June 2020 Volume 15 Number 1 ISSN 1557-6418

JAIF 20

# Journal of Advances in Information Fusion

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

Regular Papers Page



*From the Editor-In-Chief*

*Looking Back and Looking Forward*



ISIF

# INTERNATIONAL SOCIETY OF INFORMATION FUSION

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




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

# From the Editor-in-Chief:

*June 2020*

**Looking Back and Looking Forward**

Welcome to the June 2020 issue of the ISIF *Journal of Advances in Information Fusion* (JAIF). This year we embark on the 15th year of JAIF, and it is my first as Editor-in-Chief. I have served JAIF in several editorial roles since 2008, and I am privileged and honored to take over the leadership role this year.

I would encourage all of us in the information fusion community to take time to familiarize ourselves with the contents of the 28 excellent issues of JAIF that have been published in the last 14 years. This includes a total of 147 full articles, many informative editorials, and five special issues on the following topics:

- Nonlinear Derivative-Free Filters: Theory and Applications;
- Estimation Involving Directional Quantities;
- Extended Object Tracking;
- Evaluation of Uncertainty Representation and Reasoning Techniques;
- Multiple-Hypothesis Tracking.

The success of JAIF rests principally on the hard work and creative research efforts of the information fusion community, leading to a collection of excellent published manuscripts. Equally important are the efforts of the JAIF editorial board. This has been led by two distinguished Editors-in-Chief: Dale Blair (Vols. 1–8, 2006–2013) and Uwe Hanebeck (Vols. 9–14, 2014–2019). The initial vision for JAIF was encouraged and led first by our former VP for Publications Yaakov Bar-Shalom, and now by our current VP for Publications Dale Blair. Behind the scenes, exceptional effort was expended by our first Administrative Editor Robert Lynch (2006–2015), and since then by our current Administrative Editor David Krout. Managing the journal would simply be impossible without this support. I would like especially to thank David for helping me as I have taken over the Editor-in-Chief duties this year.

What is the vision for JAIF? It is to be the premier forum for archival work in information fusion. Of course, other fine publication venues do exist. Nonetheless, JAIF is uniquely positioned at the intersection of multiple lines of research that span signal-level

processing, communications, detection-level and track-level exploitation, situational awareness, and uncertainty modeling for a broad range of applications. Ultimately, the focus areas of the journal are defined empirically by what has been published and what special issues are in preparation. Again, I encourage all to browse the contents of past issues!

In addition to familiarizing ourselves with the JAIF past, I would like to call on the information fusion community to participate actively in the present and future of JAIF. First, consider volunteering to review papers or to join the editorial board. Second, consider preparing a manuscript for submission. We particularly encourage the submission of expanded versions of papers presented in the ISIF FUSION conference series. Preparing a journal manuscript is hard work, and, particularly for researchers in industry, often needs to be done on personal time in addition to other professional commitments. It can be an opportunity to develop one's work more fully while furthering its relevance to the research community.

Finally, consider serving as a guest editor while organizing a special issue on a topic of interest to the community. Note that five of the last nine JAIF issues are special issues; we plan to continue to have approximately half of future issues be special issues. This allows those outside the editorial board to have a significant impact on the direction of the journal.

Contributing to JAIF in a variety of capacities is a rewarding endeavor that will help to strengthen the standing and relevance of JAIF in the research community, to the benefit of all JAIF authors and readers. I look forward to working together with all of you in the years to come.

Stefano Coraluppi
Editor in Chief

# Context-Aware Dynamic Asset Allocation for Maritime Surveillance Operations

**LINGYI ZHANG**
**DAVID SIDOTI**
**GOPI VINOD AVVARI**
**DIEGO F. M. AYALA**
**MANISHA MISHRA**
**DAVID L. KELLMEYER**
**JAMES A. HANSEN**
**KRISHNA R. PATTIPATI**

**This paper formulates and solves a maritime surveillance problem involving the allocation of *multiple* heterogeneous assets over a large area of responsibility to detect *multiple* drug smugglers using *heterogeneous* types of transportation on the sea with varying contraband weights. The asset allocation is based on a probability of activity surface, which represents spatiotemporal target activity obtained by integrating intelligence data on drug smugglers' whereabouts/waypoints for contraband transportation, their behavior models, and meteorological and oceanographic information. A number of algorithmic concepts based on branch-and-cut with limited search and approximate dynamic programming (ADP) were investigated. We validate the proposed algorithmic concepts via realistic mission scenarios. We conduct scalability analyses of the algorithms and conclude that effective asset allocations can be obtained within seconds using rollout-based ADP. The contributions of this paper have been transitioned to and are currently being tested by Joint Interagency Task Force—South, an organization tasked with providing the initial line of defense against drug trafficking in the East Pacific Ocean and the Caribbean Sea.**

Manuscript received February 23, 2019; revised June 11, 2019 and September 26, 2019; released for publication June 30, 2020.

Refereeing of this contribution was handled by Ramona Georgescu.

L. Zhang, D. F. M. Ayala, and K. R. Pattipati are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA (E-mail: lingyi.zhang@uconn.edu, krishna.pattipati@uconn.edu).
D. Sidoti and J. A. Hansen are with the U.S. Naval Research Laboratory (NRL-MRY), Monterey, CA 93943, USA (E-mail: david.sidoti@nrlmry.navy.mil, james.hansen@nrlmry.navy.mil).
G. V. Avvari and M. Mishra are with Delphi Technologies, Kokomo, IN 46902, USA.
D. L. Kellmeyer is with the Naval Information Warfare Center (NIWC)—Pacific, San Diego, CA 92152, USA (E-mail: dave@spawar.navy.mil).
This work was supported in part by the U.S. Office of Naval Research and Naval Research Laboratory under Grant N00014-18-1-2838, Grant N00014-16-1-2036, Grant N00173-16-1-G905, and Grant HPCM034125HQU.



## I. INTRODUCTION

### A. Motivation

The illicit drug trade is an extremely profitable industry and it is estimated that the consumers in the United States alone spend as much as 150 billion USD per year on black market drugs. Of this, it is estimated that 37 billion USD is spent on cocaine alone. It is a problem of national, and increasingly international, concern [1], [2]. This problem increased exponentially with the advent of narco-terrorism and the prospect of terrorists using narcotics smuggling techniques to transport terrorists or weapons of mass destruction into the country. Given the reduction in the national resources allocated to the counter-narcotics threat, it is of paramount importance that smarter and faster decision support tools that integrate a wide variety of information are developed to assist in this challenge of using less to accomplish more. To do so requires effective hybrid human–machine systems.

The U.S. Navy has shown a growing interest in mixed-initiative human–machine systems and mastering information dominance for effective context-driven operations [3]. To do so requires the transfer of the right data from the right sources in the right context to the right decision maker (DM) at the right time for the right purpose—a concept known as 6R [4]. If a dynamically developing operational context can be understood by the DM, appropriate courses of action (COAs) can be carried out, given the unfolding events. In the context of maritime operations, DMs must assimilate information from a multitude of sources before making decisions on the strategy to be followed each day. If the DMs are better informed about what to expect given the currently accessible data, as well as what they might expect in the case of unforeseen events, effective decisions can be made on the COAs.

Currently, much planning for narcotics seizures is performed by humans interpreting large amounts of data, including weather forecasts, intelligence, and recently reported contacts of interest. Each day, the targeting analysts must process and interpret all of these data and agree upon a COA amounting to where limited detection aircraft and interdiction vessels should be allocated to disrupt the maximum amount of shipments over a multiday planning cycle. The consolidation of large amounts of data and possible strategies into a single asset allocation optimizer is beneficial for both algorithmic purposes and human understanding. To support this transition to a human–machine

collaborative mode of operation, we have developed an optimization-based modeling framework and the associated decision support software tool for dynamic surveillance and interdiction resource management in counter-smuggling operations. This tool, named COAST or Courses Of Action Simulation Tool [5], and the corresponding algorithms are intended to support targeting analysts in identifying high-probability areas of smuggler presence and to proactively develop a set of high-value COAs.

The counter-smuggling problem presented in this paper is viewed as a moving horizon stochastic control problem, as illustrated in Fig. 1, specifically from a strategic operations standpoint, i.e., decision making with regard to a schedule to follow for the upcoming time horizon. Here, each block is an entity, such as a DM, sensor, or asset, and the link from each block represents the outcome of the block and its impact or influence on the next block. The problem can be decomposed into surveillance and interdiction asset allocation subproblems. This paper focuses on the surveillance component, where DMs (also termed targeteers) are responsible for allocating multiple aircraft (namely, P-3 Orions (manned)) over a finite time horizon in an effort to detect the transportation of contraband. The interdiction component, detailed in [6], involves the allocation of multiple heterogeneous surface assets (namely, Navy ships, Coast Guard cutters), to disrupt multiple drug smugglers of varying types, similar to that which is addressed in this paper. The DMs in Fig. 1 choose which surveillance assets to allocate to which target(s) (smugglers) based on the target type and intelligence forecasting the target's trajectory (specified in the form of probability of activity (PoA) surfaces [7], [8]). After allocated assets attempt to search for potential targets, the mission environment changes due to any target detection that may occur or due to weather changes. These environment changes are recorded by sensors and operators, processed, and sent back to the DMs in the form of target types and tracks, and are combined into an updated PoA surface, providing a new forecast for the remainder of the planning time horizon. The process then repeats. Ideally, the results of this paper feed that of [6] for *coordinated* smuggler detection and interdiction.

## B. Related Research

The surveillance mission involves the search, detection, tracking, and identification of potential smugglers within a large geographic region, which plays an essential role in the counter-smuggling operation. Airborne surveillance assets (e.g., helicopters, maritime patrol aircraft) are highly efficient at determining the sea surface traffic information. However, in a real-world scenario, there are typically a limited number of surveillance assets and a large sea surface area that needs to be surveilled. The study of how to most effectively employ limited resources to locate an object, whose location is not precisely known, falls under the rubric of search theory.

The earliest foundations of search theory were built by Koopman [9] to aid the U.S. Navy in efficiently locating enemy submarines during World War II, which was further generalized in [10]. There are two major categories of search theory: 1) the optimal allocation of effort problem and 2) the best track problem [11]. For the optimal effort allocation problem, Blachman and Proschan [12] derived an optimum search pattern for a generalized problem of finding an object in one of the $n$ boxes. Pollock [13] introduced a Bayesian approach to the optimal allocation problem, where allocation decisions are made sequentially based on observations up to the current time in order to minimize the expected cost of searching to satisfy a specified probability of detection (PD). Charnes and Cooper [14] applied convex programming, along with the Kuhn–Tucker conditions, for the optimum distribution of effort computation. In this paper, we adopt Charnes and Cooper's method to compute the effort required for the optimal search in a discretized map.

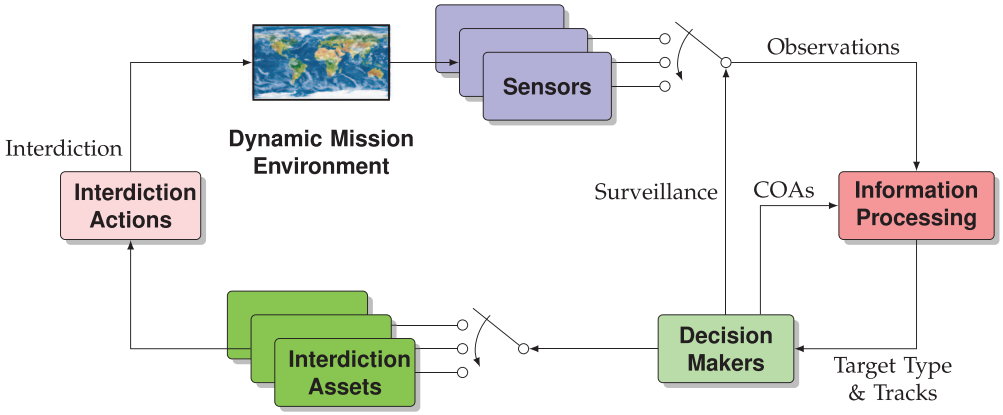



Fig. 1. The counter-smuggling problem viewed from a stochastic control standpoint. Targeteers (DMs) choose from a set of available surveillance assets and finalize a search schedule to allocate the asset(s) over a near-time planning horizon, typically 72 h. Similar to the planning/decision process presented in [6], [7], [32], and [40], after the action is carried out, information is gathered, processed, and fed back to the targeteer.

Stone [15] made use of the calculus of variations, convexity properties, and generalized Lagrange multiplier techniques to formulate a systematic treatment of search theory. For the best track problem, Lukka [16] worked out the theory of optimal search for stationary targets, targets whose motion is known, and targets whose motion is almost known. The method relies on the theory of optimal control. Mangel [17] extended Lukka's algorithms with the option of incorporating a detection rate that is either independent of or dependent on velocity.

In recent years, the problem of drug surveillance has been formulated from a variety of viewpoints. For example, Washburn and Wood [18] formulated the surveillance problem as a two-person zero-sum game and Pfeiff [19] applied search theory to a defender–attacker optimization model that maximizes the defender's probability of success. Royset and Wood approach the problem as a network flow problem, wherein an interdictor must destroy a set of arcs on a network to minimize both the interdiction cost and the maximum flow of smugglers [20]. Jacobson et al. [21] formulate the problem as a multiple traveling salesman problem with the objective of minimizing the overall search route cost for multiple platforms that visit every search location. Ng and Sancho [22] developed a dynamic programming method to solve the surveillance problem. However, the dynamic programming approach suffers from the curse of dimensionality for large problems and, consequently, near-optimal approximations are needed. A common way to overcome this curse is by approaching the problem via approximate dynamic programming (ADP) with policy iteration as in [23], where they frame the problem in terms of stochastic control with partially observable Markov decision processes. Kress et al. [24] examine a discrete-time and discrete-space stochastic dynamic programming approach to coordinate the efforts of a single aerial search asset and a single surface interdiction asset. Other approaches, including the formulation of the surveillance problem as a resource-dependent orienteering problem [25]–[27], wherein reward depends on the resource expended at each visited node, have been investigated.

Optimal search problem formulations have become versatile in their ability to account for multiple cooperating searchers, multiple targets with different characteristics, and environmental effects on the search [28]–[31]. For example, arc inspection is based on the inverse of the probabilities of detection as opposed to PoA surfaces accounting for weather and intelligence in [7], [8], and [32]. Byers [33] extended the network modeling approach to drug interdiction by including Bayesian updating of the PoA surface. He considered a scenario with one unmanned aerial vehicle and one ground-based interceptor to interdict multiple targets with different deadlines. Bessman [34] developed a defender–attacker optimization model that uses the PoA surfaces as the basis for asset allocation against smugglers. He formulated a stochastic shortest path problem and represented smuggler behavior as the output of an all-to-one label-correcting temporal dependence instead of one-step dependence. Three different sensor types (one interdiction and two surveillance) are considered for allocation to prosecute one type of target (among three possible). In this defender–attacker model, smugglers are assumed to have imperfect knowledge of possible sensor locations and are given the ability to modify their behavior in response to this information.

### C. Paper Organization

Similar to Pietz and Royset [25], we also discretized our maritime map. We adopt Charnes and Cooper's method [14] to compute the effort required for optimal search in a discretized map. Our novel algorithmic contributions are the following:

1) Fast one- and two-step lookahead ADP (1SLADP and 2SLADP) algorithms for maritime surveillance composed of heterogeneous assets and heterogeneous targets, each of which is carrying not necessarily the same amount of contraband. Our algorithms exploit the fusion of intelligence and weather information available in the PoA surfaces.
2) We measure the utility of our approach by way of comparison with more traditional branch-and-cut (B&C) algorithms to solve the surveillance problems. We develop two variations of the ADP-based surveillance asset allocation algorithms, wherein real-world constraints on the assets (e.g., endurance and rest time) are explicitly considered.

The paper is organized as follows. Section II describes the problem and the technical challenges addressed in the development of allocation algorithms underlying our decision support tool. In Section III, we discuss solution approaches, including exhaustive and greedy B&C and ADP. In Section IV, we present simulation results as applied to a benchmark scenario that has multiple targets, multiple surveillance assets, and parameters that have multiple levels of uncertainty. We additionally conduct and present results from our sensitivity analysis relating to the scalability and performance of our solution approaches in a realistic mission scenario. We conclude the paper in Section V with a summary of our findings and future work.

## II. PROBLEM MODEL AND FORMULATION

### A. Problem Definition and Solution Architecture

The complete maritime surveillance and interdiction problem is one of maritime drug trafficking disruption in the East Pacific Ocean and the Caribbean Sea. The general mission consists of two components: 1) surveillance (the detection, tracking, and identification of contacts of interest) and 2) interdiction (the interception, investigation, and potential apprehension and repatri-

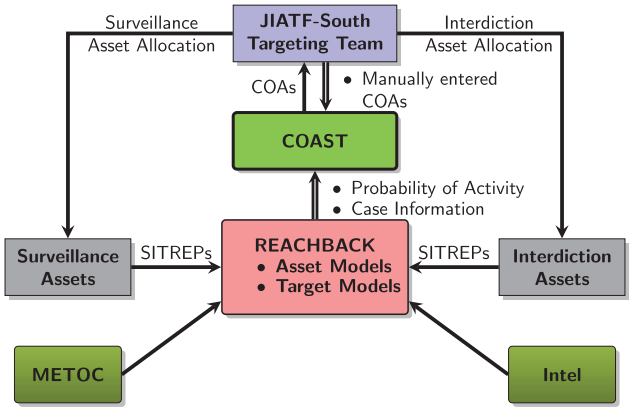



Fig. 2. Information flow and decisions (controls) in the counter-smuggling problem. The decision support tool, COAST, provides COAs to the JIATF-South Targeting Team who then modify them as they see fit. The manually entered COAs can then be fed back into the tool where the simulation is rerun providing new outcomes to the targeting team, who can then provide further feedback and modifications, if necessary.

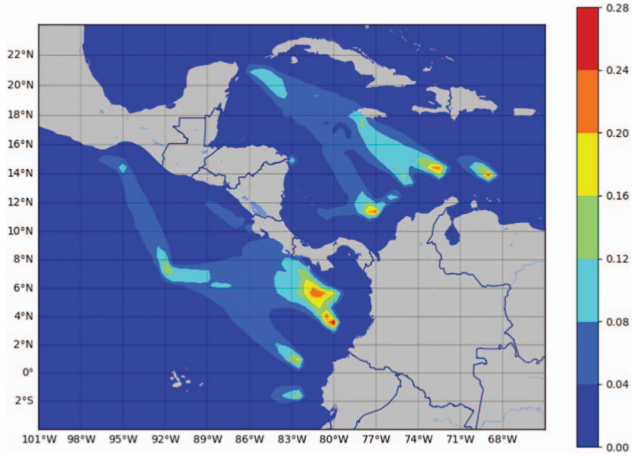



Fig. 3. PoA surface $\mathrm{PoA}(q, k, j)$ summed over all $k$.

ation of smugglers). In response to the need for information fusion, we proposed a decision support system (DSS) in [5], named COAST, to host and utilize algorithms to provide auxiliary support to Joint Interagency Task Force—South (JIATF-South) targeteers. We proposed different forms of visualizations to enable DMs to understand the behavior of our algorithms and the presently evolving context, while also providing functionality for human input and interaction in order to effectively integrate both humans and decision support algorithms for mixed-initiative planning. The information flow for the complete maritime interdiction problem is illustrated in Fig. 2.

In COAST, we solve a moving horizon dynamic resource management problem for both surveillance and interdiction operations based on user-defined mission parameters. We then provide suggested COAs that the DMs can interact with, adjust, and fine-tune to analyze various "what–if" scenarios and to obtain a satisfactory allocation. Visual and computational analytics are provided to communicate the reasons behind our algorithm's behavior. From Fig. 2, continuously updated PoA surfaces (see Fig. 3 for an example), representing the posterior probabilities of smugglers' presence, constitute the sufficient statistics for decision making [35]—that is, COAST does not need to know how specific Intel or meteorology and oceanography (METOC) features, for example, uncertainty associated with a drug trafficker, wave heights, currents, etc., and how these two inputs, along with asset and target models, are combined to produce the PoA surface. A targeteer can fine-tune the allocations, the resulting COAs are executed, and observations from surveillance and interdiction assets are sent back to the reachback cell in the form of situational reports or SITREPs (e.g., detections or nondetections) that are used to update the PoAs. The targeteer can specify multiple objective functions. The objectives considered and analyzed in this paper are as follows:

O1: Maximize the normalized weight of the contraband detected (normalized by the total possible amount of contraband).

O2: Maximize the normalized number of detections (normalized by the total possible number of cases).

O3: Maximize the normalized number of smugglers detected (normalized by the total possible number of smugglers).

Let $\alpha_j$ and $\rho_j$ denote the expected contraband weight and expected number of smugglers for case $j$. Let $C$ be the total number of cases (i.e., predicted smuggler tracks) to be searched. Then, the normalized priority weights for objectives O1–O3, respectively, are as follows:

$$\lambda_j = \frac{\alpha_j}{\sum_{g=1}^{C} \alpha_g}, \tag{1}$$

$$\lambda_j = \frac{1}{C}, \tag{2}$$

$$\lambda_j = \frac{\rho_j}{\sum_{g=1}^{C} \rho_g}. \tag{3}$$

## B. Problem Formulation

The notation used in this paper is listed in Table I.

1) PoA Surface: The foundation for each asset allocation solution is the PoA surface over multiple time epochs. The PoA surface is the result of combining METOC information with actionable intelligence with regard to uncertain smuggler departure point(s), departure times, waypoint(s), destination(s), and their behavior on the ocean. The spatiotemporal probability surface, PoA, is calculated as the joint probability of two discrete random events: 1) the case $j$, with a corresponding binary random variable $C_j$, i.e., how trustworthy the intelligence source is regarding a target, and 2) the target correspond-

TABLE I
Summary of Notations

| | |
|---|---|
| $A$ | Total number of surveillance assets |
| $\mathcal{A}_j$ | Total area to be searched for case $j$ |
| $B_{ij}$ | Great Circle distance from the base of asset $i$ to the centroid of case $j$ |
| $C$ | Total number of cases |
| CPoSD$(i, j)$ | Cumulative probability of successful detection for a given asset $i$ allocated to case $j$ |
| $d_{i\ell}$ | Landing time for asset $i$'s $\ell$th flight |
| $i$ | Surveillance asset index |
| $j$ | Case index |
| $k$ | Time epoch index |
| $K$ | End of planning time horizon |
| $L_i$ | Endurance of asset $i$ |
| PoA$(q, k, j)$ | Likelihood that a smuggler belonging to case $j$ is located in a cell $q$ at time $k$ |
| $R_i$ | Downtime of asset $i$ |
| $s_{i\ell}$ | Remaining search time available within the current sortie for asset $i$ |
| $S_{ij}$ | Sweep width of asset $i$ searching for target $j$ |
| $t_{ij}$ | Travel time for traversing the distance $B_{ij}$ |
| $v_i^{\mathrm{a}}$ | Travel speed of asset $i$ |
| $v_i^{\mathrm{s}}$ | Search speed of asset $i$ |
| $w_{ijk}$ | Reward of allocating asset $i$ to case $j$ at time $k$ |
| $x_{ijk}$ | Binary decision variable of allocating asset $i$ to case $j$ at time epoch $k$ |
| $\lambda_j$ | Priority weight of case $j$ |
| $\tau_{i\ell}$ | Departure time for asset $i$'s $\ell$th flight |
| $\gamma(i, j, \ell)$ | The set of search time indices for asset $i$ assigned to case $j$ for the $\ell$th flight |

ing to case $j$ at a location $q$ at time epoch $k$, with a corresponding binary random variable $\mathcal{X}(q, k, j)$, i.e., given that the case $j$ exists, the probability that the target exists at a location $q$ at time $k$. The probability surface PoA is indexed by a location $q$, time $k$, and case $j$, and is defined in (4)–(7):

$$\mathrm{PoA}(q, k, j) = P\left(\mathcal{C}_j = 1 \cap \mathcal{X}\left(q, k, j\right) = 1\right) \tag{4}$$

$$= E\left\{\mathcal{C}_j \cdot \mathcal{X}\left(q, k, j\right)\right\} \tag{5}$$

$$= E_{\mathcal{C}_j}\left\{\mathcal{C}_j \cdot E_{\mathcal{X}(q,k,j)|\mathcal{C}_j}\left(\mathcal{X}\left(q, k, j\right)|\mathcal{C}_j\right)\right\} \tag{6}$$

$$= \sum_{c_j=\{0,1\}} c_j \cdot P(\mathcal{C}_j = c_j) \cdot \left(\sum_{h=\{0,1\}} h \cdot P(\mathcal{X}\left(q, k, j\right) = h|\mathcal{C}_j = c)\right), \tag{7}$$

where we separate the expectation in (6) based on the law of total expectation/iterated expectations.

We assume that $P(\mathcal{C}_j = 1) = 1$, that is, the intelligence sources are always correct with 100% certainty. Then, (7) reduces to

$$\mathrm{PoA}(q, k, j) = P\left(\mathcal{X}\left(q, k, j\right) = 1\right). \tag{8}$$

Therefore, PoA$(q, k, j)$ is a number that refers to the likelihood that a smuggler, belonging to case $j$, is located in a cell $q$ at time $k$. The PoA surfaces are computed as detailed in [8] and represent all the relevant information for effective asset allocation. The DM can specify how many planning epochs to optimize over based on these PoA surfaces and the objective function to be optimized. A typical PoA surface PoA$(q, k, j)$, summed over all $k$, is shown in Fig. 3.

