2016), 15–30.

Sambhoos, K., Nagi, R., Sudit, M., and Rickard, T., Hierarchical Higher Level Data Fusion using Fuzzy Hamming and Hypercube Clustering, *JAIF*, **3**, 2 (Dec. 2008), 90–106.

Sandblom, F., *see* Särkkä, S., *JAIF*, **11**, 1 (June 2016), 31–46.

Schaub, D. E., Joint Identification of Multiple Tracked Targets *JAIF*, **12**, 1 (June 2017), 20–40.

Schlosser, M. S. and Kroschel, K., Performance Analysis of Decentralized Kalman Filters under Communication Constraints *JAIF*, **2**, 2 (Dec. 2007), 65–76.

Schneider, M. K., *see* Washburn, R. B., *JAIF*, **3**, 1 (July 2008), 3–13.

Schrag, R. C., Takikawa, M., Goger, P., and Eilbert, J., Performance Evaluation for Automated Threat Detection, *JAIF*, **2**, 2 (Dec. 2007), 77–98.

Schuster, M., Reuter, J., Wanielik, G., Multi Detection Joint Integrated Probabilistic Data Association Using Random Matrices with Applications to Radar-Based Multi Object Tracking, *JAIF*, **12**, 2 (December 2017), 175–188.

Scott, P. D., *see* Terejanu, G., *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Shen, D., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.

Siegwart, R. Y., *see* Kurz, G., *JAIF*, **11**, 2 (December 2016), 138–156.

Šimandl, M., *see* Duník, J., *JAIF*, **11**, 1 (June 2016), 91–109.

Singh, S., *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.

Singh, T., *see* George, J., *JAIF*, **6**, 1 (June 2011), 39–56.

Singh, T., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.

Singh, T., *see* Terejanu, G., *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Singla, P., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.

Singla, P., *see* Terejanu, G., *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Smarandache, F., *see* Martin, A., *JAIF*, **3**, 2 (Dec. 2008), 67–89.

Snidaro, L., *see* Biermann, J., *JAIF*, **8**, 2 (December 2013), 101–118.

Solaiman, B., *see* Alsun, M., *JAIF*, **7**, 2 (December 2012), 101–113.

Solin, A., *see* Kokkala, J., *JAIF*, **11**, 1 (June 2016), 15–30.

Song, D., *see* Kaliciak, L., *JAIF*, **10**, 2 (December 2015), 183–198.

Song, X., Willett, P., Zhou, S., Bearings-Only Localization with NLOS Reflected AoAs, *JAIF*, **8**, 1 (July 2013), 3–14.

Song, X., Willett, P., Zhou, S., Posterior Cramér-Rao Bounds for Doppler Biased Distributed Tracking, *JAIF*, **7**, 1 (June 2012), 16–27.

Steinbring, J. and Hanebeck, U. D., LRKF Revisited: The Smart Sampling Kalman Filter (S^2KF) *JAIF*, **9**, 2 (December 2014), 106–123.

Steffes, C. and Oispuu, M., Direct Position Determination for TDOA-based Single Sensor Localization *JAIF*, **11**, 2 (December 2016), 250–261.

Steinbring, J., Pander, M., Hanebeck, U. D., The Smart Sampling Kalman Filter with Symmetric Samples, *JAIF*, **11**, 1 (June 2016), 71–90.

Stone, L. D. and Anderson, S. L., Standard Bayesian Approach to Quantized Measurements and Imprecise Likelihoods *JAIF*, **10**, 1 (June 2015), 3–12.

Stotz, A., *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.

Straka, O., *see* Duník, J., *JAIF*, **11**, 1 (June 2016), 91–109.

Streit, R., A Technique for Deriving Multitarget Intensity Filters Using Ordinary Derivatives *JAIF*, **9**, 1 (July 2014), 3–12.

Streit, R., The Probability Generating Functional for Finite Point Processes, and Its Application to the Comparison of PHD and Intensity Filters *JAIF*, **8**, 2 (December 2013), 119–132.

