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.

1557-6418/18/$17.00 © 2018 JAIF

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 3.3×10^{-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 matrix 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

$$\begin{aligned}\nabla_{\mathbf{x}} \ln p(\dot r \mid \mathbf{x}) &= \frac{1}{\sigma_{\dot r}^2}\left(\dot r + \frac{(\mathbf{F}_h\mathbf{x}-\mathbf{l}_1)'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)\\ &\quad \cdot\left(\frac{\mathbf{F}_h'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|} + \frac{(\mathbf{F}_h\mathbf{x}-\mathbf{l}_1)'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|^3}(\mathbf{F}_h'\mathbf{l}_1 - \mathbf{F}_h'\mathbf{F}_h\mathbf{x})\right).\end{aligned} \tag{35}$$

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

$$\begin{aligned}\ln p(f \mid \mathbf{x}, f_{\text{Tx}}) &= -\frac{1}{2}\ln(2\pi\sigma_f^2)\\ &\quad - \frac{1}{2\sigma_f^2}\left(f - \left(1 + \frac{1}{c}\left(\frac{\mathbf{F}_h\mathbf{x}-\mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)'\dot{\mathbf{l}}_1\right) f_{\text{Tx}}\right)^2\end{aligned} \tag{36}$$

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

$$\begin{aligned}&\nabla_{\mathbf{x}} \ln p(f \mid \mathbf{x}, f_{\text{Tx}}) =\\ &\quad \frac{f_{\text{Tx}}}{c\sigma_f^2}\left(f - \left(1 + \frac{1}{c}\left(\frac{\mathbf{F}_h\mathbf{x}-\mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)'\dot{\mathbf{l}}_1\right) f_{\text{Tx}}\right)\\ &\quad \cdot\left(\frac{\mathbf{F}_h'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|} + \frac{(\mathbf{F}_h\mathbf{x}-\mathbf{l}_1)'\dot{\mathbf{l}}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|^3}(\mathbf{F}_h'\mathbf{l}_1 - \mathbf{F}_h'\mathbf{F}_h\mathbf{x})\right)\end{aligned} \tag{37}$$

$$\begin{aligned}&\frac{\partial}{\partial f_{\text{Tx}}} \ln p(f \mid \mathbf{x}, f_{\text{Tx}}) =\\ &\quad \frac{1}{\sigma_f^2}\left(f - \left(1 + \frac{1}{c}\left(\frac{\mathbf{F}_h\mathbf{x}-\mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)'\dot{\mathbf{l}}_1\right) f_{\text{Tx}}\right)\\ &\quad \cdot\left(1 + \frac{1}{c}\left(\frac{\mathbf{F}_h\mathbf{x}-\mathbf{l}_1}{\|\mathbf{F}_h\mathbf{x}-\mathbf{l}_1\|}\right)'\dot{\mathbf{l}}_1\right)\end{aligned} \tag{38}$$

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

$$\begin{aligned}0 &= -1880 + 1176x - 240x^2 + 16x^3 - 1780y\\ &\quad + 1156xy - 240x^2y + 16x^3y - 420y^2 + 284xy^2\\ &\quad - 60x^2y^2 + 4x^3y^2\end{aligned} \tag{39}$$

$$\begin{aligned}0 &= 1974 - 1780x + 578x^2 - 80x^3 + 4x^4\\ &\quad + 882y - 840xy + 284x^2y - 40x^3y + 2x^4y\end{aligned} \tag{40}$$

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 $x_1^{a_1}x_2^{a_2}\ldots x_n^{a_n}$ with $a_1, a_2, \ldots, a_n > 0$, the degree of the term is the product $a_1 a_2 \ldots a_n$. 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.