2) Optimal Search Effort Calculation: We assume the optimum distribution of search effort is known based on the model in [14]. Let $p_{jkq}$ denote the PoA of target $j$ in cell $q$ at time $k$. We first rank the nonzero PoA cells in decreasing order such that $p_{jk[1]} \geq p_{jk[2]} \geq \cdots$, where $[\kappa]$ denotes the $\kappa$th largest nonzero PoA cell. Let the total available effort to be expended by asset $i$ to search case $j$ be $\Phi_{ij}$. A critical threshold is then calculated to narrow the problem space and eliminate PoA cells not worth searching, by first finding an $n$ that satisfies the following inequality [14]:

$$\sum_{v=1}^{n} v\left[\ln p_{jk[v]} - \ln p_{jk[v+1]}\right] > \Phi_{ij}. \tag{9}$$

Then, the critical probability, $\rho_{ijk}$, corresponding to the search of case $j$ by asset $i$ at time $k$, is as in (10):

$$\rho_{ijk} = \frac{1}{n}\left(\sum_{v=1}^{n} v\left(\ln p_{jk[v]} - \ln p_{jk[v+1]}\right) - \Phi_{ij}\right) + \ln p_{jk[n+1]}. \tag{10}$$

We then select all the cells corresponding to case $j$ that have a PoA greater than the critical probability found in (10). This reduces the number of potential cells that need to be searched for each case $j$ by asset $i$. We then compute the patrol box that maximally covers the high-probability cells for each case. The allocation of assets to patrol boxes is the subject of the optimization problem discussed next.

3) Optimization Problem: The case regions are labeled by aggregating the PoA surfaces over a discrete planning time period of length $K$ (e.g., 72 h). Let us assume a moving horizon frame of reference, where $k = 0$ corresponds to the current time period of unit length ($\Delta = 1$ h), $k = 1$ corresponds to the first planning period, and $k = K$ corresponds to the final period to be planned for. Let $A$ be the total number of surveillance assets, $C$ be the total number of cases, and $q \in Q(j)$ be the set of cells in the patrol box for case $j$ as determined by the optimal search effort calculation algorithm. The size of the patrol box depends on the concept of operations and is assumed known. Let $w_{ijk}$ be the probability of successful detection (PoSD), which is the product of the PoA surface and the PD when asset $i$ is assigned to

search for case $j$ at time $k$. That is,

$$w_{ijk} = \sum_{q \in Q(j)} \text{PoA}(q, k, j)\text{PD}(i, j, k), \tag{11}$$

where

$$\text{PD}(i, j, k) = 1 - \exp\left(-\frac{S_{ijk} v_i^{\text{s}} \Delta}{\mathcal{A}_j}\right) \tag{12}$$

is the probability that asset $i$ detects case $j$ during the $k$th time epoch interval (PD$(i, j, k)$ can only be collected at the end of the $k$th time epoch interval). Let us assume that each asset travels to the search region at a speed $v_i^{\text{a}}$ and searches in the search region at a speed $v_i^{\text{s}}$. The PD equation is adopted from Koopman's random search formula [9], and offers a lower bound on the PD; advanced models may be used in place of (12) as in [15]. Here, $S_{ijk}$ is the sweep width of asset $i$ searching for case $j$ at time epoch $k$, and $\Delta$ is the interepoch interval (=1 h in this paper).

Let $B_{ij}$ represent the geodesic[1] distance that asset $i$ must traverse from its base to the centroid of case $j$. The time it takes to traverse $B_{ij}$, denoted by $t_{ij}$, is given by

$$t_{ij} = \left\lceil \frac{B_{ij}}{v_i^{\text{a}}} \right\rceil, \tag{13}$$

where $\lceil \cdot \rceil$ denotes the ceiling or rounding up to the nearest integer. Let $\tau_{i\ell}$ denote the departure time if an asset $i$ is allocated to a case for flight $\ell$, and $d_{i\ell}$ as the landing time upon its return from the corresponding search box. The index $\ell$ increments with each flight that asset $i$ is scheduled to fly over the planning time horizon. Formally,

$$\tau_{i\ell} = \begin{cases} k, 0 < k \leq K, & \text{if i is assigned to a case during} \\ & \quad \text{the } \ell^{\text{th}} \text{ flight,} \\ \infty, & \text{otherwise.} \end{cases} \tag{14}$$

A similar definition applies to $d_{i\ell}$. For each flight, the total search and travel time for each asset from its corresponding base to each case must not exceed the asset's endurance, $L_i$ (in hours), and, upon flight completion, it must rest for $R_i$ consecutive hours before it can be scheduled to depart for the next search box. The assets are assumed to be manned aircraft with an associated rest time for the pilot; additionally, each aircraft requires periodic maintenance and refueling. The minimum time it may take for an asset to become available again for search is $L_i + R_i$. Note that there is no feasible asset allocation for a case $j$ and asset $i$ if $2t_{ij} \geq L_i$; i.e., the total round trip travel time for a search region is greater than the maximum aloft time $L_i$. With PoSD defined as in (11), the cumulative probability of successful detection (CPoSD) for a given asset $i$ is

$$\text{CPoSD}(i, j) = 1 - \prod_{k=1}^{K} \left(1 - w_{ijk} x_{ijk}\right), \tag{15}$$

where $x_{ijk}$ is a binary decision variable such that $x_{ijk} = 1$ if asset $i$ is assigned to case $j$ at time epoch $k$, and 0, otherwise. The total reward that asset $i$ can collect over the planning time horizon is then

$$r_i = \sum_j \lambda_j \text{CPoSD}(i, j), \tag{16}$$

where $\lambda_j$ is the normalized priority weight of case $j$. We wish to solve the following problem:

$$\max_{x_{ijk}, \tau_{i\ell}, d_{i\ell}} J = \max \sum_{i=1}^{A} r_i, \tag{17}$$

$$\text{s.t.} \sum_i x_{ijk} \leq 1 \quad \forall j, k, \tag{18}$$

$$\sum_j x_{ijk} \leq 1 \quad \forall i, k, \tag{19}$$

$$d_{i\ell} - \tau_{i\ell} \leq L_i \quad \forall i, \ell, \tag{20}$$

$$\tau_{i\ell+1} - d_{i\ell} \geq R_i \quad \forall i, \ell, \tag{21}$$

$$\tau_{i\ell}, d_{i\ell} \in \{0, \ldots, K\} \cup \{\infty\} \quad \forall i, \ell, \tag{22}$$

$$x_{ijk} \in \{0, 1\}. \tag{23}$$

In (17), we assume that the surveillance asset cannot detect targets while it is en route to the patrol box. Constraints (18) and (19) ensure that no more than one case is allocated to an asset at one time. Constraint (20) indicates that the maximum asset aloft time must not exceed $L_i$. Constraint (21) ensures that there must be a minimum downtime of $R_i$ between asset allocations for a particular asset $i$ and that subsequent allocations must have a departure time later than the previous one(s), if any. The problem posed in (17)–(23) is NP-hard [36].

## III. SOLUTION APPROACH

### A. Exhaustive Branch-and-Cut

The first solution approach we investigated is the exhaustive B&C method, herein referred to as E-B&C. This method involves the enumeration and evaluation of all feasible solutions and is illustrated in Fig. 4 with respect to asset $i$, where each completed branch is a feasible solution with the corresponding asset–case assignments $\langle i, j^* \rangle$, given the departure times $\tau_{i\ell}$, $\ell = 1, 2, \ldots$, for each flight of asset $i$. We enumerate a complete feasible flight schedule over all flights for each asset $i$ and calculate the total reward $r_i$ for a given asset $i$ using (15) and (16). The schedule with the highest $r_i$ is selected to

[1]The geodesic distance is the shortest distance between two points on the surface of a sphere.

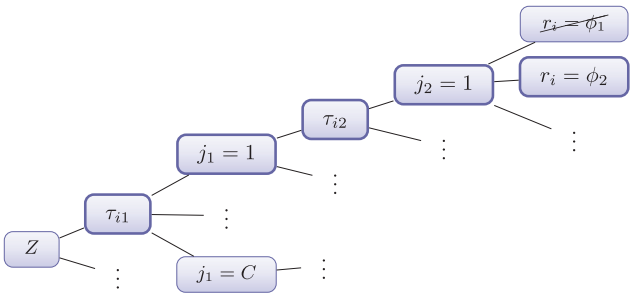



Fig. 4. Branching method with $\tau_{i1}$ and $\tau_{i2}$ being the departure time for the first and second flights and the corresponding case assignments $j_1$ and $j_2$. $r_i$ is evaluated using (15) and (16) for each completed branch. The highest $r_i$ is then saved as the best assignment for asset $i$.

be the best assignment for asset $i$. In order to find the optimal allocation, we repeat the process mentioned earlier with the full permutation of asset–case combinations. The pseudocode is shown in Algorithm 1. In Algorithm 1's pseudocode, line 1 generates the permutation of the ordering of assets for which to start the allocation. Lines 2–5 compute the best assignment for the selected asset $i$ using B&C and updates the PoA surface accordingly to avoid duplicate assignments (this is done by setting the allocated grid cells in the PoA surface to have no reward during the assigned search time(s)). Line 7 saves all the assignment for each asset sequence generated by the permutation function. Line 8 resets the PoA surface to the originally initialized surface prior to any updates in order to compute the next sequence generated by the permutation function.

ALGORITHM 1 Exhaustive Branch-and-Cut (E-B&C)

1: PermSeq = Perm$(1, \ldots, A)$ ▷ Permutation of ordering of assets for which to start allocation
2: **for each** AssetSeq in PermSeq **do**
3: **for each** $i \in$ AssetSeq **do**
4: assign$(i)$ = B&C$(i)$
5: updatePoA(assign$(i)$) ▷ Prevent overlap of patrol box assignments
6: **end for**
7: PotentialAssign ← PotentialAssign + assign ▷ Save potential assignment given we allocated in order AssetSequence
8: resetPoA ▷ Set PoA to originally initialized surface prior to any updates
9: **end for**
10: BestAssign = MaxReward(PotentialAssign) ▷ For all potential assignments found, search and find that which resulted in the maximum reward

## B. Greedy Branch-and-Cut I

Similar to E-B&C, we repeat the asset allocation process for all the available assets and fix the assignment for an asset $i^*$ with the highest $r_i$. After the asset–case–time epoch assignment is fixed, we update the PoA surface to ensure that the assigned cases are no longer available for additional scheduling during the assigned search hours. The same process is then repeated until either no more assets are available or all cases are fully allocated. We refer to this method as GB&C-I. The pseudocode is shown in Algorithm 2. In this pseudocode, line 1 states that while there are any unassigned assets, continue on to lines 2–7, where the best assignment for each unassigned asset is found using B&C. The best asset assignment is then selected in line 8 (i.e., $i^*$ becomes known among the explored potential assignments). In lines 9–11, the PoA surface is updated given the asset assignment found.

ALGORITHM 2 Greedy Branch-and-Cut I (GB&C-I)

1: **while** length(AssignedAsset) $\leq A$ **do**
2: assign = $\emptyset$
3: **for each** $i \in \{1, \ldots, A\}$ **do**
4: **if** $i \notin$ AssignedAsset **then**
5: assign$(i)$ = B&C$(i)$
6: **end if**
7: **end for**
8: assignment$(i^*)$ = MaxReward(assign)
9: AssignedAsset ← AssignedAsset + $i^*$
10: BestAssign ← BestAssign + assignment$(i^*)$
11: updatePoA(assignment$(i^*)$)
12: **end while**

## C. Greedy Branch-and-Cut II

To reduce the runtime and problem complexity, we propose a second greedy B&C method, referred to as GB&C-II. This method is similar to the E-B&C method, except that we put an additional constraint on assets. Once we enumerate all the possible departure times and find the best assignment $\{j^*\}$ corresponding to each departure time for an asset $i$, we fix the corresponding schedule. That is, we reduce the complexity of search with more than one asset from permutation ordering to a linear ordering. The same process is then repeated until all cases are fully allocated or there are no more assets available. The pseudocode is shown in Algorithm 3. Here, line 2 finds the best assignment for asset $i$ found in line 1. Line 3 updates the PoA surface and line 4 saves the best assignment found in line 2.

ALGORITHM 3 Greedy Branch-and-Cut II (GB&C-II)

1: **for each** $i \in \{1, \ldots, A\}$ **do**
2: assign$(i)$ = B&C$(i)$
3: updatePoA(assignment$(i)$)
4: BestAssign ← BestAssign + assign$(i)$
5: **end for**

### D. Parallelized Greedy Branch-and-Cut II

To further improve the computation time, we develop a parallelized version of the GB&C-II algorithm. Parallelization involves dividing a large problem into multiple independent subproblems, where each subproblem is assigned to a processor. This substantially reduces the computation time and therefore can rapidly generate allocation solutions. We use a master–slave architecture for our parallelization with the following functionalities:

**Master process**

- Pools subproblems for the slave processors to run.
- Spawns the subproblems on multiple slave processors.
- Collects the results from the slave processors.

**Slave process**

- Receives the subproblem from the master processor.
- Executes the subproblem.
- Returns the solution to the master processor.

The serial GB&C-II algorithm, executed on a single processor, searches the B&C tree by expanding live nodes one at a time. In order to parallelize this problem on $M$ processors, we set each $\tau_{i1}$ to each processor and let each processor execute the subproblem. All processes share the same memory for the PoA and other read-only data. Lastly, the master processor collects all value returns from the slave processors to evaluate the best assignment for asset $i$.

### E. Approximate Dynamic Programming

Another approach to solve the problem is via ADP, more specifically, a one-step lookahead rollout algorithm. Note that the following formulation is for a single arbitrary asset $i$ and is thus assumed given throughout. Let $j_k$ be the asset–case assignment at time epoch $k$ and $z_k$ be the remaining aloft time for an asset at time epoch $k$. We have the state equation for $z_{k+1}$ as

$$z_{k+1} = f(z_k, j_k), \tag{24}$$

where $j_k$ is the state-based control variable that selects a case $j$ at time epoch $k$ as

$$j_k = \mu_k(z_k, j_{k-1}), \qquad j_k = 0, 1, \ldots, C. \tag{25}$$

Here, $z_k = L_i$ and $j_k = 0$ implies that there is no asset–case assignment made and the asset is in the rest state at time epoch $k$. When $z_k \leq L_i$ and $j_k = 1, \ldots, C$, an asset–case assignment has been made at time epoch $k-1$ and the asset is currently in a flight state. The detailed control options are described in this section later (see (31) and (32)).

The ADP equation for the problem is defined as follows:

$$g_k(z_k, j_k) = \lambda_{j_k}\left(1 - \prod_{k \in s_{j_k} | z_k} \left(1 - w_{ij_k k}\right)\right), \tag{26}$$

$$J_k(j_k) = \max_{j_k} E\left\{g_k(z_k, j_k) + \bar{J}_{k+1}\left(f(z_k, j_k, \Lambda(k))\right)\right\}, \tag{27}$$

where $s_{j_k}$ is the set of remaining search time indices available within the current sortie for asset $i$ assigned to case $j$ and $\Lambda(k)$ is a function that indicates that the asset is currently flying its $\ell$th flight at time $k$. The variable $\lambda_j$ is the normalized priority weight for case $j$. Here, $\bar{J}_{k+1}$ is the heuristic cost-to-go and is estimated based on the following assumptions:

- H1: The asset will fly out for its maximum aloft time.
- H2: Each asset will stay on just one case for each flight.
- H3: Each asset will fly out immediately after it is fully rested.
- H4: The case with the highest total reward will be selected for the $\ell$th flight interval, as in (28):

$$j^* = \arg\max_j \lambda_j \left(1 - \prod_{k \in \gamma(i,j,\ell)} \left(1 - w_{ijk}\right)\right), \tag{28}$$

where $\gamma(i, j, \ell)$ is the set of search time indices for asset $i$ assigned to case $j$ for the $\ell$th flight. If the planning time horizon allows multiple flights, then we first compute the best case for the next flight time defined by H1 to H3 using (28). The future cost-to-go for the $\ell$th flight is as follows:

$$H(\ell) = \lambda_{j^*}\left(1 - \prod_{k \in \gamma(i,j^*,\ell)} \left(1 - w_{ij^*k}\right)\right), \tag{29}$$

where $j^*$ is computed from (28). The heuristic cost-to-go given the current flight at time $k$ is $\Lambda(k)$ and is given by

$$\bar{J}_{k+1}\left(f(z_k, j_k, \ell_k)\right) = \sum_{n=\Lambda(k)+1}^{\lceil K/(L_i+R_i)\rceil} H(n). \tag{30}$$

As mentioned earlier, the control variable $j_k$ is state-dependent. When an asset is at rest state at time $k$, the control variable $j_k$ comprises the actions of launching the asset or not with the intent of obtaining better reward at a later time epoch. That is,

$$j_k = \begin{cases} 0, & \text{do not launch the asset,} \\ \beta, \quad \beta = 1, \ldots, C, & \text{launch the asset.} \end{cases} \tag{31}$$

A comparison of expected reward between launching the asset at the current hour versus the next hour is

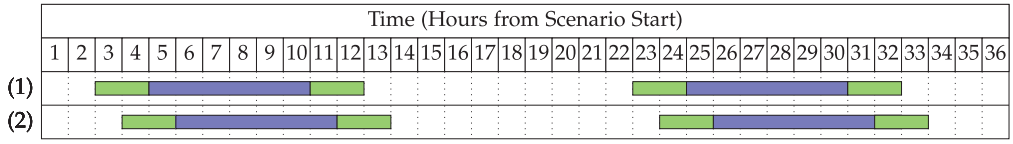



Fig. 5. Illustration of rollout for deciding when to fly with the traveling time (green) and search time (blue).

performed using rollout with the heuristic defined earlier. If launching the asset during the current time epoch results in a higher reward, then the asset will be assigned to the case with the highest total reward $r_i$ in (28) and assigned for the first search hour to the selected case. If launching the asset during the next time epoch results in a higher reward, then we simply increment the time epoch and repeat the process. Fig. 5 illustrates this rollout heuristic for determining the expected reward for launching at a different hour.

When the asset is in flight, for the second through final hour of the search, the control variable $j_k$ takes on a different set of values, detailed as follows:

$$j_k = \begin{cases} j_{k-1}, & \text{stay on the current case,} \\ \tilde{j} \neq j_{k-1}, & \text{switch to a different case with the cost of additional travel time.} \end{cases} \tag{32}$$

We illustrate the computation of the heuristic for the one-step lookahead rollout in Fig. 6. The first example illustrates the situation when the surveillance asset is searching for case $j$ and chooses to stay on case $j$ for the remaining search interval. The second example illustrates the situation, wherein the asset currently searching for case $j$ switches to a new case $\tilde{j} \neq j$, while considering the cost of additional travel time from case $j$ to case $\tilde{j}$. The travel time between the new case $\tilde{j}$ to the asset's home base is then the new return travel time for the asset. The optimal control action is selected based on the maximum expected reward, as in (27). This process is repeated for each time epoch $k$ to obtain a feasible asset–case assignment over the planning horizon.

### F. Multistep Lookahead Approximate Dynamic Programming I

We propose two multistep lookahead ADP strategies to obtain near-optimal assignments for all assets. The first method begins with an $m$-length permutation of the asset order for which to start the allocation. That is, $m = 1$ corresponds to searching over each asset; $m = 2$ corresponds to searching over all possible pairs of assets; and so on. The PoA is then updated with respect to each asset–case–time assignment to ensure that there are no duplicate asset–case–time tuples. The difference between the two methods lies in how the remaining assets are allocated. In the first proposed method, we exhaustively compute the feasible asset assignment for all the available assets and fix the allocation corresponding to the asset with the highest $r_i$. The PoA is then updated and the process is then repeated until either no more assets are available or all cases are fully allocated. Once all the assets are allocated, we reset the PoA surface to its original state and repeat the process from the beginning with the next possible $m$-length subset of assets to start the initial asset assignment over the time horizon. We refer to this method as $m$SLADP-I. The pseudocode is shown in Algorithm 4. In Algorithm 4's pseudocode, line 1 generates the $m$-length permutation of asset order, where $m$ specifies the size of the subset permutation to be used for the initial asset allocation. Lines 4–7 find the best allocation given each asset $i$ in a specific asset order from line 1 and update the PoA surface, accordingly. Then, lines 12–18 compute the best assignment for the remaining unassigned assets. Line 12 finds the best assignment for each unassigned asset and line 17 selects the best asset $i^*$ for allocation. The PoA surface is subsequently updated in line 18. Lines 20–24 save the complete assignment and reset the parameters for the next asset permutation sequence generated in line 1.

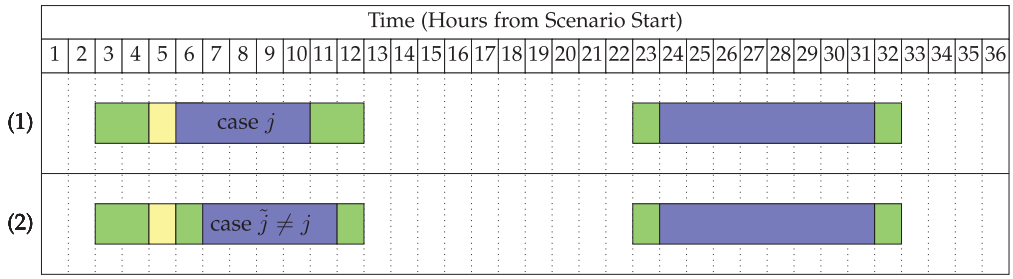



Fig. 6. Illustration of one-step lookahead. 1) Stay at current case; 2) switch to a different case with the cost of additional traveling time; and 3) return to the asset's base.

**Algorithm 4** *m*SLADP-I

1: PermSeq = Perm($\{1, \ldots, A\}, m$) ▷ $m$-length permutation of asset order, where $m$ specifies the *size* of the subset permutation to be used for initial asset allocation
2: **for each** AssetSeq in PermSeq **do**
3: **for each** $i \in$ AssetSeq **do**
4: assign($i$) = ADP($i$)
5: BestAssign ← BestAssign + assign($i$)
6: updatePoA(assign($i$))
7: AssignedAsset ← AssignedAsset + $i$
8: **end for**
9: **while** length(AssignedAsset) ≤ $A$ **do**
10: **for each** $i \in \{1, \ldots, A\}$ **do**
11: **if** $i \notin$ AssignedAsset **then**
12: assignTemp($i$) = ADP($i$)
13: **end if**
14: **end for**
15: b_assign($i^*$) = MaxReward(assignTemp) ▷ Given the previous allocations, select the asset assignment with the highest $r_i$ among the remaining available assets
16: AssignedAsset ← AssignedAsset + $i^*$
17: BestAssign ← BestAssign + b_assign($i^*$)
18: updatePoA(assignment($i^*$))
19: **end while**
20: PotentialAssign ← PotentialAssign + BestAssign
21: assign = ∅
22: AssignedAsset = ∅
23: BestAssign = ∅
24: resetPoA
25: **end for**
26: BestAssign = MaxReward(PotentialAssign)

### G. Multistep Lookahead Approximate Dynamic Programming II

The second multistep lookahead method (referred to as *m*SLADP-II), as in the first method, begins with an $m$-length permutation of asset ordering. The difference between *m*SLADP-I and *m*SLADP-II is how the algorithm computes the asset allocation for the remaining assets. In *m*SLADP-II, we iteratively compute the best asset allocation for each $i$. Once the best assignment is found for asset $i$, we immediately fix the corresponding schedule and update the PoA surface. There is no additional loop to find the best asset–case assignment among all the remaining assets. Hence, *m*SLADP-II is faster and less complex than *m*SLADP-I. The same process is then repeated until all cases are fully allocated or there are no more assets available. The pseudocode is shown in Algorithm 5. In this pseudocode, lines 1–7 remain the same as Algorithm 4. The difference lies in lines 10–17, where, for each unassigned asset, we find the best assignment corresponding to an asset $i^*$ and update the PoA surface accordingly. Once all assets are assigned, we reset all the parameters for the next asset permutation sequence generated from line 1.

**Algorithm 5** *m*SLADP-II

1: PermSeq = Perm($\{1, \ldots, A\}, m$)
2: **for each** AssetSeq in PermSeq **do**
3: **for each** $i \in$ AssetSeq **do**
4: assign($i$) = ADP($i$)
5: BestAssign ← BestAssign + assign($i$)
6: updatePoA(assign($i$))
7: AssignedAsset ← AssignedAsset + $i$
8: **end for**
9: **for each** $i \in$ {1,..., A} **do**
10: **if** $i \notin$ AssignedAsset **then**
11: assign($i$) = ADP($i$)
12: **end if**
13: **end for**
14: AssignedAsset ← AssignedAsset + $i^*$
15: BestAssign ← BestAssign + assign($i$)
16: updatePoA(assignment($i^*$))
17: PotentialAssign ← PotentialAssign + BestAssign
18: assign = ∅
19: AssignedAsset = ∅
20: BestAssign = ∅
21: resetPoA
22: **end for**
23: BestAssign = MaxReward(PotentialAssign)

## IV. SIMULATION AND COMPUTATIONAL RESULTS

The proposed algorithms were implemented in Python 2.7 on an Intel® Core™ i7-6600U CPU @ 2.60 GHz × 4 with 32 GB RAM. Our computational results are organized as follows: We first describe the mission scenario. Then, we discuss the solution quality of various algorithms with respect to objectives O1–O3 and their runtimes. Additionally, we conduct scalability analyses of the algorithms by varying the number of assets and cases, as well as robustness of the various algorithms using a signal-to-noise ratio (SNR) metric from robust design [37].