Sudit, M., Holender, M., Stotz, A., Rickard, T., and Yager, R., INFERD and Entropy for Situational Awareness, *JAIF*, **2**, 1 (July 2007), 3–21.

Sudit, M., *see* Sambhoos, K., *JAIF*, **3**, 2 (Dec. 2008), 90–106.

Svensson, L., *see* Särkkä, S., *JAIF*, **11**, 1 (June 2016), 31–46.

T

Tabrikian, J., *see* Eyal Nitzan, E., *JAIF*, **11**, 2 (December 2016), 123–137.

Takahashi, H., *see* Mori, S., *JAIF*, **12**, 1 (June 2017), 95–109.

Takikawa, M., *see* Schrag, R. C., *JAIF*, **2**, 2 (Dec. 2007), 77–98.

Terejanu, G., Singla, P., Singh, T., and Scott, P. D., A Decision-Centric Framework for Density Forecasting, *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Tharmarasa, R., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.

Tharmarasa, R., Kirubarajan, T., Nadarajah, N., Bar-Shalom, Y., Thayaparan, T., Profile-Free Launch Point Estimation for Ballistic Targets using Passive Sensors, *JAIF*, **7**, 1 (June 2012), 46–60.

Thayaparan, T., *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.

Tian, X., Bar-Shalom, Y., and Pattipati, K. R., Multi-step Look-Ahead Policy for Autonomous Cooperative Surveillance by UAVs in Hostile Environments, *JAIF*, **5**, 1 (July 2010), 3–17.

Tian, X. and Bar-Shalom, Y., Algorithms for Asynchronous Track-to-Track Fusion *JAIF*, **5**, 2 (Dec. 2010), 128–138.

Tian, X., *see* Yuan, T., *JAIF*, **6**, 2 (December 2011), 131–149.

Tian, X. and Bar-Shalom, Y., Track-to-Track Fusion Configurations and Association in a Sliding Window *JAIF*, **4**, 2 (Dec. 2009), 146–164.

Tu, F., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.

U

Ulmke, M., *see* Khan, M. A., *JAIF*, **12**, 1 (June 2017), 73–94.

V

Valin, P., Djiknavorian, P., and Bossé, É., A Pragmatic Approach for the use of Dempster-Shafer Theory in Fusing Realistic Sensor Data, *JAIF*, **5**, 1 (July 2010), 32–40.

Velmurugan, R., *see* Panakkal, V. P., *JAIF*, **10**, 2 (December 2015), 143–162.

Vivone, G., Braca, P., Granström, K., Natale, A., Chanussot, J., Converted Measurements Bayesian Extended Target Tracking Applied to X-band Marine Radar Data, *JAIF*, **12**, 2 (December 2017), 189–210.

W

Wanielik, G., *see* Schuster, M., *JAIF*, **12**, 2 (December 2017), 175–188.

Washburn, R. B. and Schneider, M. K., Optimal Policies for a Class of Restless Multiarmed Bandit Scheduling Problems with Applications to Sensor Management *JAIF*, **3**, 1 (July 2008), 3–13.

Watson, G., *see* Zhang, S., *JAIF*, **6**, 1 (June 2011), 3–23.

Willett, P., *see* Bordonaro, S., *JAIF*, **12**, 2 (December 2017), 228–242.

Willett, P., *see* Choi, S., *JAIF*, **8**, 2 (December 2013), 143–155.

Willett, P., *see* Choi, S., *JAIF*, **9**, 1 (July 2014), 27–37.

Willett, P., *see* Coraluppi, S., *JAIF*, **5**, 1 (July 2010), 18–31.

Willett, P., *see* Crouse, D. F., *JAIF*, **4**, 2 (Dec. 2009), 93–116.

Willett, P., *see* Crouse, D. F., *JAIF*, **7**, 1 (June 2012), 97–98.

Willett, P., *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.

Willett, P., *see* Dou, W., *JAIF*, **10**, 2 (December 2015), 163–182.

Willett, P., *see* Erdinc, O., *JAIF*, **2**, 1 (July 2007), 22–34.