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

$$d((i_1,s_1),(i_2,s_2)) \triangleq \begin{cases} \|\mathbf{H}\tilde{\boldsymbol{\xi}}_{i_1,s_1} - \mathbf{H}\tilde{\boldsymbol{\xi}}_{i_2,s_2}\| & i_1 \neq i_2 \text{ and } i_2 \neq \emptyset \\ \infty & i_1 = i_2 \\ c_{\max} & i_2 = \emptyset \end{cases} \tag{43}$$

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 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).

$$\mathbf{C} \triangleq \left[\begin{array}{ccccccc} \multicolumn{3}{c}{\overbrace{\hphantom{d((1,N_{C,1}),(2,1)) \quad \dots \quad d((1,N_{C,1}),(2,N_{C,2}))}}^{\text{Assignment Costs}}} & \multicolumn{4}{c}{\overbrace{\hphantom{d((1,1),\emptyset) \quad \infty \quad \dots \quad d((N_{C,1}),1),\emptyset)}}^{\text{Non-Assignment Costs}}} \\ d((1,1),(2,1)) & \dots & d((1,1),(2,N_{C,2})) & d((1,1),\emptyset) & \infty & \dots & \infty \\ d((1,2),(2,1)) & \dots & d((1,2),(2,N_{C,2})) & \infty & d((2,1),\emptyset) & \dots & \infty \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ d((1,N_{C,1}),(2,1)) & \dots & d((1,N_{C,1}),(2,N_{C,2})) & \infty & \infty & \dots & d((N_{C,1}),1),\emptyset) \end{array} \right] \tag{44}$$

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

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_{i=1}^{N_{C,1}} \sum_{j=1}^{N_{C,2}+N_{C,1}} c_{i,j}x_{i,j} \tag{45}$$

$$\text{subject to } \sum_{j=1}^{N_{C,2}} x_{i,j} = 1 \quad \forall i$$

Every row is assigned to a column.

$$\tag{46}$$

$$\sum_{i=1}^{N_{C,1}} x_{i,j} \leq 1 \quad \forall j$$

Not every column is assigned to a row.

$$\tag{47}$$

$$x_{i,j} \in \{0,1\} \quad \forall x_{i,j}$$

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),

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_{i_1=1}^{N_{C,1}} \sum_{i_2=1}^{N_{C,2}} \cdots \sum_{i_k=1}^{N_{C,k}} c_{i_1,i_2,\ldots,i_k} x_{i_1,i_2,\ldots,i_k} \tag{49}$$

$$\text{subject to} \sum_{i_2=1}^{N_{C,2}} \cdots \sum_{i_k=1}^{N_{C,k}} x_{i_1,i_2,\ldots,i_k} = 1 \quad \forall i \tag{50}$$

$$\sum_{i_1=1}^{N_{C,1}} \sum_{i_3=1}^{N_{C,3}} \cdots \sum_{i_k=1}^{N_{C,k}} x_{i_1,i_2,\ldots,i_k} \leq 1 \quad \forall j \tag{51}$$

$$\vdots \tag{52}$$

$$\sum_{i_1=1}^{N_{C,1}} \cdots \sum_{i_{k-1}=1}^{N_{C,k-1}} x_{i_1,i_2,\ldots,i_k} \leq 1 \quad \forall j \tag{53}$$

$$x_{i_1,i_2,\ldots i_k} \in \{0,1\} \quad \forall i_1, i_2, \ldots, i_k \tag{54}$$

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:

$$c_{i_1,i_2,\ldots i_k} = \sum_{n=1}^{N_C} \sum_{m=n+1}^{N_C} \|\mathbf{H}\tilde{\boldsymbol{\xi}}_{n,i_n} - \mathbf{H}\tilde{\boldsymbol{\xi}}_{m,i_m}\|. \tag{55}$$

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)

$$c_{i_1,i_2,\ldots i_k} = \sum_{n=1}^{N_C} \|\mathbf{H}\tilde{\boldsymbol{\xi}}_{n,i_n} - \mathbf{H}\boldsymbol{\mu}_{n,i_1,i_2,\ldots,i_k}\|, \tag{56}$$

where

$$\boldsymbol{\mu}_n = \sum_{n=1}^{N_C} w_n \tilde{\boldsymbol{\xi}}_{n,i_n}. \tag{57}$$