### A. Scenario Description(s)

There are two main areas of operation in the simulated scenario: the East Pacific Ocean and the Caribbean Sea. The PoA surfaces corresponding to this area of responsibility (AOR) were partitioned into a grid of 90 × 138 cells, where each cell is a square with a side length of 30 nautical miles. The total area of the AOR was ≈11 million square nautical miles. The lower left corner of the rectangular AOR had a latitude and longitude of 10°S and 110°W, respectively.

TABLE II
Smuggler Cases

| Case | Case ID | Vessel type | Speed (kts) | Payload (kg) | No. of smugglers |
|---|---|---|---|---|---|
| 1 | GF1 | Go fast | 30 | 1000 | 3 |
| 2 | PG1 | Panga | 20 | 450 | 2 |
| 3 | GF2 | Go fast | 30 | 1000 | 3 |
| 4 | PG2 | Panga | 20 | 450 | 2 |
| 5 | PG3 | Panga | 20 | 450 | 2 |
| 6 | PG4 | Panga | 20 | 450 | 2 |
| 7 | SP1 | SPSS[a] | 8 | 2500 | 3 |
| 8 | FSV1 | FSV[b] | 4 | 5000 | 2 |
| 9 | PG5 | Panga | 20 | 450 | 2 |
| 10 | PG6 | Panga | 20 | 450 | 2 |

[a]Self-propelled semi-submersible.

[b]Fully submerged vessel.

The PoA surfaces forecasted 10 smuggler cases, of which 5 were located in the East Pacific Ocean and the remaining 5 were located in the Caribbean Sea. The details for each case can be found in Table II and Fig. 7. These cases are generated based on Navy intelligence, which typically comprises estimates of the expected number of smugglers on board and the size of the contraband shipment. Often there are few "active" cases, i.e., cases that targeteers deem to have sufficient actionable intelligence to allocate assets to. We assume the PoA surfaces reflect the spatiotemporal probabilities pertaining to such "active" cases. Four different types of smuggler vessels were considered: 1) Go fast—small, fast boats capable of reaching high speeds; 2) Panga—modest-sized, fast boats that are easy to build by the smugglers; 3) self-propelled semi-submersible (SPSS)—narco-submarines capable of shifting heavy loads long distances while almost submerged under the ocean's surface [38]; and 4) fully submerged vessel—makeshift submarine-like vessels that can remain submerged with large quantities of cocaine aboard. Each case had a unique departure, destination, and waypoint combination. Waypoints are defined as possible areas in the ocean where the cargo is transferred to another vessel or a change in trajectory of the smuggler is predicted. Additionally, each case also had an associated payload measured in kg of cocaine. This is relevant when we run the algorithm with objective O2. An important fact to note is that each case had different start and end times. Fig. 8 details the time epochs when each smuggler case is deemed active. Cases with high uncertainty had wide bands of PoA. The amount of uncertainty is dependent on the type of smuggler vessel (e.g., SPSSs can be extremely difficult to detect, and thus the corresponding PoA surfaces reflect this in long and broad bands of probability reflecting spatial and temporal uncertainty) and/or departure time uncertainty.

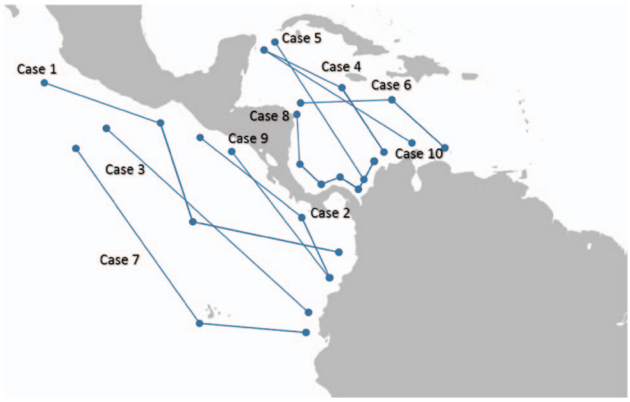



Fig. 7. Experiment scenario.

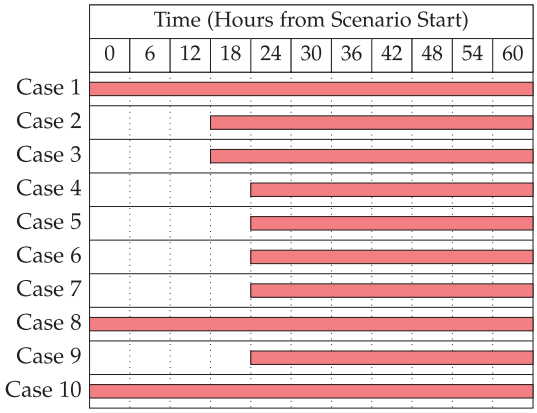



Fig. 8. Chart displaying when each smuggler case is active over the 72 h time horizon. Cases are active up through time $K = 72$ and do not necessarily end at that time, but rather, due to the time horizon of the forecast data, are truncated.

In the scenario, 10 P-3 surveillance assets were considered as available for allocation during the planning horizon. The home bases of individual surveillance assets are detailed in Table III. Each asset carries two different types of sensors with performance parameters detailed in Table IV.

We simulated the scenario with a granularity of 1 h (i.e., the forecasted surfaces were for each hour, on the hour; thus $\Delta = 1$ h). The forecasts extended to 72 h out from the current time (i.e., $K = 72$) and an asset allocation solution (e.g., $x_{ijk} = 0$ or 1) was required for each time epoch, $k$, in order for the algorithm to terminate.

Note that we omit E-B&C for large-sized scenarios in our results due to an exponential increase in computation times. Therefore, for E-B&C, we compute the solution for scenarios involving only up to 5 assets and 10 cases.

TABLE III
Asset Home Base Location (Longitude, Latitude)

| | |
|---|---|
| 1, 6 | (−69.7617, 18.5036) |
| 2, 7 | (−79.3833, 9.07111) |
| 3, 8 | (−85.5442, 10.5931) |
| 4, 9 | (−89.0558, 13.4406) |
| 5, 10 | (−92.37, 14.7942) |

TABLE IV
Sensor-to-Target Sweep Width (nm)

| Sensor type | FSV[a] | SPSS[b] | FV[c] | Panga | GF[d] | MV[e] | SV[f] | UNK[g] |
|---|---|---|---|---|---|---|---|---|
| APS 115 | 5 | 7.5 | 7.5 | 9 | 7.5 | 7.5 | 7.5 | 2.5 |
| APS 137 | 10 | 15 | 15 | 18 | 15 | 15 | 15 | 5 |

[a]Fully submerged vessel.
[b]Self-propelled semi-submersible.
[c]Fishing vessel.
[d]Go fast.
[e]Merchant vessel.
[f]Supply vessel.
[g]Unknown (other).

## B. Solution Quality with Different Objective Functions

Using the aforementioned values for the parameters, we ran the simulation for all the approaches to schedule the 10 specified assets over the 72 h planning horizon. Tables V–VII show the CPoSD for the GB&C-II method for objectives O1, O2, and O3, respectively. Parallel GB&C-II has the same result as the sequential GB&C-II. Therefore, we omit the parallel GB&C-II from the quality comparison.

We refer to Tables V–VII as COA matrices [5]. The COA matrices aid the DM in understanding the reasoning behind the algorithm's behavior and its output by giving metrics for both individual asset–case pairs and, overall, the probability an asset detects at least one case (PDC) and the probability that a case is detected by at least one asset (PDA). These matrices may be generated to assess the allocation performance at a particular time epoch, or, as shown in Tables V–VII, the cumulative asset allocation performance up to that point in time (in Tables V–VII, through $K = 72$).

Solving with respect to objective O1 (Table V) resulted in an asset allocation with the highest expected weight of contraband detected, totaling 7828 kg of cocaine compared to objectives O2 and O3 (Tables VI and VII). This implies that we have a 64% success rate of detecting the transport of contraband with respect to the total possible for the experimental scenario of 12,200 kg of contraband. The asset allocations with respect to objective O1 have 15.5% and 10.1% more contraband disrupted when compared to objectives O2 and O3, respectively. In Table V, case 8 has the most amount of contraband (5000 kg) with a CPoSD = 0.95. Solving with respect to objective O3 resulted in the detection of a higher expected weight of contraband (5.9%), expected number of detections (6.8%), and expected number of smugglers (7.5%) compared to objective O2. This could be caused by the uniform priority weight vector used in objective O2.

For the sake of compactness, we omit the COA matrices used in demonstrating the performance of the other approaches implemented and, instead, quantify the goodness of the allocation by comparing the algorithms with that of GB&C-II algorithm as measured by the expected weight of the contraband detected, expected number of detections, and expected number of smugglers detected.

The sums of the totals for each objective for each algorithm are shown in Table VIII. Fig. 9 shows a normalized representation of the results detailed in Table VIII, where the largest possible number of detections and contraband detected was utilized as a basis for normalization of both metrics, respectively, to compare the expected number of detections and contraband weight detected. Note that Fig. 9 only contains the results for 1SLADP-I and 1SLADP-II; the detailed solutions of $m$SLADP with $m > 1$ are shown later in Section IV-C.

We illustrate in Table VIII and Fig. 9 that all B&C-based algorithms optimizing objective O2 are outperformed by the same algorithms optimizing objective O3 in terms of both the expected number of detections and expected number of smugglers. When comparing GB&C-I and GB&C-II, optimizing with respect to objective O2 resulted in 4% less expected number of detec-

TABLE V
Objective O1: Maximize Weight of Contraband Detected

| Asset | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 | Case 7 | Case 8 | Case 9 | Case 10 | PDC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | – | – | – | – | – | – | – | 0.76 | – | – | 0.76 |
| 2 | – | – | – | – | – | – | 0.43 | 0.29 | – | – | 0.60 |
| 3 | 0.09 | – | – | – | – | – | 0.25 | 0.29 | – | – | 052 |
| 4 | 0.15 | – | – | – | – | 0.28 | – | – | – | 0.14 | 0.47 |
| 5 | 0.21 | 0.14 | – | – | – | – | – | – | – | – | 0.33 |
| 6 | – | – | – | – | – | – | – | 0.61 | – | – | 0.61 |
| 7 | – | – | – | – | – | – | 0.40 | – | – | – | 0.40 |
| 8 | – | – | 0.09 | — | – | – | 0.21 | – | – | – | 0.28 |
| 9 | 0.17 | – | – | – | – | 0.30 | – | – | – | – | 0.42 |
| 10 | – | – | – | 0.20 | – | 0.11 | – | – | – | 0.06 | 0.33 |
| PDA | 0.49 | 0.14 | 0.09 | 0.20 | – | 0.55 | 0.80 | 0.95 | – | 0.19 | [a,b,c] |

[a]Expected weight of contraband disrupted: 7828 kg.
[b]Expected number of detections: 3.41.
[c]Expected number of smugglers: 8.21.

TABLE VI
Objective O2: Maximize Number of Detections

| Asset | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 | Case 7 | Case 8 | Case 9 | Case 10 | PDC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | – | – | – | – | – | – | – | 0.76 | – | – | 0.76 |
| 2 | – | 0.27 | – | – | – | 0.38 | – | – | – | 0.15 | 0.61 |
| 3 | 0.20 | – | – | – | – | – | 0.28 | – | – | – | 0.43 |
| 4 | 0.15 | – | – | – | – | – | – | 0.10 | – | 0.15 | 0.35 |
| 5 | 0.19 | – | – | 0.17 | – | – | – | – | – | – | 0.33 |
| 6 | – | – | – | 0.13 | – | – | – | – | – | 0.20 | 0.31 |
| 7 | – | 0.29 | – | – | – | – | – | 0.37 | – | – | 0.55 |
| 8 | 0.13 | – | – | – | – | – | – | – | 0.23 | – | 0.33 |
| 9 | – | – | 0.10 | – | 0.18 | – | – | – | – | – | 0.25 |
| 10 | 0.17 | – | – | – | – | – | – | – | – | 0.05 | 0.21 |
| PDA | 0.61 | 0.48 | 0.10 | 0.28 | 0.18 | 0.38 | 0.28 | 0.86 | 0.23 | 0.46 | [a,b,c] |

[a]Expected weight of contraband detected: 6619 kg.
[b]Expected number of detections: 3.85.
[c]Expected number of smugglers: 8.67.

TABLE VII
Objective O3: Maximize Number of Smugglers Detected

| Asset | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 | Case 7 | Case 8 | Case 9 | Case 10 | PDC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | – | – | – | – | – | 0.21 | – | 0.61 | – | – | 0.69 |
| 2 | – | – | – | – | – | 0.34 | 0.26 | – | – | 0.15 | 0.59 |
| 3 | 0.09 | 0.23 | – | – | – | – | 0.25 | – | – | – | 0.48 |
| 4 | 0.11 | – | – | 0.21 | – | – | 0.10 | – | – | – | 0.37 |
| 5 | 0.18 | – | 0.10 | – | – | – | – | – | – | – | 0.25 |
| 6 | – | – | – | – | – | – | – | 0.58 | – | 0.15 | 0.64 |
| 7 | – | 0.27 | – | – | – | – | – | – | – | 0.11 | 0.35 |
| 8 | – | – | – | – | 0.18 | – | – | – | 0.22 | – | 0.36 |
| 9 | 0.16 | – | – | – | – | 0.30 | – | – | – | – | 0.41 |
| 10 | – | – | 0.09 | 0.16 | – | – | – | – | – | – | 0.23 |
| PDA | 0.44 | 0.44 | 0.17 | 0.34 | 0.18 | 0.63 | 0.51 | 0.84 | 0.22 | 0.35 | [a,b,c] |

[a]Expected weight of contraband detected: 7036 kg.
[b]Expected number of detections: 4.13.
[c]Expected number of smugglers: 9.37.

tions and 1.2% less expected number of smugglers than when optimizing with respect to objective O3.

In terms of the amount of contraband detected, using the GB&C-I algorithm resulted in an allocation that obtained the highest expected amount of contraband detected when solving for objective O1; however, its solutions for maximizing the expected number of detections or expected number of smugglers were inferior to the ADP-based algorithms. In general, we see that the B&C-based methods are able to obtain more contraband when

TABLE VIII
Algorithm comparison

| Contraband disrupted (kg) | | | | |
|---|---|---|---|---|
| Objective | GB&C-I | GB&C-II | 1SLADP-I | 1SLADP-II |
| O1 | 7869 | 7828 | 7520 | 7821 |
| O2 | 6658 | 6619 | 7185 | 7610 |
| O3 | 7188 | 7036 | 7594 | 7591 |
| No. of detections | | | | |
| Objective | GB&C-I | GB&C-II | 1SLADP-I | 1SLADP-II |
| O1 | 3.47 | 3.41 | 3.57 | 3.68 |
| O2 | 3.87 | 3.84 | 3.72 | 4.08 |
| O3 | 4.12 | 4.13 | 3.80 | 4.04 |
| No. of smugglers | | | | |
| Objective | GB&C-I | GB&C-II | 1SLADP-I | 1SLADP-II |
| O1 | 8.35 | 8.21 | 8.37 | 8.72 |
| O2 | 8.56 | 8.67 | 8.40 | 9.21 |
| O3 | 9.50 | 9.37 | 8.70 | 9.12 |

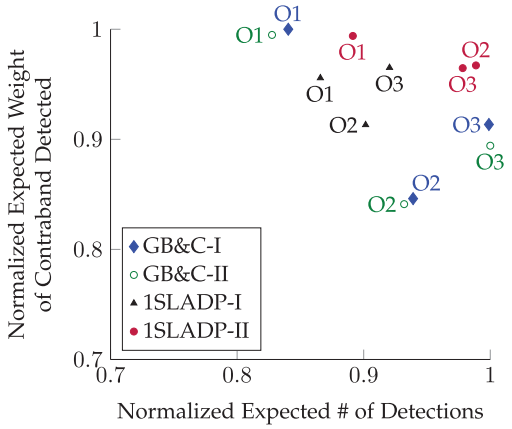



Fig. 9. A normalized view comparing the performance of all the algorithms, with respect to the expected weight of contraband disrupted (O1), the expected number of interdictions (O2), and the expected number of smugglers (O3).

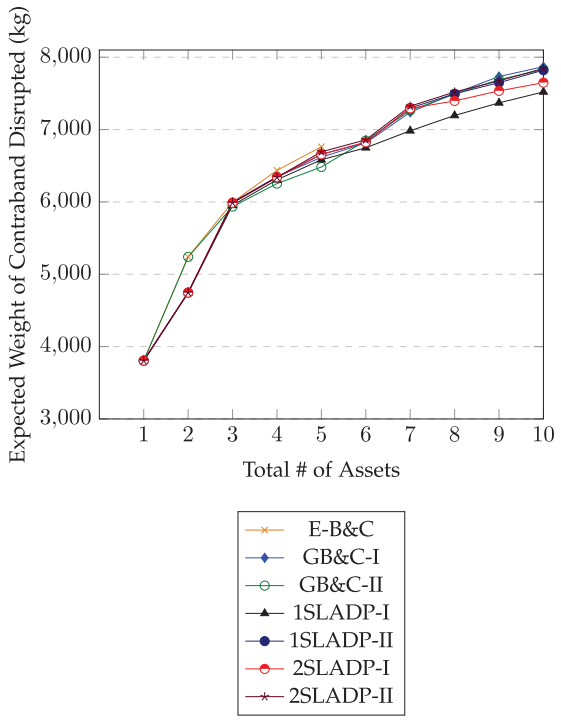



Fig. 10. The expected weight of contraband disrupted for each algorithm by varying the number of available assets.

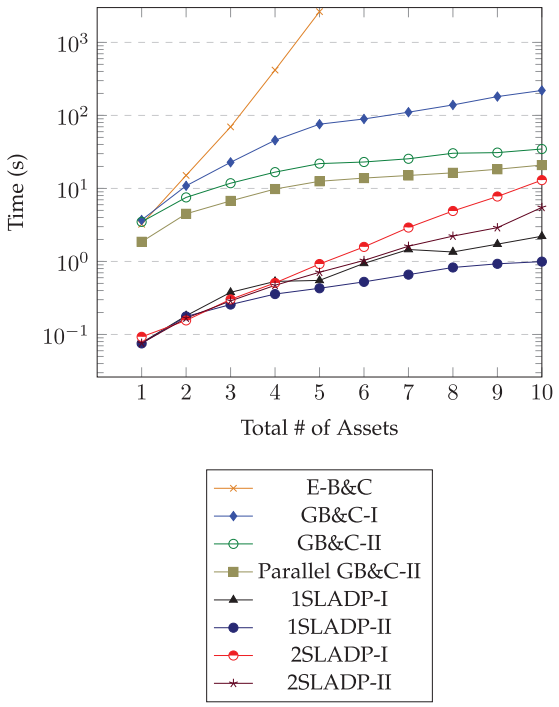



Fig. 11. The CPU runtimes for each algorithm by varying the number of available assets.

solving with respect to objective O1, while the ADP-based methods are able to get better solutions for the expected number of detections and expected number of smugglers when solving with respect to objectives O2 and O3 with the exception of 1SLADP-I for objective O3.

### C. Scalability: Available Asset Sensitivity

In this section, we use objective O1 for the scalability studies with respect to the number of assets. To measure the scalability, we limited the number of assets available for allocation for the scenario from 1 to 10 aircraft. Figs. 10 and 11 show the expected weight of contraband disrupted and the runtimes, respectively. The detailed values are given in Tables IX and X. In Fig. 10 and Table IX, we see that ADP-based algorithms (1SLADP-I, 1SLADP-II, 2SLADP-I, and 2SLADP-II) are able to obtain similar amounts of contraband disrupted, differing by only up to 339.7 kg (4.6%) of contraband.

Similarly, the B&C-based algorithms (E-B&C, GB&C-I, and GB&C-II) are able to obtain similar amounts of contraband disrupted, differing by only up to 279.1 kg of contraband among the three. E-B&C, intuitively, outperformed the other B&C variations (and all other algorithms for that matter) among the scenarios simulated until runtime became an issue. GB&C-II is able to obtain a better result compared to GB&C-I when there are two, six, or seven assets available for allocation. This is due to the nature of the scenario or the characteristics of the smuggler cases. Since GB&C-I iterates through all available assets, there is a tendency that closer (with respect to assets' home base) cases

TABLE IX
Expected Weight of Contraband Disrupted (kg) for Varying Asset Availability

| No. of assets | E-B&C | GB&C-I | GB&C-II | 1SLADP-I | 1SLADP-II | 2SLADP-I | 2SLADP-II |
|---|---|---|---|---|---|---|---|
| 1 | 3806 | 3806 | 3806 | 3806 | 3806 | 3806 | 3806 |
| 2 | 5240 | 4747 | 5240 | 4747 | 4747 | 4747 | 4747 |
| 3 | 5997 | 5997 | 5935 | 5952 | 5988 | 5988 | 5988 |
| 4 | 6436 | 6349 | 6253 | 6306 | 6341 | 6341 | 6341 |
| 5 | 6762 | 6617 | 6483 | 6583 | 6658 | 6658 | 6693 |
| 6 | – | 6816 | 6847 | 6748 | 6823 | 6823 | 6858 |
| 7 | – | 7239 | 7265 | 6983 | 7298 | 7292 | 7323 |
| 8 | – | 7501 | 7490 | 7195 | 7494 | 7396 | 7519 |
| 9 | – | 7734 | 7691 | 7369 | 7648 | 7535 | 7673 |
| 10 | – | 7869 | 7828 | 7520 | 7821 | 7648 | 7846 |

TABLE X
Simulation Runtime (s) for Varying Asset Availability

| No. of assets | E-B&C | GB&C-I | GB&C-II | Parallel GB&C-II | 1SLADP-I | 1SLADP-II | 2SLADP-I | 2SLADP-II |
|---|---|---|---|---|---|---|---|---|
| 1 | 3.23 | 3.70 | 3.48 | 1.86 | 0.08 | 0.08 | 0.09 | 0.08 |
| 2 | 15.1 | 10.9 | 7.54 | 4.49 | 0.18 | 0.18 | 0.16 | 0.17 |
| 3 | 69.6 | 22.8 | 11.8 | 6.75 | 0.38 | 0.26 | 0.30 | 0.29 |
| 4 | 418 | 45.7 | 16.7 | 9.82 | 0.53 | 0.36 | 0.51 | 0.47 |
| 5 | 2639 | 75.9 | 21.9 | 12.5 | 0.55 | 0.43 | 0.92 | 0.71 |
| 6 | – | 89.4 | 23.0 | 13.9 | 0.94 | 0.53 | 1.58 | 1.03 |
| 7 | – | 111 | 25.5 | 15.1 | 1.46 | 0.66 | 2.91 | 1.62 |
| 8 | – | 139 | 30.3 | 16.4 | 1.35 | 0.83 | 4.91 | 2.22 |
| 9 | – | 181 | 31.0 | 18.32 | 1.73 | 0.93 | 7.75 | 2.90 |
| 10 | – | 220 | 34.7 | 20.9 | 2.22 | 0.99 | 13.0 | 5.50 |

are allocated first, since there is less travel time and, hence, are more rewarding. In turn, this may limit the options available to assets considered for allocation in later iterations since cases, previously in close proximity to their home base, may already be allocated and, due to longer travel time, will be much less rewarding or not at all. Similar problems arose with 1SLADP-I algorithm, which obtains less expected contraband disrupted compared to 1SLADP-II algorithm when there are more than six assets available for allocation, differing by up to 314.9 kg of contraband. We are able to minimize the effect of this problem by applying a two-step lookahead strategy. 2SLADP-I algorithm obtains less expected contraband disrupted compared to 2SLADP-II algorithm when there are more than five assets available for allocation, differing by up to 197.6 kg of contraband.

As Fig. 11 and Table X show, E-B&C has the slowest runtime. There is a maximum speedup of 34.8, 120.6, 210.6, 4794, 6146, 2861, and 3711 and an average speedup of 9.8, 30.9, 53.7, 1177, 1542, 809, and 994 when comparing the runtimes of GB&C-I to GB&C-II, parallel GB&C-II, 1SLADP-I, 1SLADP-II, 2SLADP-I, and 2SLADP-II, respectively. Over all the asset availability scenarios tested, the average speedups of GB&C-II, parallel GB&C-II, 1SLADP-I, 1SLADP-II, 2SLADP-I, and 2SLADP-II are 3.6, 6.1, 87, 143, 52, and 72 times, respectively, faster compared to GB&C-I.

Our key finding here is that, with a 1.6% sacrifice in optimality on average, GB&C-II provides a solution nearly identical to that of E-B&C, while offering a solution in a fraction of the time (up to nearly 210.6 times faster among the simulated results). Alternatively, at a cost of 2.5% suboptimality on average, but more than 6146 times faster speedup, we can run 1SLADP-II for a given scenario. Similarly, at a cost of 2.4% suboptimality on average, 2SLADP-II offers more than 3711 times faster speedup.