Willett, P., *see* Georgescu, R., *JAIF*, **7**, 1 (June 2012), 61–77.

Willett, P., *see* Osborne, R. W., III, *JAIF*, **7**, 1 (June 2012), 3–15.

Willett, P., *see* Osborne, R. W., III, *JAIF*, **10**, 2 (December 2015), 199–210.

Willett, P., *see* Pasupuleti, D., *JAIF*, **12**, 1 (June 2017), 41–57.

Willett, P., *see* Romeo, K., *JAIF*, **10**, 2 (December 2015), 113–124.

Willett, P., *see* Song, X., *JAIF*, **7**, 1 (June 2012), 16–27.

Willett, P., *see* Song, X., *JAIF*, **8**, 1 (July 2013), 3–14.

Wu, J., *see* Jilkov, V. P., *JAIF*, **10**, 1 (June 2015), 73–88.

Wyffels, K. and Hanebeck, U. D., Priority-Based Tracking of Extended Objects *JAIF*, **12**, 2 (December 2017), 211–227.

Y

Yager, R., *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.

Yang, C. and Blasch, E., Fusion of Tracks with Road Constraints *JAIF*, **3**, 1 (July 2008), 14–32.

Yang, R., Bar-Shalom, Y., Ng, G. W., Bearings-Only Tracking with Fusion from Heterogenous Passive Sensors: ESM/EO and Acoustic, *JAIF*, **12**, 1 (June 2017), 3–19.

Yang, R., *see* Foo, P. H., *JAIF*, **4**, 1 (July 2009), 3–26.

Yang, R., *see* Huang, H. A. J., *JAIF*, **12**, 1 (June 2017), 110–124.

Yu, F., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.

Yuan, T., Bar-Shalom, Y., Tian, X., Heterogeneous Track-to-Track Fusion, *JAIF*, **6**, 2 (December 2011), 131–149.

Z

Zea, A., *see* Faion, F., *JAIF*, **10**, 1 (June 2015), 13–30.

Zhang, S. and Bar-Shalom, Y., Practical Data Association for Passive Sensors in 3D *JAIF*, **9**, 1 (July 2014), 38–46.

Zhang, S., Bar-Shalom, Y., Watson, G., Tracking with Multisensor Out-of-Sequence Measurements with Residual Biases, *JAIF*, **6**, 1 (June 2011), 3–23.

Zheng, L., *see* Crouse, D. F., *JAIF*, **7**, 1 (June 2012), 97–98.

Zheng, Y. and Blasch, E., An Exploration of the Impacts of Three Factors in Multimodal Biometric Score Fusion: Score Modality, Recognition Method, and Fusion Process *JAIF*, **9**, 2 (December 2014), 124–135.

Zhou, S., *see* Choi, S., *JAIF*, **9**, 1 (July 2014), 27–37.

Zhou, S., *see* Song, X., *JAIF*, **7**, 1 (June 2012), 16–27.

Zhou, S., *see* Song, X., *JAIF*, **8**, 1 (July 2013), 3–14.

SUBJECT INDEX

0–1 nonlinear optimization
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.

A

acoustic localization
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.
adaptive encoding
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.
Adaptive Gaussian Mixture Models
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
adaptive model selection
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
air defense
Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.
alerting
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
algorithm selection
Peel, L., *JAIF*, **6**, 2 (December 2011), 119–130.
ambiguity management
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.
angular quantities
Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.
anomaly detection
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
approximate crosscorrelation
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Asset Allocation
Mori, S., +, *JAIF*, **12**, 1 (June 2017), 95–109.
assignment
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Association ambiguity
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.
association benchmark
Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
association
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.
Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.
asynchronous
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
Asynchronous Track-to-Track fusion
Tian, X., +, *JAIF*, **5**, 2 (Dec. 2010), 128–138.
augmented state
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.
automatic target recognition (ATR)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
Autonomous Vehicles
Wyffels, K., +, *JAIF*, **12**, 2 (December 2017), 211–227.