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

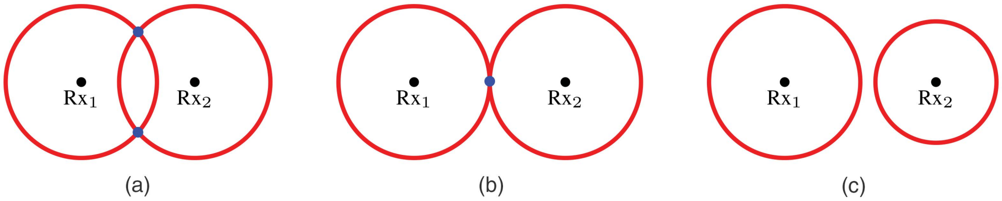


Fig. 1. A monostatic range measurement in 2D defines a circle around a sensor. Depending on the target location, there can be 2 or 1 solutions, as in (a) and (b). However, if measurements are noisy, there can be no real solutions if one tries to invert the measurement into Cartesian space as in (c). 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

$$0 = \tilde{c} + \tilde{l}_1t_1 + \tilde{l}_2t_2 + 2v_1v_2t_1t_2 + t_1^2(v_1 - \delta)(v_1 + \delta)$$
$$+ t_2^2(v_2 - \delta)(v_2 + \delta) \tag{80}$$

and in three dimensions, (78) expands to

$$0 = \tilde{c} + \tilde{l}_1t_1 + \tilde{l}_2t_2 + 2v_1v_2t_1t_2 + \tilde{l}_3t_3 + 2v_1v_3t_1t_3$$
$$+ 2v_2v_3t_2t_3 + t_1^2(v_1 - \delta)(v_1 + \delta)$$
$$+ t_2^2(v_2 - \delta)(v_2 + \delta) + t_3^2(v_3 - \delta)(v_3 + \delta) \tag{81}$$

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

$$f_{i,j}^R = \frac{r_j(\mathbf{t}-\mathbf{l}_i)'\dot{\mathbf{l}}_i + cr_ir_j}{r_i(\mathbf{t}-\mathbf{l}_j)'\dot{\mathbf{l}}_j + cr_ir_j} \tag{96}$$

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

$$\begin{aligned} 0 = {}& f_{i,j}^R r_i\mathbf{t}'\dot{\mathbf{l}}_j - r_j\mathbf{t}'\dot{\mathbf{l}}_i - f_{i,j}^R r_1(\mathbf{l}_j)'\dot{\mathbf{l}}_j + r_j(\mathbf{l}_i)'\dot{\mathbf{l}}_i \\ & + (f_{i,j}^R - 1)cr_ir_j. \end{aligned} \tag{98}$$

In 2D, the two equations to consider expand to

$$0 = -l_{i,1}^2 - l_{i,2}^2 + r_i^2 + 2l_{i,1}t_1 - t_1^2 + 2l_{i,2}t_2 - t_2^2 \tag{99}$$

$$\begin{aligned} 0 = {}& -f_{i,j}^R(\dot{l}_{j,1}l_{j,1} + \dot{l}_{j,2}l_{j,2})r_i + (\dot{l}_{i,1}l_{i,1} + \dot{l}_{i,2}l_{i,2})r_j \\ & + c(f_{i,j}^R - 1)r_ir_j + f_{i,j}^R\dot{l}_{j,1}r_1t_1 - \dot{l}_{i,1}r_jt_1 \\ & + f_{i,j}^R\dot{l}_{j,2}r_it_2 - \dot{l}_{i,2}r_jt_2 \end{aligned} \tag{100}$$

and in 3D, they are

$$\begin{aligned}0 &= -l_{i,1}^2 - l_{i,2}^2 - l_{i,3}^2 + r_i^2 + 2l_{i,1}t_1 - t_1^2 + 2l_{i,2}t_2 \\ &\quad - t_2^2 + 2l_{i,3}t_3 - t_3^2 \end{aligned} \tag{101}$$