In general, GB&C-II should be used when the total number of assets is less than 3 due to its minimal sacrifice in optimality (on average 1.6%). When the number of assets is greater than 3, 2SLADP-II should be used.

## D. Scalability: Varying the Number of Cases

Here, we vary the number of cases from 1 to 10, while fixing the number of available assets to 10. Figs. 12 and 13 show the expected weight of contraband disrupted and the runtimes, respectively. The detailed values for each figure are given in Tables XI and XII, respectively. From Fig. 12 and Table XI, we see that all the algorithms have very similar solution quality. We see a noticeable increase in contraband disruption for case 8 (5000 kg of contraband). All algorithms obtained a similar amount of expected contraband disrupted.

Fig. 13 and Table XII show the runtimes. As expected, GB&C-I has the slowest runtimes, while the 1SLADP-II algorithm has the fastest runtime (<1 s). There are maximum speedups of 7, 11, 99, 221, 17, and 40 when comparing the runtimes of GB&C-I to GB&C-II, parallel GB&C-II, 1SLADP-I, 1SLADP-II, 2SLADP-I, and

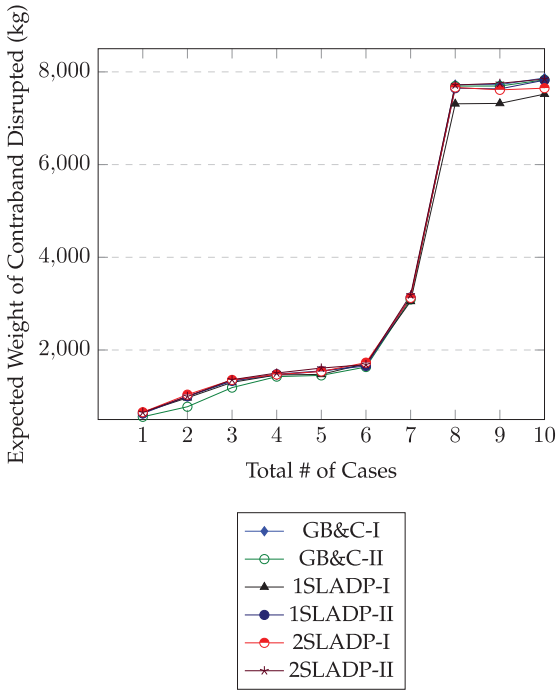



Fig. 12. The expected weight of contraband disrupted for each algorithm by varying the number of available cases.

TABLE XI
Expected Weight of Contraband Disrupted (kg) for Varying Case Availability

| No. of cases | GB&C-I | GB&C-II | 1SLADP-I | 1SLADP-II | 2SLADP-I | 2SLADP-II |
|---|---|---|---|---|---|---|
| 1 | 653.2 | 558.7 | 653.2 | 629.6 | 657.7 | 653.2 |
| 2 | 1000 | 775.8 | 968.7 | 999.4 | 1037 | 999.4 |
| 3 | 1332 | 1189 | 1302 | 1334 | 1355 | 1359 |
| 4 | 1459 | 1425 | 1462 | 1479 | 1471 | 1504 |
| 5 | 1540 | 1452 | 1474 | 1549 | 1537 | 1610 |
| 6 | 1680 | 1633 | 1707 | 1655 | 1727 | 1685 |
| 7 | 3145 | 3090 | 3049 | 3133 | 3125 | 3188 |
| 8 | 7725 | 7694 | 7311 | 7648 | 7665 | 7716 |
| 9 | 7725 | 7694 | 7320 | 7633 | 7612 | 7754 |
| 10 | 7869 | 7828 | 7520 | 7821 | 7648 | 7846 |

TABLE XII
Simulation Runtime (s) for Varying Case Availability

| No. of cases | GB&C-I | GB&C-II | Parallel GB&C-II | 1SLADP-I | 1SLADP-II | 2SLADP-I | 2SLADP-II |
|---|---|---|---|---|---|---|---|
| 1 | 0.85 | 0.34 | 1.08 | 0.68 | 0.23 | 1.99 | 0.86 |
| 2 | 4.09 | 1.36 | 4.47 | 0.66 | 0.32 | 3.74 | 1.27 |
| 3 | 5.88 | 2.67 | 1.89 | 0.85 | 0.42 | 4.13 | 1.58 |
| 4 | 11.8 | 3.93 | 3.06 | 1.00 | 0.54 | 5.24 | 1.91 |
| 5 | 18.0 | 4.77 | 4.01 | 1.48 | 0.57 | 5.51 | 2.16 |
| 6 | 24.7 | 7.48 | 5.31 | 1.34 | 0.64 | 6.95 | 2.38 |
| 7 | 56.0 | 11.7 | 7.27 | 1.45 | 0.72 | 7.79 | 2.55 |
| 8 | 108 | 13.2 | 8.56 | 1.67 | 0.80 | 9.45 | 3.13 |
| 9 | 183 | 15.9 | 9.85 | 1.92 | 0.89 | 11.9 | 4.00 |
| 10 | 220 | 34.7 | 20.9 | 2.22 | 0.99 | 13.0 | 5.50 |

2SLADP-II, respectively. On average, the speedups of the GB&C-II, parallel GB&C-II, 1SLADP-I, 1SLADP-II, 2SLADP-I, and 2SLADP-II algorithms were 4.3, 6, 33, 71.8, 6, and 16.5 times, respectively.

The key point here is that the algorithm 2SLADP-I is very efficient and is recommended for scenarios when the number of cases is less than or equal to the number of assets, which is often the case.

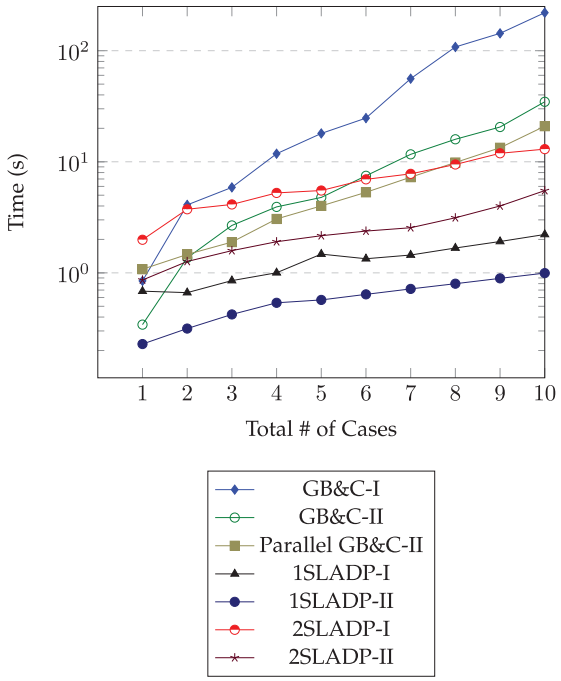



Fig. 13. The CPU runtimes for each algorithm by varying the number of available cases.

## E. Robustness: Monte Carlo Evaluation of Asset Allocation Strategies

To test the robustness of each asset allocation algorithm, we simulated 100,000 trajectories of smugglers (10,000 from each case) behaving as in our benchmark scenario. Sampling from the PoA surfaces, we obtained waypoints for each smuggler at each time epoch and joined them together to obtain a full path. From these paths, we measured whether the smuggler traversed through any allocated patrol boxes during the allocated search time, and if so, what was the aircraft's probability of detecting the target during those time epoch(s). Table XIII shows the detailed performance statistics for each algorithm over the 100,000 Monte Carlo simulations. A useful metric to measure an algorithm's goodness is that of nominal-the-best SNR [37], that is,

$$\text{SNR} = 10\log_{10}\frac{\mu^2}{\sigma^2}. \tag{33}$$

TABLE XIII
Monte Carlo Analysis (from 100,000 Runs)

| Objective | Mean of contraband detected ($\mu$) in kg | Standard deviation of contraband detected ($\sigma$) in kg | SNR (dB) |
|---|---|---|---|
| GB&C-I | 7616 | 246.3 | 29.8 |
| GB&C-II | 7632 | 252.5 | 29.6 |
| 1SLADP-I | 7645 | 244.2 | 29.9 |
| 1SLADP-II | 7610 | 218.4 | 30.8 |
| 2SLADP-I | 7648 | 240.3 | 30.1 |
| 2SLADP-II | 7612 | 203.3 | 31.5 |

Nominal-the-best SNR is a useful measure when the goal is to maximize a mean and minimize the variation. Note that maximization of this metric seeks to minimize the coefficient of variation (=standard deviation/mean) and is thus a measure of robustness of a solution. From the results of 100,000 Monte Carlo runs, we found that the algorithm 2SLADP-II performs the best with respect to objective O1, when measured using the SNR. The 2SLADP-II algorithm obtained, on average, 7612 kg of contraband (out of a total of 12,200 kg purportedly transported).

As Table XIII shows, all algorithms were able to obtain a similar expected amount of contraband, with 2SLADP-II proving to be the most robust, as measured in terms of nominal-the-best SNR.

## V. CONCLUSION AND FUTURE WORKS

In this paper, we proposed five asset allocation algorithms to the maritime surveillance problem: 1) E-B&C: enumerate all possible asset–case combinations over all times and find the optimal allocation. 2) GB&C-I: enumerate and solve for the best allocation for each asset and compute the PoSD for each asset, and iteratively generate a schedule based on the highest PoSD. 3) GB&C-II: similar to E-B&C, except the algorithm directly enforces the asset schedule once the best allocation is found. 4) *m*SLADP-I: utilize multistep lookahead rollout in a heuristic to iteratively schedule asset–case assignments for individual time epochs. 5) *m*SLADP-II: similar to *m*SLADP-I, except that the algorithm directly enforces the asset schedule based on the highest incremental reward.

We validated each algorithm and solved the NP-hard counter-smuggling surveillance problem in a relatively short amount of time for any of the three objectives examined—maximizing the contraband disrupted, number of detections, or number of smugglers detected. We found that B&C-based methods are able to obtain more contraband when optimizing the amount of contraband disrupted, while the approximate dynamic approaches are better at optimizing over the number of smugglers and the number of detections.

We conducted scalability and robustness analyses to evaluate the solution quality, runtimes, and contraband detection performance behavior of each algorithm. We found that the algorithms scale reasonably well with the problem size. We also found that ADP-based approaches are able to obtain effective asset allocations within seconds of computation time with a minimal sacrifice in optimality, while proving to provide the most robust solution as measured by the SNR metric. Additionally, we found the 2SLADP-II algorithm to be the best when measuring with respect to nominal-the-best SNR. Our future work includes further sensitivity analyses with varying asset types, aloft times, number of unavailable assets, and rest times, and spatiotemporal variations in the PoA surface (e.g., scenario-based asset allocation to handle uncertainty in PoA surfaces). Additionally, higher fidelity simulations could easily be analyzed for more accurate detection models and other operational PoA surfaces (e.g., historical flow surface and active cases). Future work also includes the incorporation of unmanned aerial vehicles (UAVs), either as in [39], where solely UAVs collaborate, or in a mixed-initiative sense, an augmentation of our proposed approach.

## REFERENCES


[1] United Nations Office on Drugs and Crime, "World Drug Report 2010," 2010.

[2] United Nations Office on Drugs and Crime, "World Drug Report 2011," 2011.

[3] Joint Chiefs of Staff, *Command and Control for Joint Maritime Operations*. Washington, DC, USA: Joint Chiefs of Staff, 2013.

[4] A. Smirnov
"Context-driven decision making in network-centric operations: Agent-based intelligent support,"
in *Proc. CKM Workshop*, vol. 7, no. 812, Jan. 2006.

[5] D. Sidoti, D. F. Ayala, S. Sankavaram, X. Han, M. Mishra, W. An, D. Kellmeyer, J. Hansen, and K. Pattipati
"Decision support information integration platform for context-driven interdiction operations in counter-smuggling missions,"
in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, 2014, pp. 659–664.

[6] D. Sidoti, X. Han, L. Zhang, G. V. Avvari, D. F. M. Ayala, M. Mishra, M. S. Sankavaram, D. L. Kellmeyer, J. A. Hansen, and K. R. Pattipati
"Context-aware dynamic asset allocation for maritime interdiction operations,"
*IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 50, no. 3, pp. 1055–1073, 2020.

[7] D. Sidoti, D. F. M. Ayala, X. Han, M. Mishra, S. Sankavaram, W. An, K. R. Pattipati, and D. L. Kleinman
"Evaluating the value of information in the presence of high uncertainty,"

in *Proc. 18th ICCRTS: C2 in Underdeveloped, Degraded, and Denied Operational Environments*, June 2013.

[8] J. A. Hansen, D. Hodyss, C. H. Bishop, and W. Campbell "Coupled METOC/INTEL risk assessment," U.S. Patent Appl. 13/272,272, Oct. 13, 2011. [Online]. Available: https://www.google.com/patents/US20120095946

[9] B. O. Koopman "A theoretical basis for method of search and screening," Columbia Univ., New York, NY, USA, Tech. Rep., 1946.

[10] B. O. Koopman *Search and Screening: General Principles with Historical Applications*, vol. 7. New York, NY, USA: Pergamon Press, 1980.

[11] L. H. Nunn "An introduction to the literature of search theory," Operations Evaluation Group, Center for Naval Analyses, Alexandria, VA, USA, Tech. Rep., 1981.

[12] N. Blachman and F. Proschan "Optimum search for objects having unknown arrival times," *Oper. Res.*, vol. 7, no. 5, pp. 625–638, 1959.

[13] S. M. Pollock "Sequential search and detection," MIT Operations Research Center, Cambridge, MA, USA, Tech. Rep., 1964.

[14] A. Charnes and W. W. Cooper "The theory of search: Optimum distribution of search effort," *Manage. Sci.*, vol. 5, no. 1, pp. 44–50, 1958.

[15] L. D. Stone *Theory of Optimal Search*, vol. 118. Amsterdam, The Netherlands: Elsevier, 1976.

[16] M. Lukka *On the Optimal Searching Tracks for a Stationary Target*, no. 4. Turku, Finland: Institute for Applied Mathematics, University of Turku, 1974.

[17] M. Mangel "Search for a randomly moving object," *SIAM J. Appl. Math*, vol. 40, no. 2, pp. 327–338, 1981.

[18] A. Washburn and K. Wood "Two-person zero-sum games for network interdiction," *Oper. Res.*, vol. 43, no. 2, pp. 243–251, 1995.

[19] D. M. Pfeiff "Optimizing employment of search platforms to counter self-propelled semi-submersibles," Naval Postgraduate School, Monterey, CA, USA, Tech. Rep., 2009.

[20] J. O. Royset and R. K. Wood "Solving the bi-objective maximum-flow network-interdiction problem," *INFORMS J. Comput.*, vol. 19, no. 2, pp. 175–184, 2007.

[21] S. H. Jacobson, L. A. McLay, S. N. Hall, D. Henderson, and D. E. Vaughan "Optimal search strategies using simultaneous generalized hill climbing algorithms," *Math. Comput. Model.*, vol. 43, no. 9–10, pp. 1061–1073, 2006.

[22] K. Ng and N. Sancho "Regional surveillance of disjoint rectangles: A travelling salesman formulation," *J. Oper. Res. Soc.*, vol. 60, no. 2, pp. 215–220, 2009.

[23] A. O. Hero, D. Castañón, D. Cochran and K. Kastella *Foundations and Applications of Sensor Management*. Berlin, Germany: Springer, 2007.

[24] M. Kress, J. O. Royset, and N. Rozen "The eye and the fist: Optimizing search and interdiction," *Eur. J. Oper. Res.*, vol. 220, no. 2, pp. 550–558, 2012.

[25] J. Pietz and J. O. Royset "Generalized orienteering problem with resource dependent rewards," *Naval Res. Logistics*, vol. 60, no. 4, pp. 294–312, 2013.

[26] S. Campos III "An analysis of mathematical models to improve counter-drug smuggling operations," Naval Postgraduate School, Monterey, CA, USA, Tech. Rep., 2014.

[27] J. Pietz and J. O. Royset "Optimal search and interdiction planning," *Military Oper. Res.*, vol. 20, no. 4, pp. 59–73, 2015.

[28] E.-M. Wong, F. Bourgault, and T. Furukawa "Multi-vehicle Bayesian search for multiple lost targets," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 3169–3174.

[29] H. Sato and J. O. Royset "Path optimization for the resource-constrained searcher," *Naval Res. Logistics*, vol. 57, no. 5, pp. 422–440, 2010.

[30] J. O. Royset and H. Sato "Route optimization for multiple searchers," *Naval Res. Logistics*, vol. 57, no. 8, pp. 701–717, 2010.

[31] K. Ng and A. Ghanmi "An automated surface surveillance system," *J. Oper. Res. Soc.*, vol. 53, no. 7, pp. 697–708, 2002.

[32] W. An, D. F. M. Ayala, D. Sidoti, M. Mishra, X. Han, K. R. Pattipati, E. D. Regnier, D. L. Kleinman, and J. A. Hansen "Dynamic asset allocation approaches for counter-piracy operations," in *Proc. 15th IEEE Int. Conf. Inf. Fusion*, 2012, pp. 1284–1291.

[33] K. Byers "Situational awareness for surveillance and interdiction operations (SASIO): Tactical installation protection," Master's thesis, Naval Postgraduate School, Monterey, CA, USA, 2010.

[34] D. L. Bessman "Optimal interdiction of an adaptive smuggler," Naval Postgraduate School, Monterey, CA, USA, Tech. Rep., 2010.

[35] D. P. Bertsekas *Dynamic Programming and Optimal Control*, vol. 1, no. 2. Belmont, MA, USA: Athena Scientific, 1995.

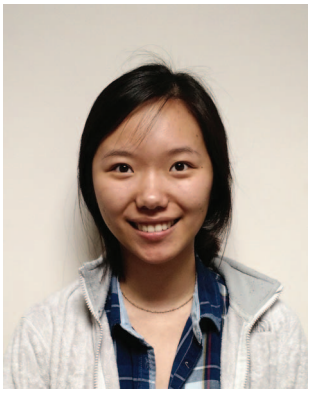
[36] M. R. Garey and D. S. Johnson "'Strong' NP-completeness results: Motivation, examples, and implications," *J. ACM*, vol. 25, no. 3, pp. 499–508, 1978.

[37] M. S. Phadke *Quality Engineering Using Robust Design*. Upper Saddle River, NJ, USA: Prentice Hall, 1995.

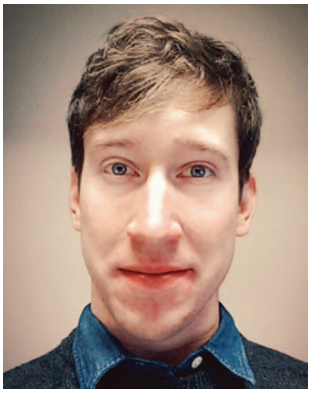
[38] L. J. Watkins "Self-propelled semi-submersibles: The next great threat to regional security and stability," Master's thesis, Naval Postgraduate School, Monterey, CA, USA, 2011.

[39] A. J. Healey, D. Horner, S. Kragelund, B. Wring, and A. Monarrez "Collaborative unmanned systems for maritime and port security operations," in *Proc. IFAC Conf. Control Adv. Marine Syst.*, Sep. 2007

[40] G. V. Avvari, D. Sidoti, M. Mishra, L. Zhang, B. K. Nadella, K. R. Pattipati, and J. A. Hansen "Dynamic asset allocation for counter-smuggling operations under disconnected, intermittent and low-bandwidth environment," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl.*, 2015, pp. 1–6.

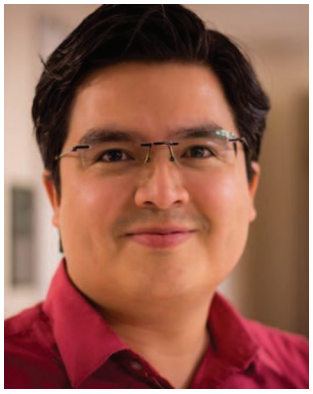


**Lingyi Zhang** received the B.S. and M.S. degrees in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA, in 2014 and 2019, respectively. She is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering at the same university. Her current research interests include modeling dynamic and uncertain environments for asset allocation and path planning, optimization-based techniques for mission planning and coordination, and adaptive Kalman filtering. She was the co-recipient of the Tammy Blair award for best student paper at FUSION 2016.



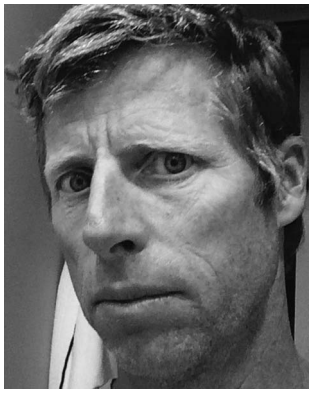
**David Sidoti** received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA, in 2011, 2016, and 2018, respectively. He is currently a Computer Scientist with the Meteorological Applications Development Branch, U.S. Naval Research Laboratory, Marine Meteorology Division (MMD), Monterey, CA, USA. He is NRLMMD's primary machine learning subject matter expert and contributes to a broad portfolio of machine learning projects relating to atmospheric and oceanographic forecasting. His current interests include multiobjective algorithms for dynamic scheduling and resource management in weather-impacted environments, and deep learning applications to numerical weather prediction. He was the co-recipient of the Tammy Blair award for best student paper at FUSION 2016. In 2018, he was recognized and awarded a Distinguished Scholar Jerome and Isabella Karle's Fellowship.

**Gopi Vinod Avvari** received the B. Tech. degree in electronics and communication engineering from Acharya Nagarjuna University, Guntur, India, in 2011, and the M.S and Ph.D. degrees in electrical engineering from the University of Connecticut, Storrs, CT, USA, in 2016 and 2018, respectively. He is currently a Machine Learning Scientist with Aptiv, Agoura Hills, CA, USA. His research interests include developing algorithms for the advanced driver assisting systems and autonomous driving.



**Diego F. M. Ayala** (M'06) received the B.S. degree in electrical engineering from the Industrial University of Santander (Universidad Industrial de Santander), Bucaramanga, Colombia, in 2007, and the M.Sc. degree in electrical engineering from the University of Connecticut, Mansfield, CT, USA, in 2014, where he is currently working toward the Ph.D. degree in electrical and computer engineering. His current research interests include development of models and methods for synthesizing adaptive organizations and for planning in dynamic and uncertain environments, multiobjective optimization techniques for engineering applications, development of models of distributed information processing in human teams, Bayesian networks, Markov processes, and machine learning in the context of big data analysis, and renewable energy technologies with SMART grid applications.

**Manisha Mishra** received the B.S. degree in electrical engineering from Punjab Technical University, Jalandhar, India, in 2004, the M.S. degree in electrical and computer engineering from the University of Hawaii, Honolulu, HI, USA, in 2008, and the Ph.D. degree in electrical and computer engineering from the University of Connecticut, Storrs, CT, USA, in 2017. She was a Software Engineer with Infosys Technologies, India, in 2005. She was an ORISE Post-Graduate Researcher with U.S. Army Engineering Research and Development Center—Cold Region Research Labs in 2017. She is currently an Algorithm Engineer with Aptiv Corporation. Her research interests include modeling dynamic and uncertain environments for asset allocation and path planning, context-aware decision support systems, risk analysis, system diagnosis and prognosis, and optimization-based techniques for mission planning and coordination.



**David L. Kellmeyer** received the M.S. degree in human factors engineering from Ohio University, Athens, OH, USA, in 1992, where his research focused on highway safety and artificial intelligence. He is currently a Human Factors Engineer and Project Manager with the User-Centered Design (UCD) Team, SPAWAR Systems Center—Pacific, San Diego, CA, USA (SSC-Pac). He served 4 years with the Army Medical Service Corp providing industrial hygiene and ergonomics support for the Center for Health Promotion and Preventive Medicine and the National Security Agency (NSA). During the past 20 years with SSC-Pac, he has led projects focusing on the need to develop human–computer interfaces that reduce workload and support better supervisor control and situational awareness. He has supported multiple industry teams to include Arsenal Ship, DD 21, and the Unmanned Combat Aerial Vehicle (UCAV). He has served in three off-site tours in Washington, DC, USA, supporting the Department of Homeland Security, the Transportation Safety Administration (TSA), and the Office of Naval Research. He was the lead designer for the Joint Inter-Agency Task Force—South's (JIATF-South) new command center. During the last 5 years, he has led two Office of Naval Research projects that transitioned supervisory control technologies to the Submarine Combat Office and the Navy's Command and Control program office. He has worked for numerous Department of Defense and government agencies, including The White House, Pacific Command, Strategic Command, NSA, TSA, Department of Homeland Security, and JIATF-South.

**James A. Hansen** received the undergraduate and master's degrees in aerospace engineering from the University of Colorado Boulder, Boulder, CO, USA, in 1992 and 1993, respectively, and the doctoral degree from the University of Oxford, Oxford, U.K., in 1998. He was a Rhodes Scholar with Atmospheric, Oceanic, and Planetary Physics Department, University of Oxford. He was an Associate Professor with the Earth, Atmospheric, and Planetary Sciences Department, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently the Superintendent of the Marine Meteorology Division, Naval Research Laboratory (NRL), Washington, DC, USA. He has been with NRL, since 2006, and during that time he has been the Lead Scientist of the NRL Probabilistic Prediction Research Office and the Head of the Meteorological Applications Development Branch. He has authored or coauthored dozens of peer-reviewed publications. His current research interests include estimation of environmental forecast uncertainty and the use of that uncertainty in decision making. He was a recipient of numerous awards and honors, the most recent of which are the Navy Meritorious Civilian Service Award for meritorious performance of service and the A. S. Flemming Award for Exceptional Federal Service. He was a Research and Development Lead in the Piracy Attack Risk Surface Project—a now operational system for coupling environmental, intelligence, and behavioral factors to predict the risk of pirate attack. He serves on several scientific advisory boards and was an Editor for *Monthly Weather Review*.