B

ballistic targets
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
Bayesian
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Bayesian and Reasoning Methods
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
Valin, P., +, *JAIF*, **5**, 1 (July 2010), 32–40.
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.
Bayesian Data Fusion
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
Bayesian filtering
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
Bayesian filters
Duník, J., +, *JAIF*, **11**, 1 (June 2016), 91–109.
Bayesian Game
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Bayesian Networks (BNs)
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
Bayesian network
Chang, K. C., +, *JAIF*, **1**, 2 (Dec. 2006), 95–107.
Bayesian theory
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
Bayesian tracking
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
bearings-only target motion analysis
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
benchmark performance
Howell, C., +, *JAIF*, **8**, 2 (December 2013), 133–142.
belief function theory
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
bearings-only tracking
Huang, H. A. J., +, *JAIF*, **12**, 1 (June 2017), 110–124.
bias
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Bias estimation
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Belfadel, D., +, *JAIF*, **12**, 1 (June 2017), 58–72.
bias observability
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
biased measurement
Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.
bias-variance trade-off
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
BOMTMA
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
BOTMA
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
brake-by-wire
Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.

C

capacities
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
cardinality tracking
Coraluppi, S., +, *JAIF*, **7**, 2 (December 2012), 153–164.
centralized fusion
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Characteristic function
Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.
circular data
Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.
circular statistics
Crouse, D. F., +, *JAIF*, **8**, 1 (July 2013), 73–89.
Kurz, G., +, *JAIF*, **11**, 1 (June 2016), 138–156.
Circular stochastic filtering
Nitzan, E., +, *JAIF*, **11**, 1 (June 2016), 123–137.
classification
Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.
Classification, Learning, Data Mining
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
clustering
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
clutter suppression
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
collation
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
Combat Power Management
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
combination
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
combination operators
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
combinatorial auctions
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
Command and Control
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
communication rate reduction
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
Community detection
Peel, L., *JAIF*, **6**, 2 (December 2011), 119–130.

composite measurement
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
composite measurements
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Belfadel, D., +, *JAIF*, **12**, 1 (June 2017), 58–72.
Con-Tracker
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
connector map
Said, S., +, *JAIF*, **11**, 1 (June 2016), 227–249.
conflict management
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
conservative fusion approach
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
Content-based Image Retrieval
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
context-aware modeling
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Continuous time systems
Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.
convergent analysis
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
coordinated turn
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
Coupled model
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
Cramér-Rao lower bound
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
credal sets
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
CRLB
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Belfadel, D., +, *JAIF*, **12**, 1 (June 2017), 58–72.
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.

D

data association
Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.
data association uncertainty
Belfadel, D., +, *JAIF*, **12**, 1 (June 2017), 58–72.
data fusion
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
Foo, P. H., +, *JAIF*, **8**, 1 (July 2013), 33–72.
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.
Data Qualification
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
Decentralized Kalman Filter (DKF)
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
decision making
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
Decision-Level Fusion (DLF)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
Dempster-Shafer
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
Valin, P., +, *JAIF*, **5**, 1 (July 2010), 32–40.
Density Forecasting
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
depth camera
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
descision fusion
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
descriptor system
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
detection
Davey, S. J., *JAIF*, **6**, 2 (December 2011), 167–186.
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
deterministic sampling
Kurz, G., +, *JAIF*, **11**, 1 (June 2016), 138–156.
DHF
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
differentiable manifold
Said, S., +, *JAIF*, **11**, 1 (June 2016), 227–249.
Differentiation
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
diffusion process
Said, S., +, *JAIF*, **11**, 1 (June 2016), 227–249.
Diffusive wavelets
Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.
Digital audio/video broadcasting
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.
Dirac Mixtures
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
Kurz, G., +, *JAIF*, **11**, 1 (June 2016), 138–156.
direction of arrival (DoA) estimation
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
direction-of-arrival (DOA) tracking
Nitzan, E., +, *JAIF*, **11**, 1 (June 2016), 123–137.
directional statistics
Crouse, D. F., +, *JAIF*, **8**, 1 (July 2013), 73–89.
Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.
Pfaff, F., +, *JAIF*, **11**, 1 (June 2016), 206–226.
discounting operators
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
Displaced Phase Center Antenna (DPCA)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
distributed detection
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.
distributed inference
Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.
distributed tracking
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.
Divided Differences
Michael Roth, +, *JAIF*, **11**, 1 (June 2016), 47–70.
DS-structures
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
DSmT
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
dynamic programming
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.
Dynamic State Estimation
Mori, S., +, *JAIF*, **12**, 1 (June 2017), 95–109.