$$\begin{aligned}0 &= -f_{i,j}^R(\dot{l}_{j,1}l_{j,1} + \dot{l}_{j,2}l_{j,2} + \dot{l}_{j,3}l_{j,3})r_i \\ &\quad + (\dot{l}_{i,1}l_{i,1} + \dot{l}_{i,2}l_{i,2} + \dot{l}_{i,3}l_{i,3})r_j \\ &\quad + c(-1 + f_{i,j}^R)r_ir_j + f_{i,j}^R\dot{l}_{j,1}r_it_1 - \dot{l}_{i,1}r_jt_1 \\ &\quad + f_{i,j}^R\dot{l}_{j,2}r_it_2 - \dot{l}_{i,2}r_jt_2 + f_{i,j}^R\dot{l}_{j,3}r_it_3 - \dot{l}_{i,3}r_jt_3. \end{aligned} \tag{102}$$

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.

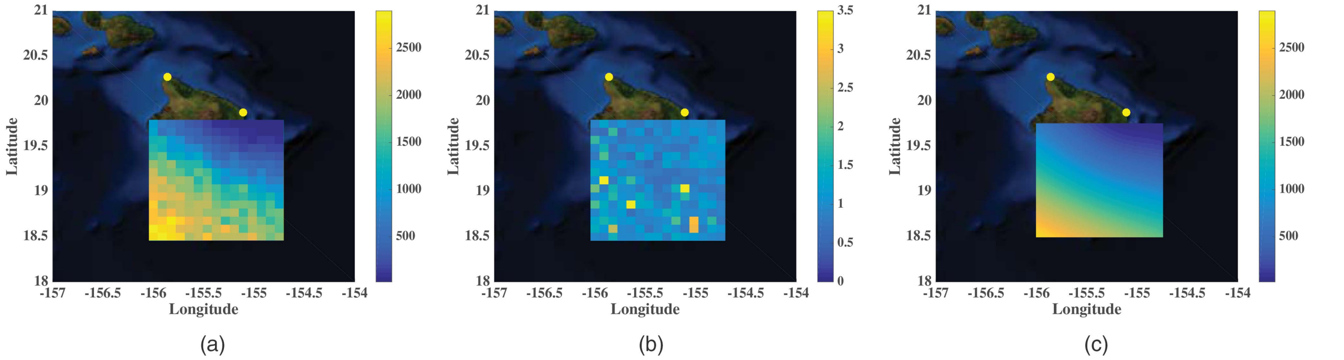


Fig. 2. The RMSE in meters and NEES of the simulations in Scenario 1 (range and emitter range-rate measurements) on a 15 × 15 grid shown on top of a map after 50 Monte Carlo runs using Bertini. Also, the square root of the trace of the approximate CRLB (in meters, on a fine grid) is shown for comparison 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.

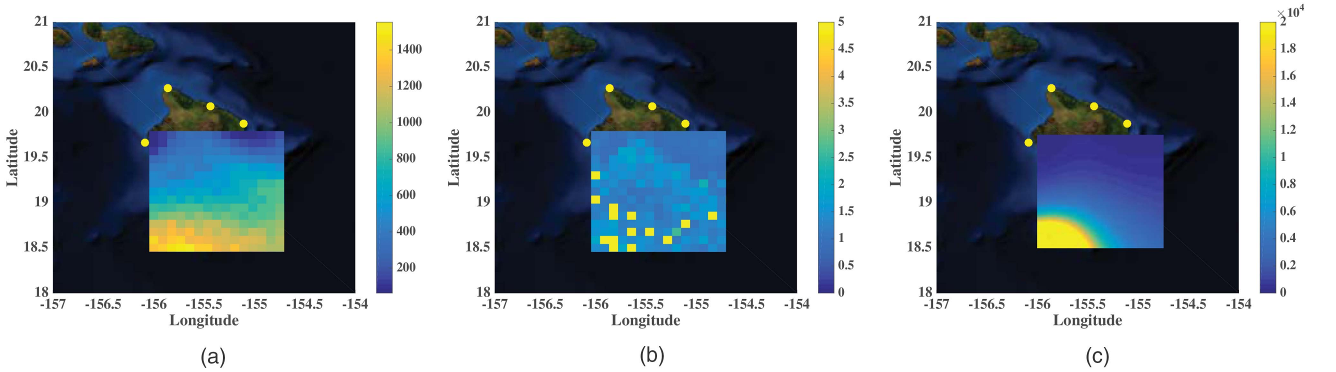
2) The second scenario is a slight variant on an "easy" scenario. All four sensors are used. Three of them produce TDOA measurements, while a bistatic path is given between a pair of them. A standard deviation of 10 m is used for the range measurement. For TDOA measurements, a standard deviation of 3.3356×10^{-8} 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