**Krishna R. Pattipati** received the B.Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, India, in 1975, and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut, Storrs, CT, USA, in 1977 and 1980, respectively. He was with AlphaTech, Inc., Burlington, MA, USA, from 1980 to 1986. He has been with the Department of Electrical and Computer Engineering, University of Connecticut, where he is currently the Board of Trustees Distinguished Professor and the UTC Chair Professor in Systems Engineering. His research activities are in the areas of proactive decision support, uncertainty quantification, smart manufacturing, autonomy, knowledge representation, and optimization-based learning and inference. A common theme among these applications is that they are characterized by a great deal of uncertainty, complexity, and computational intractability. He is a co-founder of Qualtech Systems, Inc., a firm specializing in advanced integrated diagnostics software tools (TEAMS, TEAMS-RT, TEAMS-RDS, TEAMATE), and serves on the board of Aptima, Inc. He was selected by the IEEE Systems, Man, and Cybernetics (SMC) Society as the Outstanding Young Engineer of 1984, and received the Centennial Key to the Future award. He was the Editor-in-Chief of the IEEE Transactions on Systems, Man, and Cybernetics—Part B from 1998 to 2001, Vice-President for Technical Activities of the IEEE SMC Society from 1998 to 1999, and Vice-President for Conferences and Meetings of the IEEE SMC Society from 2000 to 2001. He was co-recipient of the Andrew P. Sage Award for the Best SMC Transactions Paper for 1999, the Barry Carlton Award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-Based Health Monitoring and Diagnosis," and "Real-Time Update of Fault-Test Dependencies of Dynamic Systems: A Comprehensive Toolset for Model-Based Health Monitoring and Diagnostics," and the 2003 AAUP Research Excellence Award at UCONN. He also won the best technical paper awards at the 1985, 1990, 1994, 2002, 2004, 2005, and 2011 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences. He is an elected Fellow of IEEE and of the Connecticut Academy of Science and Engineering.

# A New Heterogeneous Track Fusion with Information Decorrelation Algorithm for Target Tracking in a Multistatic Sensor System with Non-Cooperative Moving Transmitters

**RONG YANG**
**YAAKOV BAR-SHALOM**

**This paper considers a target tracking problem in a non-cooperative multistatic system, where several transmitters are moving and their positions are unknown. The receiver listens to the signals from non-cooperative transmitters via direct and indirect (bouncing from targets) paths. The transmitters and targets are then tracked based on the measured bearings and the bistatic ranges (derived from the time difference of arrival of the direct and indirect path signals) simultaneously. In previous work, we proved that the transmitter trajectories are observable when the two transmitters are not located on the same line from the receiver, and developed an approximate algorithm to perform estimation based on covariance inflation (CI). In this paper, a new estimation algorithm, heterogeneous track fusion with information decorrelation (HTF-D), is developed. It aims to achieve optimal estimates without using a large augmented state consisting of all transmitter and target states. The approach tracks targets individually, and fuses these highly correlated tracks through a novel information decorrelation method. The performance of the HTF-D is evaluated through simulation tests. The results show that the HTF-D provides better estimates than the CI algorithm, and achieves the same accuracy as the optimal algorithm when the latter does not suffer from numerical problems due to its large augmented state. The HTF-D estimates are also consistent statistically.**

Manuscript received January 2, 2020; revised May 4, 2020; released for publication June 30, 2020.

Refereeing of this contribution was handled by Florian Meyer.
R. Yang is with the DSO National Laboratories, 12 Science Park Drive, Singapore 118225, Singapore (E-mail: yrong@dso.org.sg). Y. Bar-Shalom is with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA (E-mail: yaakov.bar-shalom@uconn.edu).



## I. INTRODUCTION

Passive detection and tracking is desirable in surveillance systems as it estimates target trajectories through "listening" to the signals emitted by others without any emission of its own. It can therefore avoid being detected by the targets it observes. This paper will develop a new passive tracking algorithm primarily for passive sonar and radar applications.

Passive tracking algorithms operate according to two basic approaches. One is to listen to target emission signals and then to estimate target trajectories. The techniques such as bearings-only tracking (BOT) and Doppler-bearing tracking (DBT) belong to this category [1]–[8]. The target motion state (comprising the target position and velocity) is estimated by measuring the target bearings and, if their emissions are narrowband, also their Doppler shifted frequencies. Since the target range cannot be measured, the conventional BOT/DBT [1], [3], [5], [6] usually cannot provide accurate trajectory estimates, even when the platform of the receiver maneuvers. The recently developed unscented Gauss–Helmert filter [2], [4], [7], [8] fuses two types of target signals (e.g., an acoustic signal and an electromagnetic signal) with different propagation delays, so that a target range can be inferred from the delays. This method increases the estimation accuracy significantly. However, it is only applicable to the targets that emit these heterogeneous signals. Another approach to passive tracking is by means of the multistatic sensor concept, which does not require any emission from targets. The receiver (listener) listens to some transmitters' signals through direct and indirect (bouncing from targets) paths. The targets can then be tracked based on the measured bearings and bistatic ranges. The bistatic ranges are derived from the time difference of arrival (TDOA) of the direct and indirect path signals. This paper will present a generalization of the multistatic approach, detailed in the sequel.

The multistatic tracking has been extensively studied. The international Multistatic Tracking Working Group, organized in 2005, dedicated research to this topic. Their research results were summarized in [9]. The algorithms they used include Bayesian tracking, distributed multiple hypothesis tracking, probabilistic multiple hypothesis tracking, maximum likelihood probabilistic data association algorithm, and Gaussian mixture cardinalized probability hypothesis density tracking. All of them solve the tracking problem for the conventional multistatic system where the positions of transmitters and receivers are assumed known. However, the transmitters' positions are not always known, especially for moving transmitters, which do not inform the receiver of their positions in a timely manner. We designate this type of transmitters as non-cooperative transmitters. To widen the application of multistatic systems, research on non-cooperative transmitters is of special interest. In general, the moving target trajectories are not observable based on the measurements from a

non-cooperative transmitter only. Additional information needs to be provided to estimate the transmitter positions. A well-known method is to estimate the transmitter position from objects with known positions (e.g., landmarks, rocks, or wrecks) [10]–[12]. The environmental information such as walls and other reflectors can also be utilized [13]–[15].[1] Once the transmitter position is known, the target trajectories are observable in a multistatic system. A study on non-cooperative transmitter system without using additional information was also conducted recently [16]. It estimates the positions of targets and a transmitter simultaneously purely from the measurements. However, this approach is only applicable for stationary targets.

This paper focuses on a new non-cooperative multistatic configuration [17], [18] that can track transmitters and targets simultaneously, where targets and transmitters can be either moving or stationary, and no additional information such as reflections from known objects and environment is required. The only requirement is that the number of non-cooperative transmitters must be greater than or equal to 2. This configuration was first proposed in [17]. The observability of the problem was proved when the two transmitters are not located on the same line of sight from the receiver (see Appendix A). Simulation tests were conducted using synchronous transmitters for proof of concept. Further study [18] extended the algorithm from synchronous transmitters to the more realistic asynchronous case. This is because non-cooperative transmitters do not negotiate each other to synchronize their emission times. The covariance inflation (CI) filter [19] was used for transmitter and target trajectory estimation in [18] and its estimation results were compared to the optimal algorithm that estimates the augmented state consisting of all transmitter and target states. Unfortunately, the CI filter, being an approximate approach, has a significant accuracy gap to the optimal algorithm. The optimal algorithm is straightforward but, unfortunately, has the following issues:

1) The state size will be large when many targets exist. This will cause the estimation to be more sensitive to the system errors, and it is likely to have numerical problems.
2) The state size varies with the number of targets in the system. This complicates the real system applications.
3) The system may face scalability issues as all target and transmitter estimation is handled by one estimator.

Thus, a better tracking algorithm is needed.

[1]Strictly speaking, the systems addressed in these references are not multistatic systems since they consider only transmitters and receivers. Non-emitting targets are not there. However, the approaches for the transmitter/receiver location estimation in these papers can be utilized in a non-cooperative multistatic system when the additional environmental information is available.

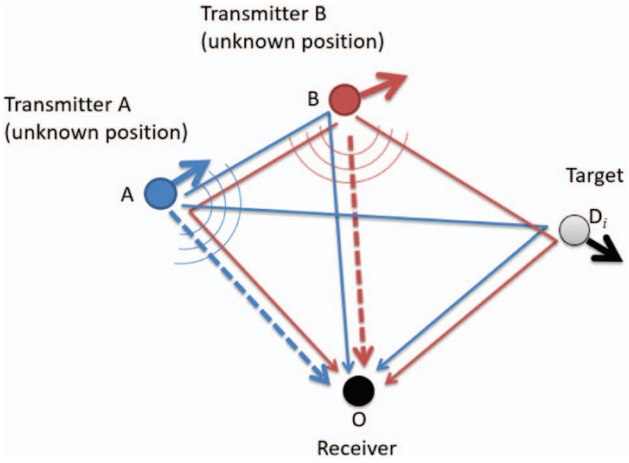



Fig. 1. Noncooperative multistatic system with two asynchronous transmitters with unknown positions. Transmitters A and B move and emit "blue" and "red" signals (on different frequencies), respectively. The dashed lines are the direct signal paths, and the solid lines are the indirect signal paths.

In this paper, we present a novel heterogeneous track fusion with information decorrelation (HTF-D) algorithm that provides similar accuracy to the optimal algorithm and estimates each target trajectory individually, so that the drawbacks of the optimal algorithm caused by the large state size are avoided. The technical challenge is that the cross-covariances among the target estimation errors cannot be addressed properly if we track the targets individually. The optimal algorithm estimates a large augmented state so that these cross-covariances can be computed easily. The HTF-D, tracking the targets individually, does not compute the cross-covariances between individual tracks. To overcome this, we develop a new *heterogeneous and correlated state fusion method* on top of the individual tracks, so that the cross-covariances can be handled properly through the fusion.

The rest of paper is organized as follows. Section 2 describes the problem formulation. Section 3 reviews the existing estimation algorithms relevant to our problem. Section 4 describes the HTF-D algorithm. Section 5 shows the simulation results, and conclusions are given in Section 6.

## II. PROBLEM FORMULATION

The problem is illustrated in Fig. 1. A stationary receiver[2] is located at the origin of the Cartesian coordinate system $O(0, 0)$. Two moving platforms A and B carry emitters that emit asynchronous pulse signals periodically with different frequencies (represented in red and blue). One moving target (without emitter) reflects the signals to the receiver. Note that only one target is shown in Fig. 1 for simplicity. The approach is

[2]The approaches described in the paper are not limited to a stationary receiver; they can also work on a moving receiver, assuming the trajectory of the receiver is known.

applicable to multiple targets as shown later in the simulation results. The positions of the transmitters and target are unknown to the receiver. The direct signal paths from the two transmitters are represented by dashed lines, while the indirect signal paths reflected via another transmitter or target are represented by solid lines. It is easy for the receiver to differentiate the direct and indirect signals from their amplitudes, as the direct signals are much stronger than the indirect signals [20].

The two transmitters emit signals asynchronously, based on their own schedules. The receiver needs to perform estimation when it receives signals from any one of transmitters.

The states to be estimated are

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

$$\mathbf{x}^{\mathrm{B}}(t_k) = [x^{\mathrm{B}}(t_k)\ \ y^{\mathrm{B}}(t_k)\ \ \dot{x}^{\mathrm{B}}(t_k)\ \ \dot{y}^{\mathrm{B}}(t_k)]', \tag{2}$$

$$\mathbf{x}^{\mathrm{D}_i}(t_k) = \left[x^{\mathrm{D}_i}(t_k)\ \ y^{\mathrm{D}_i}(t_k)\ \ \dot{x}^{\mathrm{D}_i}(t_k)\ \ \dot{y}^{\mathrm{D}_i}(t_k)\right]', \tag{3}$$

where $\mathbf{x}^{\mathrm{A}}$ and $\mathbf{x}^{\mathrm{B}}$ are the states of the transmitters A and B, respectively, $\mathbf{x}^{\mathrm{D}_i}$ is the target state with $i = 1, \ldots, n$, and $n$ is the number of targets in the system. The state transition model assuming a nearly constant velocity (NCV) model is

$$\mathbf{x}^*(t_k) = \mathbf{F}(T_{k,k-1})\mathbf{x}^*(t_{k-1}) + \mathbf{v}^*(T_{k,k-1}), \tag{4}$$

where $\boxtimes^*$ represents $\boxtimes^{\mathrm{A}}$, $\boxtimes^{\mathrm{B}}$, or $\boxtimes^{\mathrm{D}_i}$,

$$\mathbf{F}(T_{k,k-1}) = \begin{bmatrix} 1 & 0 & T_{k,k-1} & 0 \\ 0 & 1 & 0 & T_{k,k-1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

with the time interval (not a constant in the asynchronous case)

$$T_{k,k-1} = t_k - t_{k-1}, \tag{6}$$

and $\mathbf{v}^*$ is the zero-mean process noise for the interval $T_{k,k-1}$. Based on the discretized continuous-time white noise acceleration model [21], its covariance is

$$E[\mathbf{v}^*(\cdot)\mathbf{v}^*(\cdot)'] = \mathbf{Q}(T_{k,k-1}) = \begin{bmatrix} \frac{T_{k,k-1}^3}{3} & 0 & \frac{T_{k,k-1}^2}{2} & 0 \\ 0 & \frac{T_{k,k-1}^3}{3} & 0 & \frac{T_{k,k-1}^2}{2} \\ \frac{T_{k,k-1}^2}{2} & 0 & T_{k,k-1} & 0 \\ 0 & \frac{T_{k,k-1}^2}{2} & 0 & T_{k,k-1} \end{bmatrix} q, \tag{7}$$

where $q$ is the power spectral density (PSD) of the continuous-time (acceleration) process noise (same for $x$ and $y$, and assumed independent between the coordinates). The measurement vectors are

$$\mathbf{z}^{\mathrm{A}}(t_k) = \begin{cases} b^{\mathrm{A}}(t_k), & \text{when A is emitting at } t_k, \\ [r^{\mathrm{AB}}(t_k)\ \ b^{\mathrm{A}}(t_k)]', & \text{when B is emitting at } t_k, \end{cases} \tag{8}$$

$$\mathbf{z}^{\mathrm{B}}(t_k) = \begin{cases} b^{\mathrm{B}}(t_k), & \text{when B is emitting at } t_k, \\ [r^{\mathrm{BA}}(t_k)\ \ b^{\mathrm{B}}(t_k)]', & \text{when A is emitting at } t_k, \end{cases} \tag{9}$$

$$\mathbf{z}^{\mathrm{D}_i}(t_k) = \begin{cases} [r^{\mathrm{D}_i\mathrm{A}}(t_k)\ \ b^{\mathrm{D}_i}(t_k)]', & \text{when A is emitting at } t_k, \\ [r^{\mathrm{D}_i\mathrm{B}}(t_k)\ \ b^{\mathrm{D}_i}(t_k)]', & \text{when B is emitting at } t_k, \end{cases} \tag{10}$$

where (see Fig. 1)

$$r^{\mathrm{AB}}(t_k) = |AO| + |AB| - |BO| + w_r^{\mathrm{AB}}(t_k), \tag{11}$$

$$r^{\mathrm{BA}}(t_k) = |BO| + |AB| - |AO| + w_r^{\mathrm{BA}}(t_k), \tag{12}$$

$$r^{\mathrm{D}_i\mathrm{A}}(t_k) = |D_iO| + |AD_i| - |AO| + w_r^{\mathrm{D}_i\mathrm{A}}(t_k), \tag{13}$$

$$r^{\mathrm{D}_i\mathrm{B}}(t_k) = |D_iO| + |BD_i| - |BO| + w_r^{\mathrm{D}_i\mathrm{B}}(t_k), \tag{14}$$

with

$$|AO| = \sqrt{x^{\mathrm{A}}(t_k)^2 + y^{\mathrm{A}}(t_k)^2}, \tag{15}$$

$$|BO| = \sqrt{x^{\mathrm{B}}(t_k)^2 + y^{\mathrm{B}}(t_k)^2}, \tag{16}$$

$$|D_iO| = \sqrt{x^{\mathrm{D}_i}(t_k)^2 + y^{\mathrm{D}_i}(t_k)^2}, \tag{17}$$

$$|AB| = \sqrt{[x^{\mathrm{A}}(t_k) - x^{\mathrm{B}}(t_k)]^2 + [y^{\mathrm{A}}(t_k) - y^{\mathrm{B}}(t_k)]^2}, \tag{18}$$

$$|AD_i| = \sqrt{[x^{\mathrm{A}}(t_k) - x^{\mathrm{D}_i}(t_k)]^2 + [y^{\mathrm{A}}(t_k) - y^{\mathrm{D}_i}(t_k)]^2}, \tag{19}$$

$$|BD_i| = \sqrt{[x^{\mathrm{B}}(t_k) - x^{\mathrm{D}_i}(t_k)]^2 + [y^{\mathrm{B}}(t_k) - y^{\mathrm{D}_i}(t_k)]^2}, \tag{20}$$

are the four bistatic ranges, and $w_r^{\mathrm{AB}}$, $w_r^{\mathrm{BA}}$, $w_r^{\mathrm{D}_i\mathrm{A}}$, and $w_r^{\mathrm{D}_i\mathrm{B}}$ are their measurement noises, assumed to be zero-mean white Gaussian with variance $\sigma_r^2$, and

$$b^*(t_k) = \tan^{-1}\left[\frac{x^*(t_k)}{y^*(t_k)}\right] + w_b^*(t_k) \tag{21}$$

are three bearings, where $\boxtimes^*$ represents $\boxtimes^{\mathrm{A}}$, $\boxtimes^{\mathrm{B}}$, or $\boxtimes^{\mathrm{D}_i}$, and $w_b^*$ is the bearing measurement Gaussian error with variance $\sigma_b^2$.

## III. REVIEW ON THE EXISTING APPROACHES

The three existing estimation approaches that can be applied to our problem are reviewed in this section. They are the optimal, simple, and CI algorithms. The review focuses on the dynamic state estimation. The parameter

estimation used for the track initiation is not included. Interested readers can find the track initiation details in [18].

### A. The Optimal Algorithm

The optimal algorithm [18] is straightforward. It estimates the augmented state $\mathbf{x}^{\mathrm{ALL}}$ using the augmented measurement $\mathbf{z}^{\mathrm{ALL}}$, which are defined as

$$\mathbf{x}^{\mathrm{ALL}}(t_k) = [\mathbf{x}^{\mathrm{A}}(t_k)' \ \mathbf{x}^{\mathrm{B}}(t_k)' \ \mathbf{x}^{\mathrm{D}_1}(t_k)' \ \cdots \ \mathbf{x}^{\mathrm{D}_n}(t_k)']', \quad (22)$$

$$\mathbf{z}^{\mathrm{ALL}}(t_k) = [\mathbf{z}^{\mathrm{A}}(t_k)' \ \mathbf{z}^{\mathrm{B}}(t_k)' \ \mathbf{z}^{\mathrm{D}_1}(t_k)' \ \cdots \ \mathbf{z}^{\mathrm{D}_n}(t_k)']', \quad (23)$$

where the sizes of $\mathbf{x}^{\mathrm{ALL}}(t_k)$ and $\mathbf{z}^{\mathrm{ALL}}(t_k)$ are determined by the number of targets and transmitters in the system. The state transition model and measurement model are

$$\mathbf{x}^{\mathrm{ALL}}(t_k) = \mathbf{F}^{\mathrm{ALL}}(T_{k,k-1})\mathbf{x}^{\mathrm{ALL}}(t_{k-1}) + \mathbf{v}^{\mathrm{ALL}}(T_{k,k-1}), \quad (24)$$

$$\mathbf{z}^{\mathrm{ALL}}(t_k) = \mathbf{h}^{\mathrm{ALL}}[\mathbf{x}^{\mathrm{ALL}}(t_k)] + \mathbf{w}^{\mathrm{ALL}}(t_k), \quad (25)$$

where

$$\mathbf{F}^{\mathrm{ALL}}(T_{k,k-1}) = \begin{bmatrix} \mathbf{F}(T_{k,k-1}) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{F}(T_{k,k-1}) \end{bmatrix}, \quad (26)$$

$\mathbf{h}^{\mathrm{ALL}}[\cdot]$ can be derived from (8)–(21), and $\mathbf{v}^{\mathrm{ALL}}$ and $\mathbf{w}^{\mathrm{ALL}}$ are the process noise and measurement noise, respectively.

The optimal algorithm performs estimation without losing any information. Thus, it can in theory obtain the best estimates. However, when the system is marginally observable, the augmented state is large, and the measurement noise is large, the optimal algorithm is inferior to the new algorithm (as it will be shown in Section 4) because the latter handles small states (each target state is estimated separately). The state estimate error covariance (time $t_k$ is omitted for simplicity) for the optimal algorithm's augmented state is

$$\mathbf{P}^{\mathrm{ALL}} = \begin{bmatrix} \mathbf{P}^{\mathrm{A}} & \mathbf{P}^{\mathrm{AB}} & \mathbf{P}^{\mathrm{AD}_1} & \cdots & \mathbf{P}^{\mathrm{AD}_n} \\ \mathbf{P}^{\mathrm{BA}} & \mathbf{P}^{\mathrm{B}} & \mathbf{P}^{\mathrm{BD}_1} & \cdots & \mathbf{P}^{\mathrm{BD}_n} \\ \mathbf{P}^{\mathrm{D}_1\mathrm{A}} & \mathbf{P}^{\mathrm{D}_1\mathrm{B}} & \mathbf{P}^{\mathrm{D}_1} & \cdots & \mathbf{P}^{\mathrm{D}_1\mathrm{D}_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^{\mathrm{D}_n\mathrm{A}} & \mathbf{P}^{\mathrm{D}_n\mathrm{B}} & \mathbf{P}^{\mathrm{D}_n\mathrm{D}_1} & \cdots & \mathbf{P}^{\mathrm{D}_n} \end{bmatrix}. \quad (27)$$

The off-diagonal blocks are the cross-covariances among the transmitter and target state estimate errors. In this problem, the transmitter and target state estimate errors are all cross-correlated. The off-diagonal blocks are therefore nonzero.

Although the optimal algorithm can provide optimal solution in theory, it has three drawbacks given in Section 1. Thus, to track the targets individually (without large augmented state size), accounting for the coupling is preferred.

### B. The Simple Algorithm

The simple algorithm [18] tracks the transmitters and targets individually in a completely decoupled manner. The state and measurement vectors are $\mathbf{x}^*(t_k)$ in (1)–(3) and $\mathbf{z}^*(t_k)$ in (8)–(10), respectively. The state transition model is as in (4), and the measurement model is (8)–(21). It can be seen that the bistatic ranges in (11)–(14) are functions of the target state and the emitting transmitter position (which can be either transmitter A or transmitter B based on who is emitting at time $t_k$). At a particular time $t_k$, the emitting transmitter state must be estimated first, so that the updated transmitter position can be used in the measurement model for estimating the other transmitter and the target states.

This approach ignores the transmitter state estimate errors [namely, $\mathbf{P}^{\mathrm{A}}$ and $\mathbf{P}^{\mathrm{B}}$ in (27)] in the other transmitter and target state estimation. Furthermore, since the transmitters and targets are estimated individually, the cross-covariances among estimate errors in the optimal algorithm [all the off-diagonal blocks in (27)] are ignored.

### C. The CI Algorithm

This approach was used in [18] based on the CI method [19]. Similarly to the simple approach, the transmitter states are estimated first, and then the estimated transmitter positions are used to estimate the target states. Unlike the simple algorithm, it includes the transmitter position errors (assumed white, which they are not) in the measurement noise when performing target state estimation. Two CI filters were presented in [18]. One tracks the transmitters with augmented transmitter states, whereas another one tracks the transmitters individually. We will review the first one, as it yields better performance.

For the transmitter state estimation, the augmented transmitter state and measurement vectors are

$$\mathbf{x}^{\mathrm{AB}}(t_k) = [\mathbf{x}^{\mathrm{A}}(t_k)' \ \mathbf{x}^{\mathrm{B}}(t_k)']', \quad (28)$$

$$\mathbf{z}^{\mathrm{AB}}(t_k) = [\mathbf{z}^{\mathrm{A}}(t_k)' \ \mathbf{z}^{\mathrm{B}}(t_k)']', \quad (29)$$

respectively. The state transition model and measurement model are

$$\mathbf{x}^{\mathrm{AB}}(t_k) = \mathbf{F}^{\mathrm{AB}}(T_{k,k-1})\mathbf{x}^{\mathrm{AB}}(t_{k-1}) + \mathbf{v}^{\mathrm{AB}}(T_{k,k-1}), \quad (30)$$

$$\mathbf{z}^{\mathrm{AB}}(t_k) = \mathbf{h}^{\mathrm{AB}}[\mathbf{x}^{\mathrm{AB}}(t_k)] + \mathbf{w}^{\mathrm{AB}}(t_k), \quad (31)$$

where

$$\mathbf{F}^{\mathrm{AB}}(T_{k,k-1}) = \begin{bmatrix} \mathbf{F}(T_{k,k-1}) & \mathbf{0} \\ \mathbf{0} & \mathbf{F}(T_{k,k-1}) \end{bmatrix}. \quad (32)$$

The measurement function $\mathbf{h}^{\mathrm{AB}}[\cdot]$ can be derived from (8)–(21), and $\mathbf{v}^{\mathrm{AB}}$ and $\mathbf{w}^{\mathrm{AB}}$ are the process noise and measurement noise, respectively.