E

early fusion
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
EM Algorithm
Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
entropy
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
estimator efficiency
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.
Expected Loss
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
experiment design
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
experimental results
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
Extended Objects
Wyffels, K., +, *JAIF*, **12**, 2 (December 2017), 211–227.
Extended Object Tracking
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.

F

false measurements
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
fault diagnosis
Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.
features
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
feature extraction
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
feature-aided tracking
Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.
FIM
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.

Finite point processes, Multitarget tracking
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
Fire Control
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
flight profile
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
Fourier series
Pfaff, F., +, *JAIF*, **11**, 1 (June 2016), 206–226.
Fourier-based stochastic filtering via root-finding (FB-SFRF)
Nitzan, E., +, *JAIF*, **11**, 1 (June 2016), 123–137.
fuse-before-track
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.
fusion
Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.
Fusion Applications
Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
Fusion Architectures and Management Issues
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Chang, K. C., +, *JAIF*, **1**, 2 (Dec. 2006), 95–107.
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
Tian, X., +, *JAIF*, **5**, 1 (July 2010), 3–17.
fusion of clustering results
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
fuzzy
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
fuzzy hamming distance
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
fuzzy inference
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
fuzzy logic
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
fuzzy rule-base
Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.

G

game theory
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
Gauss-Newton method
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
genetic algorithms
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Geodesy
Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.
geological data
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
global
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Graph Matching
Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.
grid-based filtering
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
GMTI
Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.
gunfire detection system
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.

H

heterogeneous track-to-track fusion
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
hidden Markov model
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
hidden Markov model (HMM)
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
hierarchical entity aggregation
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
High Level Data Fusion
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
High Range Resolution (HRR)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
histogram PMHT
Davey, S. J., *JAIF*, **6**, 2 (December 2011), 167–186.
Hough transform
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.
human cognition
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
human factors
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.
Human-Fusion Interaction
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
human identification
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
Human-Machine Interface
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
hybrid relevance feedback
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
Hybrid Systems
Wyffels, K., +, *JAIF*, **12**, 2 (December 2017), 211–227.
hypercube distance
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
hypotheses
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
hypothesis generator
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
hypothesis management
Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.

I

Image Fusion
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
image fusion performance assessment
Howell, C., +, *JAIF*, **8**, 2 (December 2013), 133–142.
image fusion task performance
Howell, C., +, *JAIF*, **8**, 2 (December 2013), 133–142.
imetrics
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
imprecise
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
imprecise probability
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
index rules
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.
indicator
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
INFERD
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
Influence Diagram
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Information and data fusion
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
information extraction
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
information fusion
Bar-Shalom, Y., +, *JAIF*, **1**, 1 (July 2006), 3–14.
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Foo, P. H., +, *JAIF*, **8**, 1 (July 2013), 33–72.
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
information gain heuristic
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.

information heuristics
Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.
Information Retrieval
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
Integration Rules
Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.
intelligence
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
intent inference
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
intensity filter
Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.
Interacting Multiple Model (IMM) Tracking
Mori, S., +, *JAIF*, **12**, 1 (June 2017), 95–109.
interacting multiple models
Huang, H. A. J., +, *JAIF*, **12**, 1 (June 2017), 110–124.
interaction networks
Peel, L., *JAIF*, **6**, 2 (December 2011), 119–130.
internet video
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.
Interpolation
Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.
IPDA
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
Iterative solution
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.

K

Kalman filter
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.
Kernel estimators
Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.
Kernel density estimation
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
Knowledge Representation
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.