(a) (b) (c)

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.

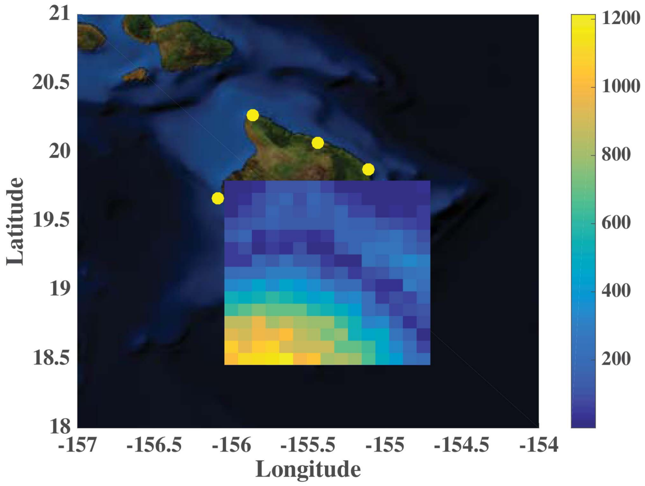


Fig. 4. The magnitude of the bias of the estimates in Scenario 2. The bias is caused by only selecting the closest hypothesis for comparison to the RMSE. The true distribution is multimodal and as the modes separate, the bias becomes larger. Additionally, due to the estimator itself not being guaranteed 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.

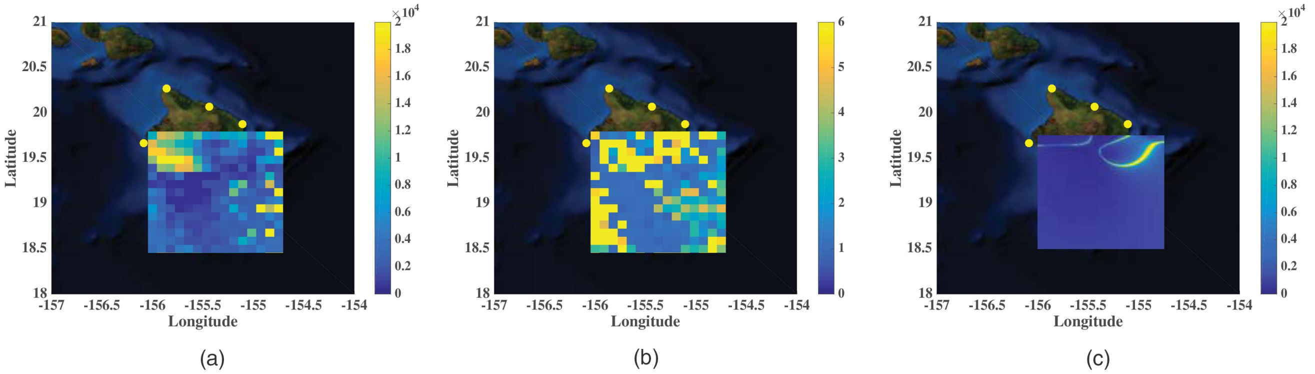
Despite the difficulty of the problem and the slow run times of the solvers, one can see in Fig. 5 that over much of the region, estimates better than 10 km accuracy are possible. This suggests that with additional refinement and optimization, one should be able to obtain a usable estimation routine.