For the target estimation, the state and measurement vectors are $\mathbf{x}^{D_i}$ and $\mathbf{z}^{D_i}$ given in (3) and (10), respectively. The state transition model has been given in (4), and the measurement model can be derived from (10)–(21). The CI filter with the following steps is used for the dynamic target state estimation.

**Prediction**:

$$\hat{\mathbf{x}}^{D_i}(t_k|t_{k-1}) = \mathbf{F}(T_{k,k-1})\hat{\mathbf{x}}^{D_i}(t_{k-1}|t_{k-1}), \tag{33}$$

$$\mathbf{P}^{D_i}(t_k|t_{k-1}) = \mathbf{F}(T_{k,k-1})\mathbf{P}^{D_i}(t_{k-1}|t_{k-1})\mathbf{F}(T_{k,k-1})' + \mathbf{Q}^{D_i}(T_{k,k-1}). \tag{34}$$

**Update**:

$$\hat{\mathbf{x}}^{D_i}(t_k|t_k) = \hat{\mathbf{x}}^{D_i}(t_k|t_{k-1}) + \mathbf{K}^{D_i}(t_k)\left[\mathbf{z}^{D_i}(t_k) - \hat{\mathbf{z}}^{D_i}(t_k)\right], \tag{35}$$

$$\mathbf{P}^{D_i}(t_k|t_k) = \mathbf{P}^{D_i}(t_k|t_{k-1}) - \mathbf{K}^{D_i}(t_k)\mathbf{S}^{D_i}(t_k)\mathbf{K}^{D_i}(t_k)', \tag{36}$$

where

$$\hat{\mathbf{z}}^{D_i}(t_k) = \mathbf{h}\left[\hat{\mathbf{x}}^{D_i}(t_k|t_{k-1}), \hat{\mathbf{x}}^{A/B}(t_k)\right], \tag{37}$$

$$\mathbf{K}^{D_i}(t_k) = \mathbf{P}^{D_i}(t_k|t_{k-1})\mathbf{H}^{D_i}(t_k)'\mathbf{S}^{D_i}(t_k)^{-1}, \tag{38}$$

$$\mathbf{S}^{D_i}(t_k) = \mathbf{H}^{D_i}(t_k)\mathbf{P}^{D_i}(t_k|t_{k-1})\mathbf{H}^{D_i}(t_k)' + \mathbf{H}^{A/B}(t_k)\mathbf{P}^{A/B}(t_k)\mathbf{H}^{A/B}(t_k)' + \mathbf{R}^{D_i}, \tag{39}$$

and

$$\mathbf{H}^{D_i}(t_k) = \frac{\partial \mathbf{h}[\mathbf{x}^{D_i}(t_k), \mathbf{x}^{A/B}(t_k)]}{\partial \mathbf{x}(t_k)}, \tag{40}$$

$$\mathbf{H}^{A/B}(t_k) = \frac{\partial \mathbf{h}[\mathbf{x}^{D_i}(t_k), \mathbf{x}^{A/B}(t_k)]}{\partial \mathbf{x}^{A/B}(t_k)}, \tag{41}$$

where $\mathbf{x}^{A/B}$ and $\mathbf{P}^{A/B}$ are the state and error covariance of transmitter A or B based on who is emitting at time $t_k$.

It can be seen that the CI filter is the same as the Kalman filter except the innovation covariance computation in (39). The measurement error covariance $\mathbf{R}^{D_i}$ in the Kalman filter is replaced by the inflated measurement error covariance

$$\mathbf{H}^{A/B}(t_k)\mathbf{P}^{A/B}(t_k)\mathbf{H}^{A/B}(t_k)' + \mathbf{R}^{D_i} \tag{42}$$

in the CI filter. The transmitter error covariance $\mathbf{P}^{A/B}$ is added to the measurement noise. However, $\mathbf{P}^{A/B}$ is not "white," as the transmitter state estimate error is correlated to its history. This conflicts with the Kalman filter requirement on the measurement noise to be white. As indicated in [19], it is clearly not correct since, although the noise in (42) is zero mean, it is by no means white. It yields inconsistent estimates with covariances that are too small. However, the transmitter state estimation can still yield consistent results, as the augmented state $\mathbf{x}^{AB}$ is used.

Furthermore, the target and transmitter states are estimated separately in the CI algorithm, which ignores the contributions of the target measurements $\mathbf{z}^{D_i}$ to the transmitter state; namely, the cross-covariances [$\mathbf{P}^{AD_i}$ and $\mathbf{P}^{BD_i}$ in (27) in the optimal algorithm] are not handled correctly. Similarly to the simple approach, the individual target state estimation also causes the cross-covariances [$\mathbf{P}^{D_iD_j}$ with $i \neq j$ in (27) in the optimal algorithm] to be not treated correctly. Although the CI approach is better than the simple approach in term of information loss, it still has a gap to the optimal approach.

## IV. THE HTF-D ALGORITHM

It is desirable to have an approach to track the targets individually with small state size, and obtain the estimates without losing any information. In this section, we will develop such an approach. The principle, shown in Fig. 2, consists of the following two steps:

(A) To track targets individually with minimal augmented states [defined in (43)]. An augmented state consists of the two transmitter states and one particular target state. It is updated by the corresponding transmitter and target measurements defined in (44). Since the transmitter measurements, $\mathbf{z}^A$ and $\mathbf{z}^B$, are used to update all the individual tracks, the individual track state estimation errors are highly correlated, since they include common information.

(B) To fuse the individual track estimates through a special information fusion algorithm. The common information among the individual tracks is computed first, so that the common information can be decorrelated, and the individual tracks can then be fused without double counting the common information.

We designate this approach as the HTF-D. The details of the two steps are given next.

### A. Individual Target Tracking

Assume there are $n$ targets in the system. The $i$th individual target state and measurement vectors are defined as

$$\mathbf{x}^i(t_k) = [\mathbf{x}^A(t_k)' \ \mathbf{x}^B(t_k)' \ \mathbf{x}^{D_i}(t_k)']', \tag{43}$$

$$\mathbf{z}^i(t_k) = [\mathbf{z}^A(t_k)' \ \mathbf{z}^B(t_k)' \ \mathbf{z}^{D_i}(t_k)']', \tag{44}$$

where $i = 1, \ldots, n$, and $n$ is the number of the targets. The NCV state transition model and measurement model are

$$\mathbf{x}^i(t_k) = \mathbf{F}^i(T_{k,k-1})\mathbf{x}^i(t_{k-1}) + \mathbf{v}^i(T_{k,k-1}), \tag{45}$$

$$\mathbf{z}^i(t_k) = \mathbf{h}^i[\mathbf{x}^i(t_k)] + \mathbf{w}^i(t_k), \tag{46}$$

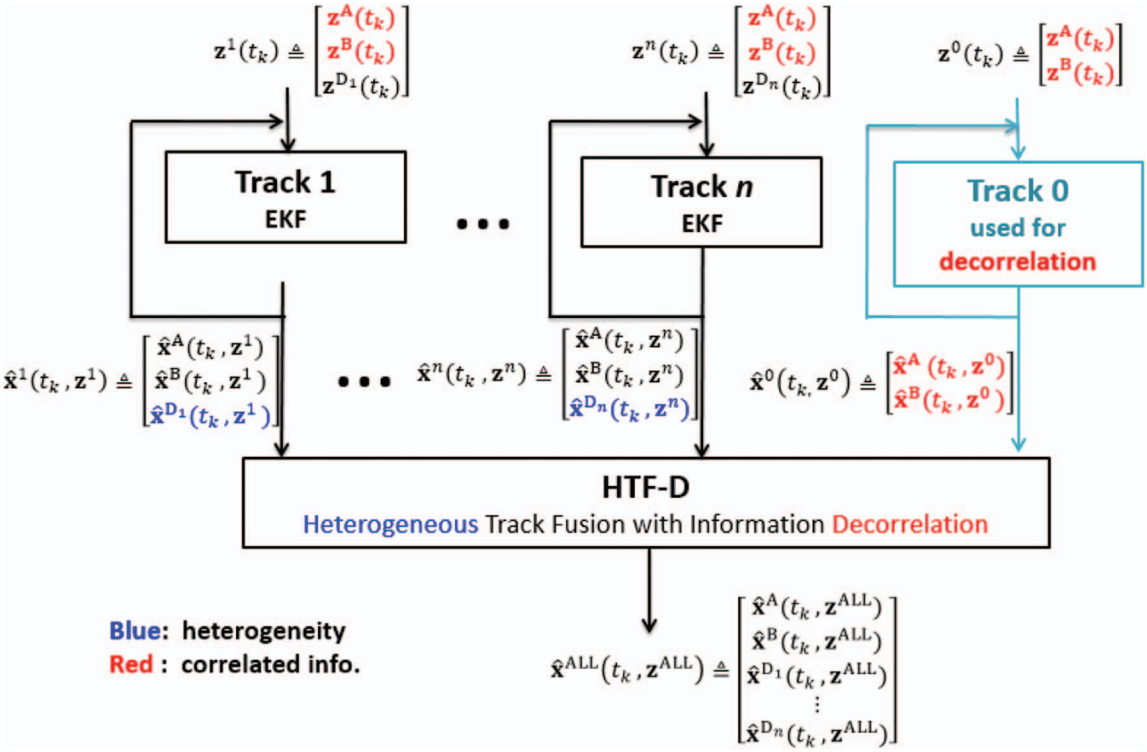



Fig. 2. The flowchart of the HTF-D algorithm. It consists of two steps: 1) Individual target tracking to generate tracks 0–$n$. These tracks are heterogeneous and correlated. 2) Fusion of these heterogeneous and correlated tracks.

where

$$\mathbf{F}^i(T_{k,k-1}) = \begin{bmatrix} \mathbf{F}(T_{k,k-1}) & \mathbf{0}_{4\times4} & \mathbf{0}_{4\times4} \\ \mathbf{0}_{4\times4} & \mathbf{F}(T_{k,k-1}) & \mathbf{0}_{4\times4} \\ \mathbf{0}_{4\times4} & \mathbf{0}_{4\times4} & \mathbf{F}(T_{k,k-1}) \end{bmatrix} \tag{47}$$

is the state transition matrix, $\mathbf{h}^i[\cdot]$ is the measurement function that follows from (8)–(21), and $\mathbf{v}^i(\cdot)$ and $\mathbf{w}^i(\cdot)$ are the process and measurement noises, respectively. The extended Kalman filter is used for the dynamic estimation. The $i$th individual track state estimate is

$$\hat{\mathbf{x}}^i(t_k, \mathbf{z}^i) = [\hat{\mathbf{x}}^{\mathrm{A}}(t_k, \mathbf{z}^i)' \ \hat{\mathbf{x}}^{\mathrm{B}}(t_k, \mathbf{z}^i)' \ \hat{\mathbf{x}}^{\mathrm{D}_i}(t_k, \mathbf{z}^i)']', \tag{48}$$

where $\mathbf{z}^i$ indicates that the state is contributed by the measurement $\mathbf{z}^i$ consisting of $\mathbf{z}^{\mathrm{A}}, \mathbf{z}^{\mathrm{B}}$, and $\mathbf{z}^{\mathrm{D}_i}$.

It can be seen that the individual state estimates $\hat{\mathbf{x}}^1(t_k), \ldots, \hat{\mathbf{x}}^n(t_k)$ are heterogeneous and their errors will be highly correlated. The heterogeneity of the individual track states shown in blue in Fig. 2 is due to different $\mathbf{x}^{\mathrm{D}_1}, \ldots, \mathbf{x}^{\mathrm{D}_n}$ in the states. The correlation shown in red in Fig. 2 is caused by the common information from the transmitter measurements $\mathbf{z}^{\mathrm{A}}$ and $\mathbf{z}^{\mathrm{B}}$.

To quantify the correlation among the individual state estimates, we add an additional reference track 0. This reference track is contributed by the transmitter measurements (the common measurements) $\mathbf{z}^{\mathrm{A}}$ and $\mathbf{z}^{\mathrm{B}}$ only. The state and measurement vectors of the reference track are defined as

$$\mathbf{x}^0(t_k) = [\mathbf{x}^{\mathrm{A}}(t_k)' \ \mathbf{x}^{\mathrm{B}}(t_k)']', \tag{49}$$

$$\mathbf{z}^0(t_k) = [\mathbf{z}^{\mathrm{A}}(t_k)' \ \mathbf{z}^{\mathrm{B}}(t_k)']'. \tag{50}$$

The NCV model is applied for the reference track estimation. The estimate state is

$$\hat{\mathbf{x}}^0(t_k, \mathbf{z}^0) = [\hat{\mathbf{x}}^{\mathrm{A}}(t_k, \mathbf{z}^0)' \ \hat{\mathbf{x}}^{\mathrm{B}}(t_k, \mathbf{z}^0)']', \tag{51}$$

where $\mathbf{z}^0$ indicates that the state is contributed by the measurement $\mathbf{z}^0$, namely, $\mathbf{z}^{\mathrm{A}}$ and $\mathbf{z}^{\mathrm{B}}$ only.

The individual track initiation uses the iterated least squares algorithm [18], [19] on a batch of measurements.

*B. Individual Heterogeneous Track Fusion*

This section shows how to combine the individual track states into one fused state estimate

$$\hat{\mathbf{x}}^{\mathrm{ALL}}(t_k, \mathbf{z}^{\mathrm{ALL}}) = [\hat{\mathbf{x}}^{\mathrm{A}}(t_k, \mathbf{z}^{\mathrm{ALL}})' \ \hat{\mathbf{x}}^{\mathrm{B}}(t_k, \mathbf{z}^{\mathrm{ALL}})' \ \hat{\mathbf{x}}^{\mathrm{D}_1}(t_k, \mathbf{z}^{\mathrm{ALL}})' \ \cdots \ \hat{\mathbf{x}}^{\mathrm{D}_n}(t_k, \mathbf{z}^{\mathrm{ALL}})']', \tag{52}$$

where $\mathbf{z}^{\mathrm{ALL}}$ indicates that the state is contributed by all transmitter and target measurements.

To fuse these highly correlated and heterogeneous tracks, we need to examine the information content of each individual track estimate. Fig. 3 shows the details for a scenario with three targets. For a particular individual track $i$, its information vector and information matrix are

$$\hat{\mathbf{y}}^i(t_k, \mathbf{z}^i) = \mathbf{P}^i(t_k, \mathbf{z}^i)^{-1}\hat{\mathbf{x}}^i(t_k, \mathbf{z}^i), \tag{53}$$

$$\mathbf{Y}^i(t_k, \mathbf{z}^i) = \mathbf{P}^i(t_k, \mathbf{z}^i)^{-1}. \tag{54}$$

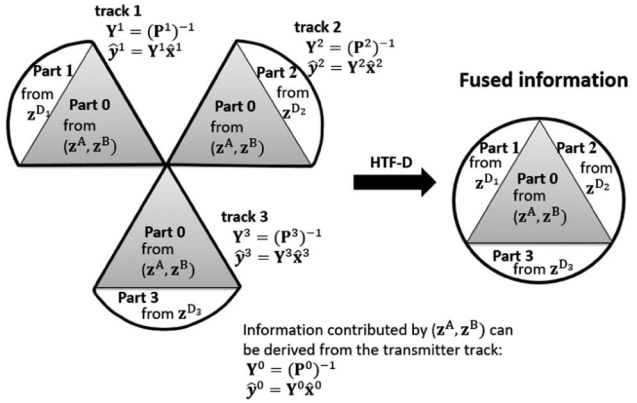



Fig. 3. Information contribution distribution in a three-target scenario.

The above can be separated into two parts. Part **0** (the gray area in Fig. 3) is contributed by the transmitter measurements

$$\mathbf{z}^{\mathrm{A}}(t_1), \ldots, \mathbf{z}^{\mathrm{A}}(t_k), \tag{55}$$

$$\mathbf{z}^{\mathrm{B}}(t_1), \ldots, \mathbf{z}^{\mathrm{B}}(t_k), \tag{56}$$

and Part **i** (the white area in Fig. 3, $\mathbf{i} \in \{\mathbf{1}, \mathbf{2}, \mathbf{3}\}$) is contributed by the particular target measurement

$$\mathbf{z}^{\mathrm{D}_i}(t_1), \ldots, \mathbf{z}^{\mathrm{D}_i}(t_k). \tag{57}$$

The fused information (the full circle shown in Fig. 3) is the sum of Parts **0**, **1**, **2**, and **3**. Part **0** cannot be used more than once as it is from the same source.

The key challenge is to separate the information corresponding to the individual track $i$ into Part **0** and Part **i**. This can be done through the information provided by the reference track 0, which is contributed by the transmitter measurements only, and it is actually equivalent to Part **0** embedded in each individual track. The information vector and matrix of Part **i** in track $i$ can then be decorrelated by

$$\hat{\mathbf{y}}^{\mathbf{i},i}(t_k, \mathbf{z}^{\mathrm{D}_i}) = \hat{\mathbf{y}}^i(t_k, \mathbf{z}^i) \ominus \hat{\mathbf{y}}^0(t_k, \mathbf{z}^0), \tag{58}$$

$$\mathbf{Y}^{\mathbf{i},i}(t_k, \mathbf{z}^{\mathrm{D}_i}) = \mathbf{Y}^i(t_k, \mathbf{z}^i) \ominus \mathbf{Y}^0(t_k, \mathbf{z}^0), \tag{59}$$

where $\ominus$ is heterogeneous "information subtraction." Since $[\hat{\mathbf{y}}^i(\cdot), \hat{\mathbf{Y}}^i(\cdot)]$ with $i = 1, \ldots, n$ and $[\hat{\mathbf{y}}^0(\cdot), \hat{\mathbf{Y}}^0(\cdot)]$ are heterogeneous (of dimensions 12 and 8, respectively), the subtraction needs homogenization. The reference track has no information about $\mathbf{x}^{\mathrm{D}_i}$; 0s are used to pad the missing elements in $[\hat{\mathbf{y}}^0(\cdot), \hat{\mathbf{Y}}^0(\cdot)]$. The information subtraction in (58)–(59) is rewritten as

$$\begin{aligned}\hat{\mathbf{y}}^{\mathbf{i},i}(t_k, \mathbf{z}^{\mathrm{D}_i}) &= \hat{\mathbf{y}}^i(t_k, \mathbf{z}^i) \ominus \hat{\mathbf{y}}^0(t_k, \mathbf{z}^0) \\ &= \begin{bmatrix} \hat{\mathbf{y}}^{\mathrm{A}}(t_k, \mathbf{z}^i) \\ \hat{\mathbf{y}}^{\mathrm{B}}(t_k, \mathbf{z}^i) \\ \hat{\mathbf{y}}^{\mathrm{D}_i}(t_k, \mathbf{z}^i) \end{bmatrix} \ominus \begin{bmatrix} \hat{\mathbf{y}}^{\mathrm{A}}(t_k, \mathbf{z}^0) \\ \hat{\mathbf{y}}^{\mathrm{B}}(t_k, \mathbf{z}^0) \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{y}}^{\mathrm{A}}(t_k, \mathbf{z}^i) \\ \hat{\mathbf{y}}^{\mathrm{B}}(t_k, \mathbf{z}^i) \\ \hat{\mathbf{y}}^{\mathrm{D}_i}(t_k, \mathbf{z}^i) \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{y}}^{\mathrm{A}}(t_k, \mathbf{z}^0) \\ \hat{\mathbf{y}}^{\mathrm{B}}(t_k, \mathbf{z}^0) \\ \mathbf{0}_{4\times 1} \end{bmatrix}, \end{aligned} \tag{60}$$

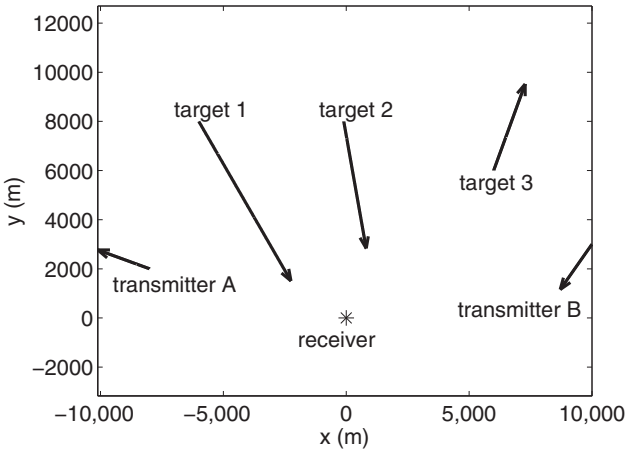



Fig. 4. Scenario 1.

$$\begin{aligned}\mathbf{Y}^{\mathbf{i},i}(t_k, \mathbf{z}^{\mathrm{D}_i}) &= \mathbf{Y}^i(t_k, \mathbf{z}^i) \ominus \mathbf{Y}^0(t_k, \mathbf{z}^0) \\ &= \hat{\mathbf{Y}}^i(t_k, \mathbf{z}^i) - \begin{bmatrix} \hat{\mathbf{Y}}^0(t_k, \mathbf{z}^0) & \mathbf{0}_{8\times 4} \\ \mathbf{0}_{4\times 8} & \mathbf{0}_{4\times 4} \end{bmatrix}. \end{aligned} \tag{61}$$

After all individual track information is decorrelated into two parts, the fused the information vector and matrix are computed as

$$\begin{aligned}\hat{\mathbf{y}}^{\mathrm{ALL}}(t_k, \mathbf{z}^{\mathrm{ALL}}) &= \hat{\mathbf{y}}^0(t_k, \mathbf{z}^0) \oplus \hat{\mathbf{y}}^{\mathbf{1},1}(t_k, \mathbf{z}^{\mathrm{D}_1}) \oplus \cdots \\ &\quad \oplus \hat{\mathbf{y}}^{\mathbf{n},n}(t_k, \mathbf{z}^{\mathrm{D}_n}), \end{aligned} \tag{62}$$

$$\begin{aligned}\hat{\mathbf{Y}}^{\mathrm{ALL}}(t_k, \mathbf{z}^{\mathrm{ALL}}) &= \hat{\mathbf{Y}}^0(t_k, \mathbf{z}^0) \oplus \hat{\mathbf{Y}}^{\mathbf{1},1}(t_k, \mathbf{z}^{\mathrm{D}_1}) \oplus \cdots \\ &\quad \oplus \hat{\mathbf{Y}}^{\mathbf{n},n}(t_k, \mathbf{z}^{\mathrm{D}_n}), \end{aligned} \tag{63}$$

where $\oplus$ is heterogeneous "information addition," which is implemented through information homogenization as done in the heterogeneous "information subtraction" before, namely, to pad with 0s the missing elements.

The fused vector and matrix are then computed as

$$\hat{\mathbf{x}}(t_k, \mathbf{z}^{\mathrm{ALL}}) = \mathbf{Y}(t_k, \mathbf{z}^{\mathrm{ALL}})^{-1}\hat{\mathbf{y}}(t_k, \mathbf{z}^{\mathrm{ALL}}), \tag{64}$$

$$\mathbf{P}(t_k, \mathbf{z}^{\mathrm{ALL}}) = \mathbf{Y}(t_k, \mathbf{z}^{\mathrm{ALL}})^{-1}. \tag{65}$$

The individual transmitter and target state estimates and error covariances can then be obtained as the appropriate parts/blocks from $\hat{\mathbf{x}}(t_k, \mathbf{z}^{\mathrm{ALL}})$ and $\mathbf{P}(t_k, \mathbf{z}^{\mathrm{ALL}})$.

Since the HTF-D does not discard or ignore any information, it provides the optimal estimates. Furthermore, it does not have the drawbacks of the "optimal" algorithm,[3] as it does not estimate a big augmented state.

[3] In the sequel, we shall use the designation "optimal" for the optimal algorithm when it suffers from numerical problem, due its large augmented state.

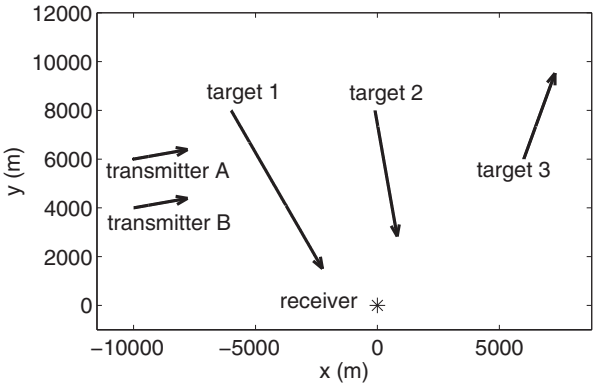



Fig. 5. Scenario 2.