L

L2/L3 fusion
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Lagrangian multiplier
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
late fusion
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
launch point estimation
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
LCD
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
least squares
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
least squares estimation
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
legacy sensors
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.
level 2 fusion
Chang, K. C., +, *JAIF*, **1**, 2 (Dec. 2006), 95–107.
level 4 data fusion
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
likelihood
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
likelihood sampling
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
linear minimum mean square error
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
link-analysis
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
localization
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
Log-homotopy
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
LOS
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
LRKF
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.

M

m-Best Soft Assignment Algorithm
Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.
machine learning
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.
Mahler/Fixs?n rule
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
Maneuvering target
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
Markov Jump Linear Systems
Mori, S., +, *JAIF*, **12**, 1 (June 2017), 95–109.
maximum likelihood
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *JAIF*, **12**, 1 (June 2017), 58–72.
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
maximum likelihood estimate
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
maximum likelihood estimation
Crouse, D. F., +, *JAIF*, **8**, 1 (July 2013), 73–89.
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
maximum likelihood fusion
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
Mean Integrated Square Error (MISE)
Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.
mean opinion score (MOS)
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.
measurement fusion
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
measurement origin uncertainty
Panakkal, V. P., +, *JAIF*, **10**, 2 (December 2015), 143–162.
Medical Decision Support Systems
Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.
Medical Reasoning
Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.
MHT
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
military application
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
MIMO radar
Song, X., +, *JAIF*, **7**, 1 (June 2012), 16–27.
minimum variance distortionless response (MVDR)
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
misassociation
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Mixture of densities
Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.
ML-PMHT
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
model-based classifier
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
modelling
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
Moment Computation Problem
Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.
moment matching
Kurz, G., +, *JAIF*, **11**, 1 (June 2016), 138–156.
moving target identification
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
multi-channel signal subspace (MSS)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
multi-criteria classification
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
multi-hypothesis tracking
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.
Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.
multi-sensor multi-target tracking
Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.
multi-sensor target tracking
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
multitarget tracking
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
Marković, I., +, *JAIF*, **11**, 1 (June 2016), 157–172.

multidimensional assignment
Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Multimedia Retrieval
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
Multimodal biometric score fusion
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
multiple dimension assignment (MDA)
Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.
multiobjective optimization
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
multipath
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
multiple-hypothesis tracking
Coraluppi, S., +, *JAIF*, **7**, 2 (December 2012), 153–164.
Multiple-Model Adaptive Estimator
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
multisensor tracking
Bar-Shalom, Y., +, *JAIF*, **1**, 1 (July 2006), 3–14.
Blair, W. D., +, *JAIF*, **7**, 1 (June 2012), 28–45.
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
multispectral face recognition
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
multistatic
Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.
multitarget multisensor
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Multitarget Multisensor Tracking
Georgescu, R., +, *JAIF*, **7**, 1 (June 2012), 61–77.
multitarget tracking filters
Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.
mutli-sensor tracking
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.
multivariate filtering
Pfaff, F., +, *JAIF*, **11**, 1 (June 2016), 206–226.

N

Naval Anti-Air Warfare
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
nearest neighbor
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
nearest neighbor classification
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.
neural networks
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.
Non-Gaussian noise
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
Nonlinear filtering
Duník, J., +, *JAIF*, **11**, 1 (June 2016), 91–109.
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
Kurz, G., +, *JAIF*, **11**, 1 (June 2016), 138–156.
Nonlinear Kalman Filtering
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
non-uniform quantization
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.
Numerical Derivatives
Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.

O

Object Tracking
Wyffels, K., +, *JAIF*, **12**, 2 (December 2017), 211–227.
observability
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Belfadel, D., +, *JAIF*, **12**, 1 (June 2017), 58–72.
OFDM modulation
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.
Organizational Structures Identification
Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.
OTHR
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
out-of-sequence measurement
Huang, H. A. J., +, *JAIF*, **12**, 1 (June 2017), 110–124.
Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.