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



(a)

(b)

(c)

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

$$p(\mathbf{x} \mid \hat{\mathbf{z}}_i) = |2\pi\mathbf{R}_i|^{-1/2} e^{-(1/2)(h_i(\mathbf{x})-\hat{\mathbf{z}}_i)'\mathbf{R}_i^{-1}(h_i(\mathbf{x})-\hat{\mathbf{z}}_i)} \tag{112}$$

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

$$p(\mathbf{x} \mid \hat{\mathbf{z}}_i) = \prod_{i=1}^{n} p(\mathbf{x} \mid \hat{\mathbf{z}}_i). \tag{113}$$

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

$$\hat{\mathbf{x}}_{ML} = \arg\min_{\mathbf{x}} \sum_{i=1}^{N} (h_i(\mathbf{x}) - \hat{\mathbf{z}}_i)'\mathbf{R}_i^{-1}(h_i(\mathbf{x}) - \hat{\mathbf{z}}_i). \tag{114}$$

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

$$0 = \sum_{i=1}^{N} (h_i(\mathbf{x}) - \mathbf{z})'\mathbf{R}_i^{-1} \left(\frac{\partial}{\partial x_i} h_i(\mathbf{x})' \right). \tag{115}$$

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

$$\mathcal{C}_{i,j} = \frac{1}{\sigma_r^2}(r^M - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\| - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|)^2 \tag{116}$$

where σ_r^2 is the variance of the measurement. The gradient with respect to the elements of $\mathbf{x}$ is

$$\begin{aligned}\nabla_{\mathbf{x}}\mathcal{C}_{i,j} = {} & -\frac{2}{\sigma_r^2}(r^M - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\| - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|) \\ & \cdot \left(\frac{\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}_i}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\|} + \frac{\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}_j}{\|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|} \right).\end{aligned} \tag{117}$$

Introducing the additional equations in terms of r_i, r_j, $\tilde{r}_i$ and $\tilde{r}_j$, one gets

$$r_i^2 - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_i\|^2 = 0 \tag{118}$$

$$r_j^2 - \|\mathbf{F}_h\mathbf{x} - \mathbf{l}_j\|^2 = 0 \tag{119}$$

$$\tilde{r}_i r_i - 1 = 0 \tag{120}$$

$$\tilde{r}_j r_j - 1 = 0 \tag{121}$$

and one can write (117) as

$$\begin{aligned}\nabla_{\mathbf{x}}\mathcal{C}_{i,j} = {} & -\frac{2}{\sigma_r^2}((\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}_j)(\tilde{r}_j(r^M - r_i) - 1) \\ & + (\mathbf{F}_h'\mathbf{F}_h\mathbf{x} - \mathbf{F}_h'\mathbf{l}_i)(\tilde{r}_i(r^M - r_j) - 1))\end{aligned} \tag{122}$$

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

$$\mathbf{l}_1 = \begin{bmatrix} -9.088503107295082 \\ -3.592899795686701 \\ 11.379375304771440 \end{bmatrix} \tag{123}$$

$$\mathbf{l}_2 = \begin{bmatrix} -8.737071884578572 \\ 12.184039601570143 \\ 0.461252502515841 \end{bmatrix} \tag{124}$$

$$\mathbf{l}_3 = \begin{bmatrix} -10.997619107424038 \\ -0.372458566000544 \\ 10.193804421541278 \end{bmatrix} \tag{125}$$

$$\mathbf{l}_4 = \begin{bmatrix} -12.099924003565013 \\ -2.341482530709476 \\ 8.550397573582972 \end{bmatrix} \tag{126}$$

where $\rho_{i,j} = \mathrm{TDOA}_{i,j}/c$ is the range difference between Sensor i and Sensor j and

$$\rho_{1,2} = -6.634727887894795 \tag{127}$$

$$\rho_{1,3} = -1.197770770524833 \tag{128}$$

$$\rho_{1,4} = -2.553916244191934. \tag{129}$$

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,

$$a_6 + a_5x + a_4x^2 + a_3y + a_2xy + a_1y^2 = 0 \tag{130}$$

$$b_6 + b_5x + b_4x^2 + b_3y + b_2xy + b_1y^2 = 0, \tag{131}$$

one can eliminate x to get a univariate equation of the form

$$c_0 + c_1y + c_2y^2 + c_3y^3 + c_4y^4 = 0 \tag{132}$$

having coefficients (pay attention to parentheses)