## V. SIMULATION RESULTS

### A. Test Scenarios

We consider two test scenarios. Scenario 1 is shown in Fig. 4. The receiver stays at position $(0, 0)$. The two transmitters A and B move with speed of 3 m/s. Their headings are $-70^{\circ}$ and $-145^{\circ}$ (clockwise from True North), respectively. There are three targets. Targets 1 and 2 are approaching to the receiver with speeds of 10 and 7 m/s, respectively. Target 3 is moving away from the receiver with speed of 5 m/s. The headings of the three targets are $150^{\circ}$, $170^{\circ}$, and $20^{\circ}$, respectively. This scenario has good observability. The two transmitters are well separated,[4] and targets are not located in the blind zone of the multistatic system.[5]

Scenario 2 is shown in Fig. 5. The receiver location and three target trajectories are the same as in Scenario 1, but the transmitters move close to each other. The observability of this scenario is not as good as that of Scenario 1, as the two transmitters are not well separated.

In both scenarios, transmitters A and B emit asynchronous pulse signals with starting times 1 and 10 s, respectively. They have the same pulse interval 30 s. The total simulation duration is 750 s. The receiver receives the pulse signals with the appropriate signal propagation delays (the delays are not shown for simplicity) at

$$\{1\,\text{s}\ \ 31\,\text{s}\ \ \ldots\ \ 721\,\text{s}\}\ \text{from transmitter A}, \tag{66}$$

$$\{10\,\text{s}\ \ 40\,\text{s}\ \ \ldots\ \ 730\,\text{s}\}\ \text{from transmitter B}. \tag{67}$$

The bistatic range measurement error standard deviation is $\sigma_r = 20$ m in both scenarios. The bearing measurement error standard deviation is $\sigma_b = 1^{\circ}$ in Scenario 1, and it increases to $\sigma_b = 1.5^{\circ}$ in Scenario 2 to make estimation more difficult. The probability of detection in both scenarios is 0.8. Two random false alarms were added at each time. A preprocessing with a Kalman filter with 2-D assignment data association is used to track the bistatic range and bearing to filter out the false alarms. The measurements that passed the preprocessing will be used for trajectory estimation. The preprocessing details can be found in [22], where the approach is used to filter false alarms in the measurements of bearing and Doppler frequency.

### B. Algorithms Used in the Simulation Test

Four algorithms are tested for comparison. They are as follows:

- *Simple*: This approach estimates the transmitter and target trajectories individually. It estimates the transmitter states first, and then the target states. The drawback is that the transmitter state estimation errors are not taken into consideration in the target estimation. The algorithm details can be found in Section 3.2.
- *CI*: This approach estimates the two transmitters using an augmented state first, and then estimates target states individually. The transmitter estimation errors are taken into consideration in the target state estimation (but only as extra white measurement noise), and some cross-correlations are ignored. The details can be found in Section 3.3.
- *Optimal*: This approach estimates a large augmented state consisting of all transmitter and target states. It can provide the optimal estimates. However, the large augmented state introduces some numerical issues and increases the system sensitivity to the noise. The algorithm details can be found in Section 3.1.
- *HTF-D*: This is the algorithm proposed in the paper. It tracks targets individually (without a large augmented state), and obtains the optimal estimation through heterogeneous track fusion. The details are given in Section 4.

In the simulation tests, the process noise PSD $q$ in (7) is set to $10^{-6}$ m$^2$/s$^3$. The batch measurements for track initiation are taken from time 0 to 100 s.

### C. Results of Scenario 1

We conducted 100 Monte Carlo runs on the first scenario using the four algorithms mentioned earlier. The position estimate root mean square errors (RMSEs) of the two transmitters and three targets versus time are presented in Figs. 6–10, respectively.

From the test results, we can observe that the simple algorithm (which ignores the transmitter estimation errors) provides the worst accuracy. The CI algorithm (which considers the transmitter estimation errors as white, and ignores some cross-correlations) is better than

[4]The problem is not observable when the two transmitters are located on the same line from the receiver (see Appendix A).

[5]When a target and a transmitter are located on similar bearings from the receiver, the weak indirect path signal is blocked by the strong direct path signal. This creates a detection blind zone for a target near the line between the transmitter and the receiver. The target observability in the blind zone is also very marginal due to the geometry.

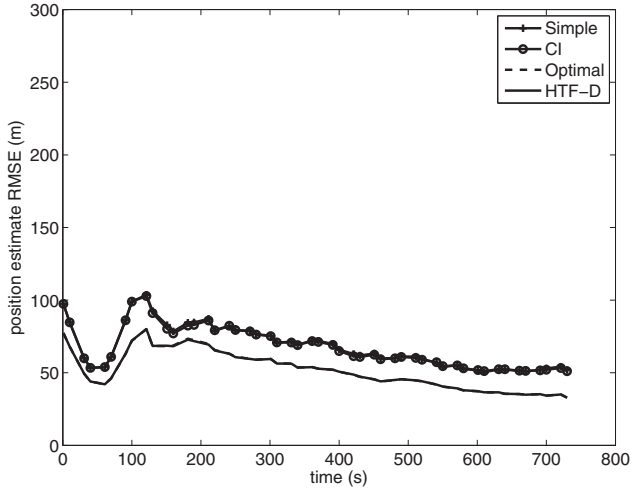



Fig. 6. Position estimate RMSE of transmitter A versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are overlapping each other as they provide the same result.

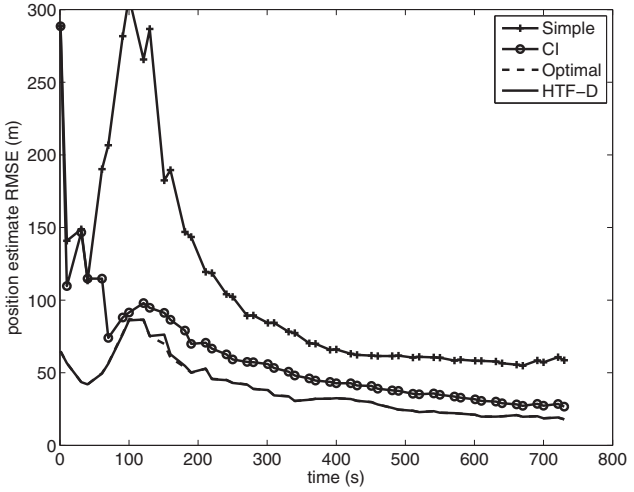



Fig. 8. Position estimate RMSE of target 1 versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are almost overlapping each other as they provide similar results.

the simple algorithm. The HTF-D and the optimal algorithms provide the best estimates and have practically the same results. Their accuracy (solid and dash) lines overlap in Figs. 6–10, except a slight difference in Fig. 8 (around time 150 s), which is due to numerical issues of Matlab computation. So, we can say that the HTF-D achieves the optimal performance in this scenario.

To evaluate the consistency of the algorithms, the average normalized estimation error squared (NEES) of position is evaluated. The average position NEES [19] at time $t_k$ for $N = 100$ Monte Carlo runs is

$$\bar{\epsilon}(t_k) = \frac{1}{Nn} \sum_{i=1}^{N} \tilde{\mathbf{x}}^i_{1:2}(t_k)' \mathbf{P}^i_{1:2,1:2}(t_k)^{-1} \tilde{\mathbf{x}}^i_{1:2}(t_k), \qquad (68)$$

where $n = 2$ is the dimension of the position state, $\mathbf{P}_{1:2,1:2}(t_k)$ is the position estimation error covariance at time $t_k$, $i$ is the run index,

$$\tilde{\mathbf{x}}_{1:2}(t_k) = \mathbf{x}_{1:2}(t_k) - \hat{\mathbf{x}}_{1:2}(t_k) \qquad (69)$$

is the position estimation error at time $t_k$, and $\tilde{\mathbf{x}}_{1:2}(t_k)$ and $\mathbf{x}_{1:2}(t_k)$ are the position estimate and ground truth at time $t_k$, respectively. The two-sided 95% probability region for an $Nn = 200$ degrees of freedom ($N = 100, n = 2$) chi-square random variable is [163, 241]. Dividing by 200, the average NEES should be in the interval [0.815, 1.205]. Figs. 11–15 show the average position NEES versus time of the CI, and optimal and HTF-D algorithms for the two transmitters and three targets, respectively. The NEES of the simple algorithm (around 50–150) is too far away from the boundary, so it is not displayed in the figures. It can be seen that the NEESs of the HTF-D and optimal algorithms are all within the 95% probabil-

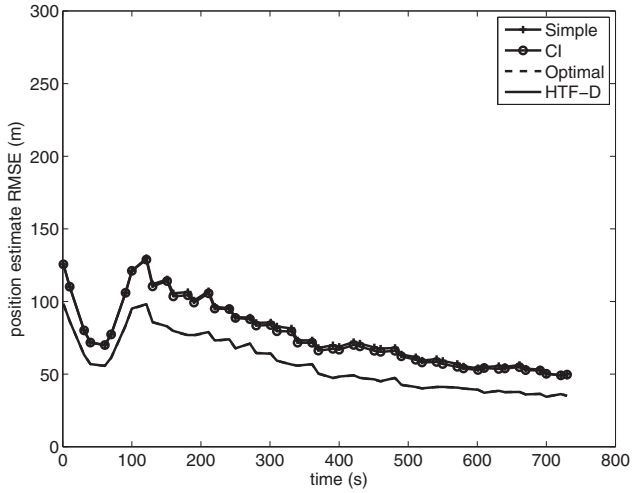



Fig. 7. Position estimate RMSE of transmitter B versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are overlapping each other as they provide the same result.

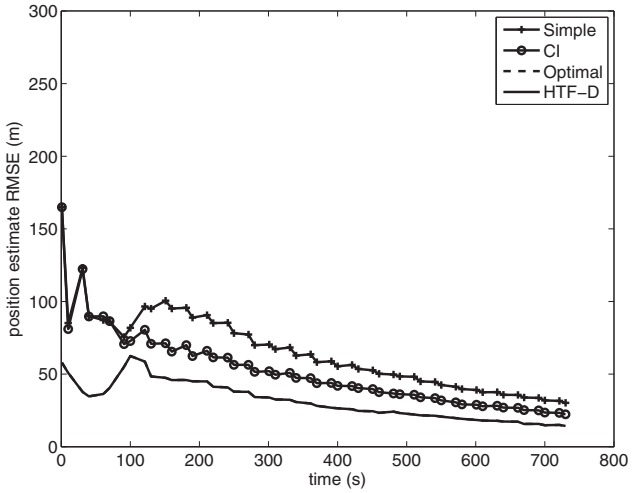



Fig. 9. Position estimate RMSE of target 2 versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are overlapping each other as they provide the same result.

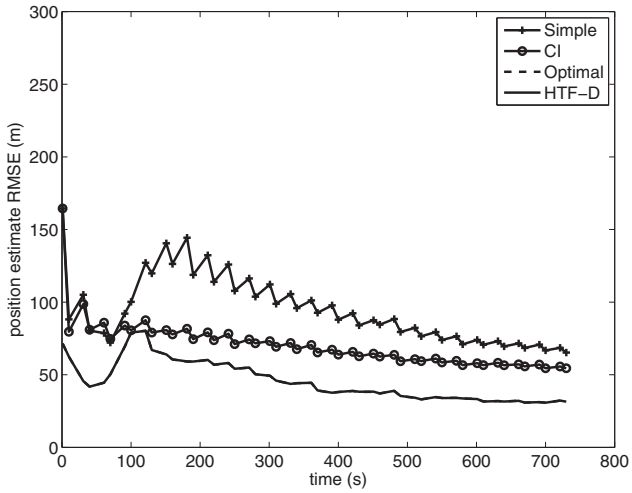



Fig. 10. Position estimate RMSE of target 3 versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are overlapping each other as they provide the same result.

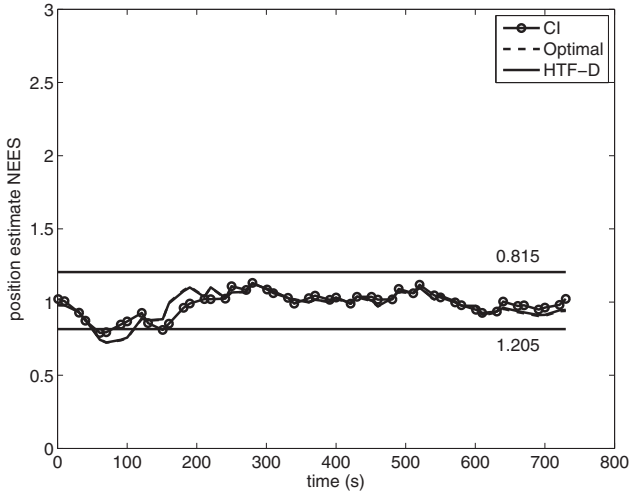



Fig. 11. Position NEES of transmitter A versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are overlapping each other as they provide the same result.

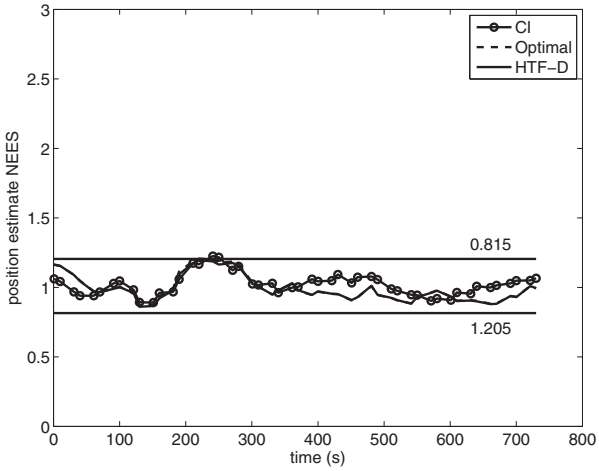



Fig. 12. Position NEES of transmitter B versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are overlapping each other as they provide the same result.

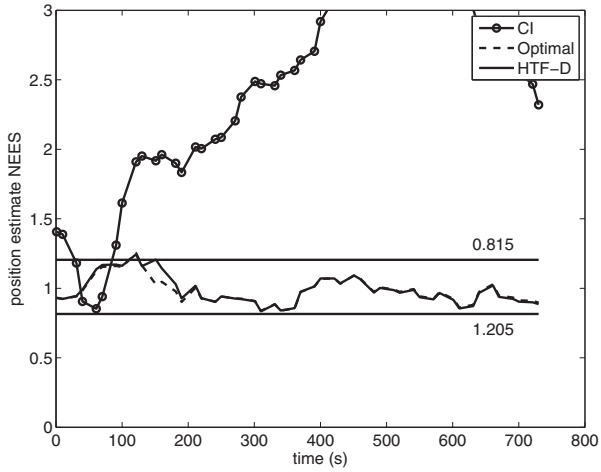



Fig. 13. Position NEES of target 1 versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are almost overlapping each other as they provide similar results.

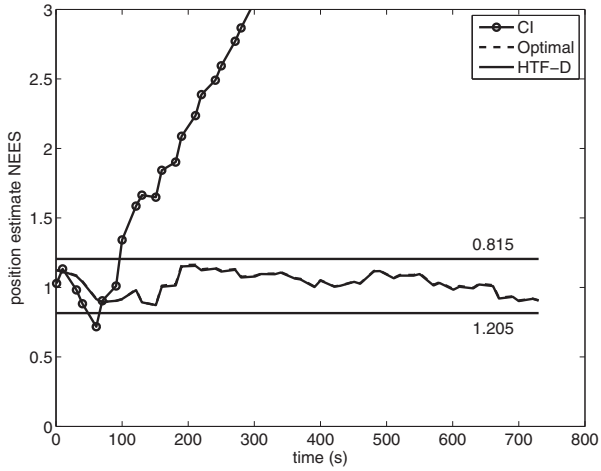



Fig. 14. Position NEES of target 2 versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are overlapping each other as they provide the same result.

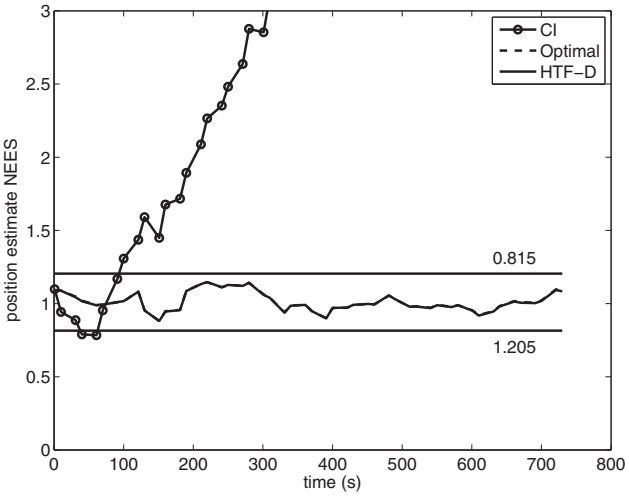



Fig. 15. Position NEES of target 3 versus time from 100 Monte Carlo runs (Scenario 1). The results of the HTF-D (solid line) and optimal (dashed line) algorithms are overlapping each other as they provide the same result.

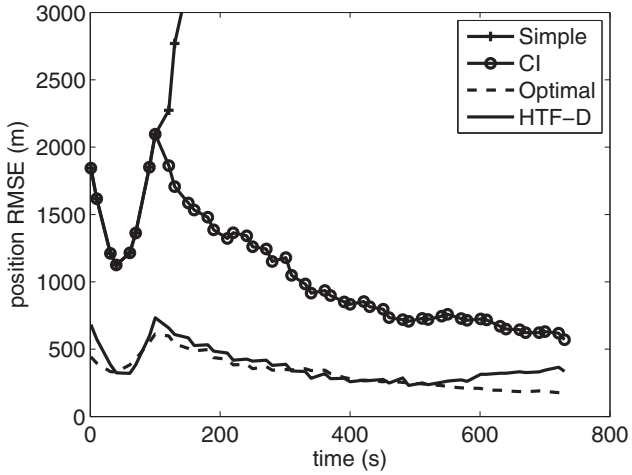



Fig. 16. Position estimate RMSE of transmitter A versus time from 100 Monte Carlo runs (Scenario 2).

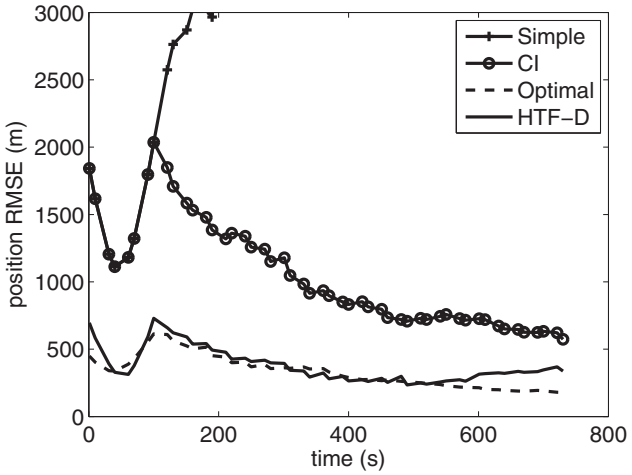



Fig. 17. Position estimate RMSE of transmitter B versus time from 100 Monte Carlo runs (Scenario 2).

ity region [0.815, 1.205]. This shows that both the HTF-D and optimal algorithms yield consistent estimation results.

For the CI algorithm, the transmitter position NEESs are within the 95% probability region (see Figs. 11 and 12). However, the target position NEESs are much larger than the upper bound after track initiation (see Figs. 13–15, $t > 100$ s). This is because the transmitters are estimated using the augmented state given in (28). The estimation error cross-covariance between the transmitters is taken into consideration. The transmitter estimates are therefore yield consistent estimates. However, the target state estimation treats the transmitter position error as white noise, but this is actually not white. The cross-correlations with the historical transmitter estimation errors are ignored. This leads to the target state estimation error covariances smaller than their actual values. The NEESs are therefore above the upper bound.

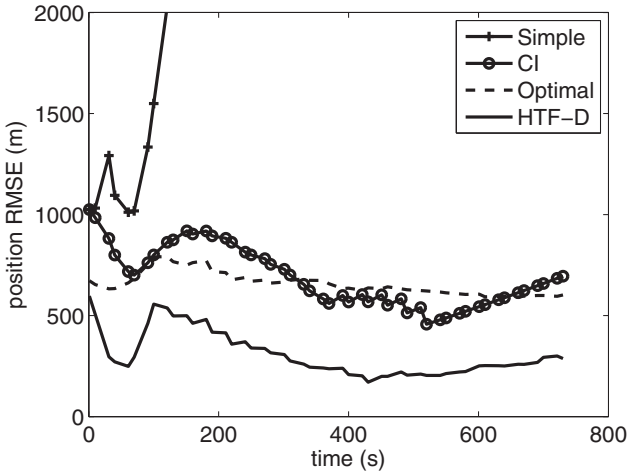



Fig. 18. Position estimate RMSE of target 1 versus time from 100 Monte Carlo runs (Scenario 2).

## D. Results of Scenario 2

We also conducted 100 Monte Carlo runs for Scenario 2, which has marginal observability due to its geometry and larger measurement error ($\sigma_b = 1.5^{\circ}$) than Scenario 1. The position estimation RMSEs of the two transmitters and three targets versus time are presented in Figs. 16–20, respectively.

It can be seen that the simple algorithm diverges for all the transmitter and target estimates. The CI approach still shows noticeable gaps compared to the HTF-D.

An interesting observation is that the "optimal" algorithm performs worse than the HTF-D for the target 1 estimation shown in Fig. 18. This is because the "optimal" algorithm has a state vector with large size ($5 \times 4 = 20$, for the 2 transmitters plus 3 targets, each of them has 4 state elements). This creates more numerical problems than a system with a smaller state, and causes the "optimal" algorithm to yield larger errors than the HTF-D algorithm, especially when the measurement noise is large.

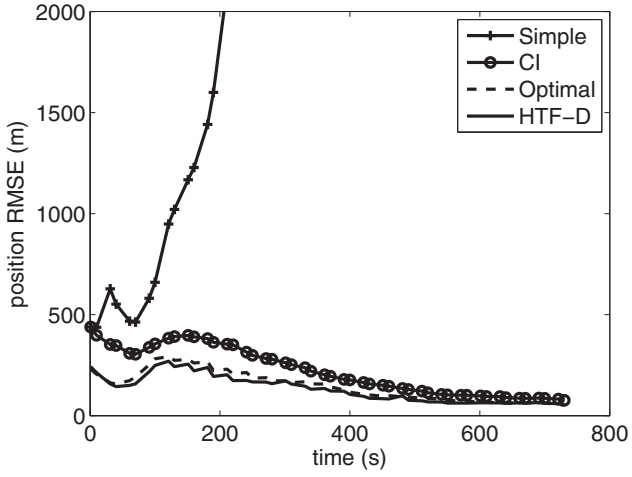



Fig. 19. Position estimate RMSE of target 2 versus time from 100 Monte Carlo runs (Scenario 2).

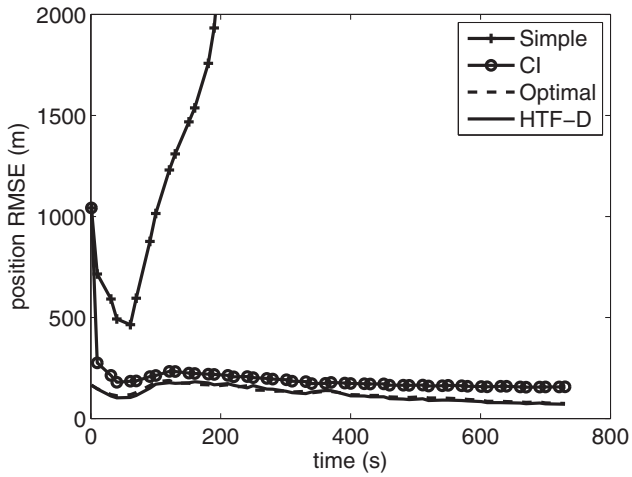



Fig. 20. Position estimate RMSE of target 3 versus time from 100 Monte Carlo runs (Scenario 2).

These results show that the "optimal" algorithm is not robust when the measurement error is large and observability is not good. The number of targets in the system also affects its estimation quality.

## VI. CONCLUSIONS

This paper deals with target tracking using a new configuration of a non-cooperative multistatic system where the positions of moving transmitters are unknown. The new HTF-D algorithm was developed to track each target individually, and fuse these correlated and heterogeneous tracks using a novel decorrelation approach. Since it does not lose any information, it can achieve the accuracy of the optimal algorithm. As the HTF-D tracks each target individually, its state size is fixed and small. The drawbacks of the optimal algorithm (such as sensitivity to higher noise and difficulty to implement in real systems) are avoided.

Simulation results showed that the HTF-D indeed obtained the accuracy of the optimal algorithm. The NEES results showed that the HTF-D is consistent statistically as well. We also observed that the "optimal" algorithm yielded larger estimation error than the HTF-D in Scenario 2, where the measured bearing error is larger and observability is not good. This showed that the HTF-D is more robust than the "optimal" algorithm.

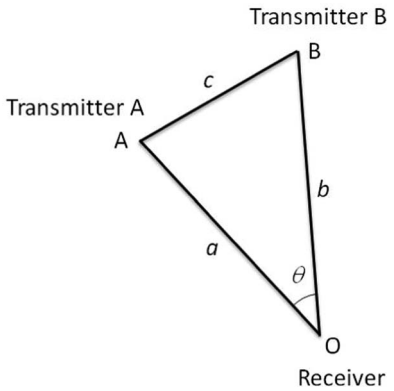



Fig. A1. The triangle formed by the receiver and two transmitters.