P

parallel and distributed computing, GPU
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
particle filter
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
particle filtering
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
Said, S., +, *JAIF*, **11**, 1 (June 2016), 227–249.
particle flow filter
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
Particle flow filters
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
passive acoustic sensor network
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Passive radar
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.
passive sensor
Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.
PCRLB, LFM
Song, X., +, *JAIF*, **7**, 1 (June 2012), 16–27.
PDA
Panakkal, V. P., +, *JAIF*, **10**, 2 (December 2015), 143–162.
performance analysis
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
performance evaluation
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
performance model
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
Performance Prediction
Blair, W. D., +, *JAIF*, **7**, 1 (June 2012), 28–45.
periodic risks
Nitzan, E., +, *JAIF*, **11**, 1 (June 2016), 123–137.
PMHT
Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
point cloud
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
point processes
Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.
Possibilistic Reasoning
Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.
posterior Cramer-Rao lower bound
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.
predetection Fusion
Georgescu, R., +, *JAIF*, **7**, 1 (June 2012), 61–77.
probabilistic data association
Marković, I., +, *JAIF*, **11**, 1 (June 2016), 157–172.
Probabilistic Inference
Wyffels, K., +, *JAIF*, **12**, 2 (December 2017), 211–227.
Probabilistic Multi-Hypothesis Tracker (PMHT)
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.
probability
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Probability density estimation
Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.
Probability generating function
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
Probability generating functional
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
probability hypothesis density
Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.
Probability hypothesis density filter
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.
profile-free method
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
Propagation delay
Huang, H. A. J., +, *JAIF*, **12**, 1 (June 2017), 110–124.
proportional conflict redistribution rules
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.

Q

Quadratic Assignment Problem
Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.
qualitative beliefs
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.

quality of experience (QoE)
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.
quantized
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.

R

radar
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
random sets
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
Real-time Fusion
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
realistic sensor data
Valin, P., +, *JAIF*, **5**, 1 (July 2010), 32–40.
reasoning
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
recursive Bayesian estimation
Pfaff, F., +, *JAIF*, **11**, 1 (June 2016), 206–226.
recursive filtering
Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.
Redrawing
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
reflection
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
resource allocation
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
resource management
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.
restless bandits
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.
risk
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
road map
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
road network extraction
Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.
Robot Perception
Wyffels, K., +, *JAIF*, **12**, 2 (December 2017), 211–227.
rollout algorithm
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
Rotation group $SO(3)$
Le Bihan, N., +, *JAIF*, **11**, 1 (June 2016), 173–185.

S

S^2KF
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
***S*-D algorithm**
Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.
sample-based stochastic filtering via root-finding (SB-SFRF)
Nitzan, E., +, *JAIF*, **11**, 1 (June 2016), 123–137.
Sampling
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
Schmidt-Kalman filter
Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.
score fusion evaluation metric
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
Secular function
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
Selection Bias
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
self-fusion
Howell, C., +, *JAIF*, **8**, 2 (December 2013), 133–142.
sensor assignment
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
sensor fusion
Blair, W. D., +, *JAIF*, **7**, 1 (June 2012), 28–45.
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
Sensor Layout
Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.
sensor localization
Crouse, D. F., +, *JAIF*, **8**, 1 (July 2013), 73–89.
sensor management
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.
Sensor network
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
Sensor Networks
Georgescu, R., +, *JAIF*, **7**, 1 (June 2012), 61–77.
sensor resource management
Chang, K. C., +, *JAIF*, **1**, 2 (Dec. 2006), 95–107.
sensor scheduling
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
Shooter localization
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
Shrinkage estimators
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
Sigma-Points
Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.
Similarity Estimation
Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.
similarity index
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Simulation
Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.
situation assessment
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
Situation Awareness
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
situational assessment
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
situational awareness
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
smoothing
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
social networks
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
sonar
Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.
space-time adaptive processing (STAP)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
space tracking
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Belfadel, D., +, *JAIF*, **12**, 1 (June 2017), 58–72.
spline
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
Stansfield estimator
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
state constraints
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
state estimation
Duník, J., +, *JAIF*, **11**, 1 (June 2016), 91–109.
statistical efficiency
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Belfadel, D., +, *JAIF*, **12**, 1 (June 2017), 58–72.
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
stereo camera
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
Stiff ODE
Khan, M. A., +, *JAIF*, **12**, 1 (June 2017), 73–94.
stochastic filtering
Said, S., +, *JAIF*, **11**, 1 (June 2016), 227–249.
stochastic systems
Duník, J., +, *JAIF*, **11**, 1 (June 2016), 91–109.
structure discovery
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
structured hypotheses
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
subgraph matching
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
submodular minimization
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.
sudden maneuvers
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
surface estimation
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
survival models
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.

surveillance
Tian, X., +, *JAIF*, **5**, 1 (July 2010), 3–17.
Switching Financial Market Modeling
Mori, S., +, *JAIF*, **12**, 1 (June 2017), 95–109.