$$\begin{aligned} c_0 = {} & a_6^2b_4^2 + b_6(a_5^2b_4 - a_4a_5b_5 + a_4^2b_6) \\ & + a_6(-a_5b_4b_5 + a_4(b_5^2 - 2b_4b_6)) \end{aligned} \tag{133}$$

$$\begin{aligned} c_1 = {} & a_5^2b_3b_4 + a_6b_4(2a_3b_4 - a_2b_5) + 2a_4^2b_3b_6 \\ & - a_5(a_6b_2b_4 + a_4b_3b_5 + a_3b_4b_5 + a_4b_2b_6 - 2a_2b_4b_6) \\ & + a_4(-2a_6b_3b_4 + 2a_6b_2b_5 + a_3b_5^2 - 2a_3b_4b_6 - a_2b_5b_6) \end{aligned} \tag{134}$$

$$
\begin{aligned}
c_2 &= b_4(a_5^2b_1 + (a_3^2 + 2a_1a_6)b_4 - a_5(a_3b_2 - 2a_2b_3 + a_1b_5) \\
&\quad - a_2(a_6b_2 + a_3b_5) + a_2^2b_6) + a_4^2(b_3^2 + 2b_1b_6) \\
&\quad - a_4(-a_6b_2^2 + a_5b_2b_3 + 2a_6b_1b_4 + 2a_3b_3b_4 + a_5b_1b_5 \\
&\quad - 2a_3b_2b_5 + a_2b_3b_5 - a_1b_5^2 + a_2b_2b_6 + 2a_1b_4b_6)
\end{aligned}
\tag{135}
$$

$$
\begin{aligned}
c_3 &= 2a_4^2b_1b_3 - a_4(a_5b_1b_2 - a_3b_2^2 + a_2b_2b_3 + 2a_3b_1b_4 \\
&\quad + 2a_1b_3b_4 + a_2b_1b_5 - 2a_1b_2b_5) \\
&\quad + b_4(-a_1a_5b_2 + a_2^2b_3 + 2a_1a_3b_4 \\
&\quad + a_2(2a_5b_1 - a_3b_2 - a_1b_5))
\end{aligned}
\tag{136}
$$

$$
\begin{aligned}
c_4 &= a_4^2b_1^2 + b_4(a_2^2b_1 - a_1a_2b_2 + a_1^2b_4) \\
&\quad + a_4(-a_2b_1b_2 + a_1(b_2^2 - 2b_1b_4)).
\end{aligned}
\tag{137}
$$

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:

$$
x = \frac{-a_5 - a_2y \pm \sqrt{(a_5 + a_2y)^2 - 4a_4(a_6 + a_3y + a_1y^2)}}{2a_4}.
\tag{138}
$$

Substituting into (131), one similarly gets

$$
x = \frac{-b_5 - b_2y \pm \sqrt{(b_5 + b_2y)^2 - 4b_4(b_6 + b_3y + b_1y^2)}}{2b_4}.
\tag{139}
$$

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 ± 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.

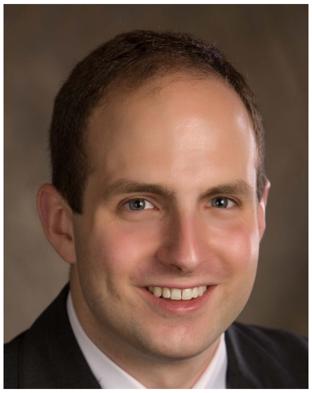
[70] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walkder
"Algorithm 777: HOMPACK90: A suite of Fortran 90 codes for globally convergent homotopy algorithms,"
ACM Transactions on Mathematical Software, vol. 23, no. 4, pp. 514–549, Dec. 1997.

[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.