T

target attributes
Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.
Target Engageability
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Target Existence Uncertainty
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
target tracking
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.
target-death problem
Coraluppi, S., +, *JAIF*, **7**, 2 (December 2012), 153–164.
TBM, Monte Carlo simulation
Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.
templates
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
test sequencing
Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.
textual representation
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
threat evaluation
Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.
threat prediction
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
total least squares
Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.
track association
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.
track detection
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
track fusion
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
track initialization
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
track-before-detect
Davey, S. J., *JAIF*, **6**, 2 (December 2011), 167–186.
Track-to-Track
Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.
Track-to-track association
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.
tracking
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
Bar-Shalom, Y., +, *JAIF*, **1**, 1 (July 2006), 3–14.
Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Song, X., +, *JAIF*, **7**, 1 (June 2012), 16–27.
Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.
Tian, X., +, *JAIF*, **5**, 2 (Dec. 2010), 128–138.
tracking and classification
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
tracking, optimisation
Duník, J., +, *JAIF*, **11**, 1 (June 2016), 91–109.
Traffic Surveillance
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
trafficability
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Transferable Belief Model
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.

U

UAV search
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
UAV
Tian, X., +, *JAIF*, **5**, 1 (July 2010), 3–17.
uncertainty
Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.
Uncertainty Quantification
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
unscented Gauss-Helmert filter
Huang, H. A. J., +, *JAIF*, **12**, 1 (June 2017), 110–124.
unscented transform
Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.
Unscented Transformation
Roth, M., +, *JAIF*, **11**, 1 (June 2016), 47–70.

V

video quality metrics
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.
Video quality of service (QoS)
Pasupuleti, D., +, *JAIF*, **12**, 1 (June 2017), 41–57.
visual representation
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
VLO
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
volcanic eruptions
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
von Mises-Fisher distribution
Marković, I., +, *JAIF*, **11**, 1 (June 2016), 157–172.
voting algorithms
Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.

W

wake
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
weapon allocation
Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

2017 BOARD OF DIRECTORS*

2013–2015	2014–2016	2015–2017
Jean Dezert	Sten F. Andler	Joachim Biermann
Gee-Wah Ng	Murat Efe	Frederik Gustafsson
Anne-Laure Jousselme	Lyudmila Mihaylova	Jesús García

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Jean Dezert, 2016	Stefano Coraluppi, 2010	Xiao-Rong Li, 2003
Darin Dunham, 2015	Elisa Shahbazian, 2009	Yaakov Bar-Shalom, 2002
Darin Dunham, 2014	Darko Musicki, 2008	Pramod Varshney, 2001
Wolfgang Koch, 2013	Erik Blasch, 2007	Yaakov Bar-Shalom, 2000
Roy Streit, 2012	Pierre Valin, 2006	Jim Llinas, 1999
Joachim Biermann, 2011	W. Dale Blair, 2005	Jim Llinas, 1998
	Chee Chong, 2004	

SOCIETY VISION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied information fusion technologies.

SOCIETY MISSION

Advocate

To advance the profession of fusion technologies, propose approaches for solving real-world problems, recognize emerging technologies, and foster the transfer of information.

Serve

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

Communicate

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

Educate

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

Integrate

Integrate ideas from various approaches for information fusion, and look for common threads and themes–look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

Disseminate

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors are encouraged to submit their manuscripts for peer review http://isif.org/journal.

Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at http://jaif.msubmit.net. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.

Journal of Advances in Information Fusion Volume 13 Number 1 June 2018 ISSN 1557-6418