## APPENDIX A OBSERVABILITY

It is known that a multistatic tracking system is observable when the positions of transmitters and receiver are known. If both transmitters' positions can be solved from the measurements $r^{AB}$, $r^{BA}$, $b^{A}$, and $b^{B}$ [defined in (11), (12), and (21), respectively] uniquely, the problem defined in Section 2 is observable.

Without loss of generality, the problem can be simplified as to determine the triangle $\triangle$OAB shown in Fig. A1 from $r^{AB}$, $r^{BA}$, $b^{A}$, and $b^{B}$. This can be done in one time cycle. The orientations of $\overline{OA}$ and $\overline{OB}$ are fixed by $b^{A}$ and $b^{B}$, and

$$\theta = b^{A} - b^{B} \tag{A.1}$$

is known. The problem is then of solving for $a$ and $b$ from $r^{AB}$ and $r^{BA}$ using the nonlinear model

$$\begin{bmatrix} r^{AB} \\ r^{BA} \end{bmatrix} = \mathbf{h}_0 \left( \begin{bmatrix} a \\ b \end{bmatrix} \right) + \mathbf{w}_0, \tag{A.2}$$

where $\mathbf{w}_0$ is the error of $[r^{AB}\ r^{BA}]'$,

$$h_{0,1}(\cdot) = c + a - b, \tag{A.3}$$

$$h_{0,2}(\cdot) = c + b - a, \tag{A.4}$$

and

$$c = \sqrt{a^2 + b^2 - 2ab\cos\theta}. \tag{A.5}$$

This problem is observable when

$$\det(\mathbf{H}_0) \neq 0, \tag{A.6}$$

where

$$\mathbf{H}_0 = [\nabla_{\mathbf{y}} \mathbf{h}_0(\mathbf{y})']' \tag{A.7}$$

with $\mathbf{y} = [a\ b]'$. $\mathbf{H}_0$ is derived as

$$\mathbf{H}_0 = \begin{bmatrix} \dfrac{a - b\cos\theta + c}{c} & \dfrac{b - a\cos\theta - c}{c} \\ \dfrac{a - b\cos\theta - c}{c} & \dfrac{b - a\cos\theta + c}{c} \end{bmatrix} \tag{A.8}$$

and its determinant is

$$\det(\mathbf{H}_0) = \frac{2(a+b)(1-\cos\theta)}{c}. \tag{A.9}$$

Since $(a + b) > 0$, $\det(\mathbf{H}_0)$ is 0 when $(1 - \cos\theta)/c$ is 0. This latter condition occurs only when $\theta$ is $0^{o}$ (or $180^{o}$). Thus, the problem is observable when the two transmitters are not located on the same line of sight from the receiver.

## REFERENCES

[1] V. J. Aidala
"Kalman filter behavior in bearings-only tracking applications,"

*IEEE Trans. Aerosp. Electron. Syst.*, vol. 1, no. 1, pp. 29–39, Jul. 1978.

[2] H. A. J. Huang, Y. Bar-Shalom, R. Yang, and G. W. Ng
"Tracking a maneuvering target using two heterogeneous passive sensors on a single stationary platform with IMM estimation,"
*J. Adv. Inf. Fusion*, vol. 12, no. 1, pp. 110–124, Jun. 2017.

[3] C. Jauffret and Y. Bar-Shalom
"Track formation with bearing and frequency measurements in clutter,"
*IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 6, pp. 999–1009, Nov. 1990.

[4] C. Jauffret and A.-C. Pignol
"Target motion analysis by inverse triangulation,"
in *Proc. 18th Int. Conf. Inf. Fusion*, Washington, DC, USA, Jul. 2015.

[5] A. G. Lindgren and K. F. Gong
"Position and velocity estimation via bearing observations,"
*IEEE Trans. Aerosp. Electron. Syst.*, vol. 14, no. 4, pp. 564–577, Jul. 1978.

[6] J. M. Passerieux, D. Pillon, P. Blanc-Benon, and C. Jauffret
"Target motion analysis with bearing and frequencies measurements,"
in *Proc. 22nd Asilomar Conf.*, Pacific Grove, CA, USA, Nov. 1988.

[7] R. Yang, Y. Bar-Shalom, and G. W. Ng
Bearings-only Tracking with Fusion from Heterogenous Passive Sensors: ESM/EO and Acoustic.
In *Proceedings of the 18th International Conference on Information Fusion*, Washington, DC, July 2015.

[8] R. Yang, Y. Bar-Shalom, and G. W. Ng
"Bearings-only tracking with tusion from heterogenous passive sensors: ESM/EO and acoustic,"
*J. Adv. Inf. Fusion*, vol. 12, no. 3, pp. 3–17, Jul. 2017.

[9] D. Grimmett, S. Coraluppi, R. B. La Cour, C. G. Hempel, T. Lang, P. A. M. de Theije, and P. Willett
"MSTWG multistatic tracker evaluation using simulated scenario data sets,"
in *Proc. 11th Int. Conf. Inf. Fusion*, Cologne, Germany, Jul. 2008.

[10] S. Benen, D. Maiwald, and H. Schmidt-Schierhorn
"Low frequency towed active sonar (LFTAS) in multistatic applications,"
*GI Jahrestagung, LNI*, vol. 154, pp. 2413–2421, 2009.

[11] M. Broetje and K. Pikora
"Parameter estimation for multistatic active sonar using extended fixed points,"
in *Proc. 18th Int. Conf. Inf. Fusion*, Washington, DC, USA, Jul. 2015.

[12] M. Michaelis, M. Broetje, and F. Ehlers
"Case study: limitations of parameter estimation for non-cooperative multistatic active sonar by association ambiguities,"
in *Proc. Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, Oct. 2014, pp. 1–6.

[13] V. Algeier, B. Demissie, W. Koch, and R. Thoma
"State space initiation for blind mobile terminal position tracking,"
*EURASIP J. Adv. Signal Process.*, vol. 1, pp. 1–14, 2008.

[14] R. Mendrzik, F. Meyer, G. Bauch, and M. Win
"Enabling situational awareness in millimeter wave massive MIMO systems,"
*IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 1196–1211, Sep. 2019.

[15] W. Xu, F. Quitin, M. Leng, W. P. Tay, and S. G. Razul
"Distributed localization of a RF target in NLOS environments,"
*IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1317–1330, Jul. 2015.

[16] M. Michaelis
"Bistatic simultaneous transmitter localization and mapping,"
in *Proc. Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, Oct. 2015, pp. 1–6.

[17] R. Yang and Y. Bar-Shalom
"Adaptive target tracking using multistatic sensor with unknown moving transmitter positions,"
in *Proc. 7th IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Curacao, Dec. 2017.

[18] R. Yang, Y. Bar-Shalom, and G. W. Ng
"Target tracking using an asynchronous multistatic sensor system with unknown transmitter positions,"
in *Proc. 21st Int. Conf. Inf. Fusion*, Cambridge, U.K., Jul. 2018.

[19] Y. Bar-Shalom, P. Willett, and X. Tian
*Tracking and Data Fusion: A Handbook of Algorithms*. Storrs, CT, USA: YBS Publishing, 2011.

[20] P. Willett and S. Coraluppi
"Application of the MLPDA to bistatic sonar,"
in *Proc. 2005 IEEE Aerosp. Conf.*, Big Sky, MT, USA, 2005, pp. 2063–2073.

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

[22] R. Yang, Y. Bar-Shalom, and G. W. Ng
"Tracking/fusion and deghosting with Doppler frequency from two passive acoustic sensors,"
in *Proc. 16th Int. Conf. Inf. Fusion*, Istanbul, Turkey, Jul. 2013.

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

**Yaakov Bar-Shalom** was born on May 11, 1941. He received the B.S. and M.S. degrees in electrical engineering from the Technion, Israel Institute of Technology, Haifa, Israel, in 1963 and 1967, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 1970. From 1970 to 1976, he was with the Systems Control, Inc., Palo Alto, CA, USA. He is currently Board of Trustees Distinguished Professor in the Department of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering with the University of Connecticut, Storrs, CT, USA. His current research interests include estimation theory, target tracking, and data fusion. He has authored or coauthored more than 500 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993; translated into Russian, MGTU Bauman, Moscow, Russia, 2011), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate texts *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), *Tracking and Data Fusion* (YBS Publishing, 2011), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He has been consulting to numerous companies and government agencies, and originated the series of *Multitarget-Multisensor Tracking* short courses offered via UCLA Extension, at Government Laboratories, private companies, and overseas. During 1976 and 1977, he was an Associate Editor for the IEEE Transactions on Automatic Control and from 1978 to 1981 as an Associate Editor for *Automatica*. He was General Chairman of the 1985 ACC and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1987–1989, he was a member of the Board of Governors of the IEEE CSS. He was General Chairman of FUSION 2000, President of ISIF in 2000 and 2002, and Vice President for Publications in 2004–2013. In 1987, he was the recipient of the IEEE CSS Distinguished Member Award. Since 1995, he has been a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He was the corecipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002, he was the recipient of the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008, he was the recipient of the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012, the Connecticut Medal of Technology. In 2015, he was the recipient of the Lifetime of Excellence in Information Fusion award from the International Society of Information Fusion, renamed in 2016 as "ISIF YaakovBar-Shalom Award for Lifetime of Excellence in Information Fusion". He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in aerospace engineering based on the citations of his work.

# Asynchronous and Heterogeneous Track-to-Track Fusion with Mapped Process Noise and Cross-Covariance

**KAIPEI YANG**
**YAAKOV BAR-SHALOM**
**PETER WILLETT**

**Track-to-track fusion has been studied extensively for both homogeneous and heterogeneous cases, these cases denoting common and disparate state models. However, as opposed to homogeneous fusion, the cross-covariance for heterogeneous local tracks (LTs) in different state spaces that accounts for the relationship between the process noises of the heterogeneous models seems not to be available in the literature. This work provides the derivation of the cross-covariance for heterogeneous LTs of different dimensions where the local states are related by a nonlinear transformation (with no inverse transformation). First, the relationship between the process noise covariances of the motion models in different state spaces is obtained. The cross-covariance of the local estimation errors is then derived in a recursive form by taking into account the relationship between the local state model process noises. Both the synchronous and asynchronous systems are considered. A linear minimum mean square fusion is carried out for a scenario involving tracks from two LTs: one from an active sensor and one from a passive sensor.**

Manuscript received January 15, 2019; revised May 23, 2019 and August 11, 2019; released for publication June 30, 2020

The authors are with the Department of Electrical and Computer Engineering, University of Connecticut, U-4157, Storrs, CT 06269-4157, USA E-mail: kaipei.yang@uconn.edu, yaakov.bar-shalom@uconn.edu, peter.willett@uconn.edu.
This work was supported by NSWC N00174181004. Associate Editor: Chee-Yee Chong.

1557-6418/20/$17.00 

## I. INTRODUCTION

In a heterogeneous system, the state models used by local sensors are in different state spaces with different dimensions. The fusion for heterogeneous systems needs investigation since it is closely related to the real-world problems. One reason for using distinct system models in the local trackers is the different sensor characteristics—active versus passive. For example, for self-driving vehicle system perception, heterogeneous tracks are inevitable due to different coordinate systems, which are related by a nonlinear transformation. Low-level fusion, or centralized track/fusion (CTF), is characterized by transferring raw data from each sensor to the fusion center (FC). It requires communication with high bandwidth since all the raw data need to be transferred on demand, which is not feasible for most of the practical applications. The track-to-track fusion (T2TF) to be considered is characterized by local tracking at each of the sensors and a fusion combining the tracks from multiple sensors at the FC. For several applications in defense systems and self-driving vehicle systems, T2TF fusion is preferred due to communication constraints.

For T2TF, it is critical to consider the cross-covariance between the track estimation errors of the same target at different local trackers. The fusion of homogeneous local tracks (LTs)—when the state models at two sensors are the same—considering the "common process noise" of the LTs is discussed in [1, Ch. 9]. The work [2] considered homogeneous T2TF with the cross-covariance for the asynchronous case. The fusion of heterogeneous LTs from local sensors that use different state models was presented in [3]; however, it assumed the cross-covariance between the local state estimation errors to be zero since the cross-covariance was not available. The contributions of this paper compared to [3] are as follows: (i) the cross-covariance of the process noises of the heterogeneous models is derived and (ii) the cross-covariance of the heterogeneous estimation errors from the LTs is derived. The recent work [4] deals with the heterogeneous T2TF in 3D using an infrared search and track (IRST) sensor and an air moving target indicator (MTI) radar based on information matrix fusion taking into account the cross-covariance between the LTs. However, the main limitation of [4] is that the authors assumed that the local state vectors have the same dimension with a unique inverse mapping, which is not realistic in most of the heterogeneous T2TF scenarios where the inverse transformation does not exist due to the different state space dimensions.

This work provides the derivation of the cross-covariance for the heterogeneous LTs of different dimensions where the local states are related by a nonlinear transformation without inverse transformation. The heterogeneous T2TF considered here has no information feedback. The relationship between the process noise covariances of the two motion models is presented. The state model process noise covariance

in the smaller state space can be obtained (through a mapping based on the nonlinear relationship) from the state model process noise covariance and the estimate in the larger state space. The cross-covariance of the estimation errors from two LTs is derived in a recursive form by taking into account the relationship between the local state model process noises. Both synchronous and asynchronous systems are considered. In the asynchronous case, where the sensors are having arbitrary sampling times, the fusion happens at the union of the sampling times of the two trackers, that is, with LT-driven communication. The asynchronous T2TF fusion is carried out for a scenario of two tracks of a single target (one from an active sensor and one from a passive sensor) with a linear minimum mean square (LMMSE) fuser in the simulation. The cross-correlation of the estimation errors is shown to be sometimes positive and sometimes negative depending on the sensor-trajectory geometry, which confirms the results in [3] from a Monte Carlo (MC) investigation.

The paper is organized as follows. Section II formulates the heterogeneous fusion problem and derives the relationship of the process noise covariances of the LT models. Section III presents the cross-covariance of the estimation errors. In Section IV, the state and measurement models for both the active sensor and passive sensor are introduced. Section V formulates the LMMSE fuser. Section VI presents the simulation results from MC runs. The summary and conclusions are provided in Section VII.

## II. PROBLEM FORMULATION AND THE RELATIONSHIP BETWEEN THE PROCESS NOISES

### A. Synchronous Case

In the synchronous case, the LTs share the same sampling time and are assumed to have the full rate communication with the FC. Consider the state models at sensors $i$ and $j$ in different state spaces with dimensions $n_{\mathbf{x}}^i$ and $n_{\mathbf{x}}^j$, respectively,

$$\mathbf{x}^i(k+1) = f^i[k, \mathbf{x}^i(k)] + \mathbf{v}^i(k), \tag{1}$$

$$\mathbf{x}^j(k+1) = f^j[k, \mathbf{x}^j(k)] + \mathbf{v}^j(k), \tag{2}$$

and the measurements of dimensions $n_{\mathbf{z}}^i$ and $n_{\mathbf{z}}^j$, respectively,

$$\mathbf{z}^i(k) = h^i[k, \mathbf{x}^i(k)] + \mathbf{w}^i(k), \tag{3}$$

$$\mathbf{z}^j(k) = h^j[k, \mathbf{x}^j(k)] + \mathbf{w}^j(k), \tag{4}$$

where $\mathbf{v}^m(k)$ and $\mathbf{w}^m(k), m = i, j$, are the process noises and measurement noises assumed to be additive, zero mean, and white with corresponding covariance matrices $Q^m(k)$ and $R^m(k)$ $(m = i, j)$. All the noises are also assumed to be mutually independent, except $\mathbf{v}^i$ is correlated with $\mathbf{v}^j$ since they pertain to the motion of the same target, although in different state spaces. The recursion of the cross-covariance $Q^{ij}(k)$ between $\mathbf{v}^i(k)$ and $\mathbf{v}^j(k)$ will be discussed later.

The nonlinear functions $f^m[\cdot, \cdot]$ and $h^m[\cdot, \cdot], m = i, j$, are distinct and may be time varying. The two state vectors have a nonlinear relationship[1]

$$\mathbf{x}^j = \alpha\left[\mathbf{x}^i\right]; \tag{5}$$

with $n_{\mathbf{x}}^i > n_{\mathbf{x}}^j$, it is clear that (5) has no inverse.

Substituting (1) into the above equation yields

$$\mathbf{x}^j(k+1) = \alpha\left[\mathbf{x}^i(k+1)\right] = \alpha\left[f^i[k, \mathbf{x}^i(k)] + \mathbf{v}^i(k)\right]. \tag{6}$$

The vector Taylor series expansion of (6) up to the first-order term is

$$\alpha\left[\mathbf{x}^i(k+1)\right] = \alpha\left[f^i[k, \mathbf{x}^i(k)]\right] + [\nabla_x \alpha(x)']'|_{x=f^i[k,\mathbf{x}^i(k)]}\mathbf{v}^i(k). \tag{7}$$

Thus,

$$\mathbf{v}^j(k) \approx [\nabla_x \alpha(x)']'|_{x=f^i[k,\mathbf{x}^i(k)]}\mathbf{v}^i(k) = A(k)\mathbf{v}^i(k), \tag{8}$$

where

$$A(k) \triangleq [\nabla_x \alpha(x)']'|_{x=f^i[k,\mathbf{x}^i(k)]} \tag{9}$$

is the $(n_{\mathbf{x}}^j \times n_{\mathbf{x}}^i$, i.e., not square) Jacobian corresponding to (8).

Then, $Q^j(k)$, the covariance of the process noise $v^j(k)$, can be expressed using $Q^i(k)$ as follows:

$$Q^j(k) = E\left[\mathbf{v}^j(k)\mathbf{v}^j(k)'\right] = A(k)Q^i(k)A(k)', \tag{10}$$

and the cross-covariance between the two process noises is

$$Q^{ij}(k) = E\left[\mathbf{v}^i(k)\mathbf{v}^j(k)'\right] = Q^i(k)A(k)'. \tag{11}$$

Note that in the estimation problem where the state is not available, the Jacobian (9) will have to be evaluated at the latest estimate.

### B. Asynchronous Case

With LT (local filter/tracker)-driven communication, the fusion of an asynchronous system (i.e., with tracks from radar and infrared/electro-optical sensors) is carried out whenever the FC receives new information. As shown in Fig. 1, sensor $i$ is assumed to be the active one with state vector in the larger state space (of dimension $n_{\mathbf{x}}^i$) and sensor $j$ is the passive one with state vector in the smaller state space (of dimension $n_{\mathbf{x}}^j, n_{\mathbf{x}}^j < n_{\mathbf{x}}^i$). For the FC, the fusion times are equal to the times when new information is obtained. From this figure, we have

$$t_{k+1} = t_{m+1}^j \tag{12}$$

[1]This is in general, and subsumes cases of equivalent states, situations in which one state is a subset of the other, and other more complicated relationships.

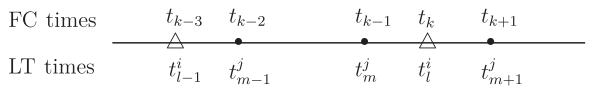



Fig. 1. FC times and LT times in asynchronous T2TF.

and

$$t_k = t_l^i, \tag{13}$$

where $l$ and $m$ denote the respective LT sampling indices.

Since the state error cross-covariance will have to be iterated according to the FC times[2]

$$\{t_k\} = \{t_l^i\} \bigcup \{t_m^j\}, \tag{14}$$

we will develop the relationship between the process noises of the different local states

$$\begin{aligned} &f^j[t_{k+1}, t_k, \mathbf{x}^j(t_k)] + \mathbf{v}^j(t_{k+1}, t_k) \\ &\quad = \alpha\left[f^i[t_{k+1}, t_k, \mathbf{x}^i(t_k)] + \mathbf{v}^i(t_{k+1}, t_k)\right]. \end{aligned} \tag{15}$$

Different from the synchronous case, the nonlinear functions $f^i[\cdot,\cdot,\cdot]$ and $f^j[\cdot,\cdot,\cdot]$ have three arguments: propagation end time, propagation start time, and the state at the propagation start time. The process noises $\mathbf{v}^i(\cdot,\cdot)$ and $\mathbf{v}^j(\cdot,\cdot)$ have two arguments: propagation end time and propagation start time. Following (7) with first-order Taylor expansion, the second term on the left-hand side is given by

$$\mathbf{v}^j(t_{k+1}, t_k) = A(t_k)\mathbf{v}^i(t_{k+1}, t_k), \tag{16}$$

where

$$A(t_k^i) \triangleq \left[\nabla_x \alpha(x)'\right]'\Big|_{x=f^i[t_{k+1},t_k,\mathbf{x}^i(t_k)]} \tag{17}$$

with dimension $n^j \times n^i$.

The covariance matrix of $\mathbf{v}^j(t_{k+1}, t_k)$ can be obtained by

$$\begin{aligned} Q^j(t_{k+1}, t_k) &\triangleq E[\mathbf{v}^j(t_{k+1}, t_k)\mathbf{v}^j(t_{k+1}, t_k)'] \\ &= E[A(t_k)\mathbf{v}^i(t_{k+1}, t_k)\mathbf{v}^i(t_{k+1}, t_k)'A(t_k)'] \\ &= A(t_k)Q^i(t_{k+1}, t_k)A(t_k)'. \end{aligned} \tag{18}$$

The cross-covariance between the process noises is

$$\begin{aligned} Q^{ij}(t_{k+1}, t_k) &\triangleq E[\mathbf{v}^i(t_{k+1}, t_k)\mathbf{v}^j(t_{k+1}, t_k)'] \\ &= E[\mathbf{v}^i(t_{k+1}, t_k)\mathbf{v}^i(t_{k+1}, t_k)'A(t_k)'] \\ &= Q^i(t_{k+1}, t_k)A(t_k)'. \end{aligned} \tag{19}$$

The process noises are assumed to be additive, zero mean, and white. It should be noted that to ensure the whiteness of the discrete time process noises over the arbitrary sampling intervals, one has to use the discretized continuous-time white noise state propagation models.

[2]For asynchronous homogeneous sensors (with same LT states), the cross-covariance iteration is given in [1, eq. (9.3.2-5)] based on the "common process noise." This has to be generalized to the heterogeneous states where there is no common process noise but the process noises in the different states models are related.

## III. THE CROSS-COVARIANCE OF THE ESTIMATION ERRORS

### A. Synchronous Case

Consider a tracker at a single sensor (this could be sensor $i$ or sensor $j$) with the state model and measurement model to be

$$\mathbf{x}(k+1) = f[k, \mathbf{x}(k)] + \mathbf{v}(k), \tag{20}$$

$$\mathbf{z}(k+1) = h[k, \mathbf{x}(k)] + \mathbf{w}(k). \tag{21}$$

The updated state at time $k$ is, using an extended Kalman filter (EKF),

$$\hat{\mathbf{x}}(k|k) = f\left[\hat{\mathbf{x}}(k-1|k-1)\right] + W(k)\nu(k). \tag{22}$$

Expanding $h[k, \mathbf{x}(k)]$ around $\hat{\mathbf{x}}(k|k-1)$ yields the innovation

$$\begin{aligned} \nu(k) &= h[k, \mathbf{x}(k)] + \mathbf{w}(k) - h[k, \hat{\mathbf{x}}(k|k-1)] \\ &= h[k, \hat{\mathbf{x}}(k|k-1)] + H(k)[\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1)] \\ &\quad + \mathbf{w}(k) - h[k, \hat{\mathbf{x}}(k|k-1)] \\ &= H(k)[\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1)] + \mathbf{w}(k), \end{aligned} \tag{23}$$

where

$$H(k) = \left[\nabla_x h(k, x)'\right]'|_{x=\hat{\mathbf{x}}(k|k-1))}. \tag{24}$$

Using the dynamic equation (20), (23) can be written as

$$\begin{aligned} \nu(k) = H(k)\big\{&f[\mathbf{x}(k-1)] + \mathbf{v}(k-1) \\ &- f[\hat{\mathbf{x}}(k-1|k-1))]\big\} + \mathbf{w}(k). \end{aligned} \tag{25}$$

Expanding $f[\mathbf{x}(k-1)]$ around $\hat{\mathbf{x}}(k-1|k-1)$ yields

$$\begin{aligned} \nu(k) &= H(k)\big\{f[\hat{\mathbf{x}}(k-1|k-1)] + F(k-1)[\mathbf{x}(k-1) \\ &\quad - \hat{\mathbf{x}}(k-1|k-1)] + \mathbf{v}(k-1) \\ &\quad - f[\hat{\mathbf{x}}(k-1|k-1)]\big\} + \mathbf{w}(k) \\ &= H(k)F(k-1)\tilde{\mathbf{x}}(k-1|k-1) \\ &\quad + H(k)\mathbf{v}(k-1) + \mathbf{w}(k), \end{aligned} \tag{26}$$

where

$$F(k-1) = \left[\nabla_x f(k-1, x)'\right]'|_{x=\hat{\mathbf{x}}(k-1|k-1)}. \tag{27}$$

The estimation error at time $k-1$ is

$$\tilde{\mathbf{x}}(k-1|k-1) = \mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1|k-1). \tag{28}$$

Substituting (26) into (22) yields

$$\begin{aligned} \hat{\mathbf{x}}(k|k) = &f[\hat{\mathbf{x}}(k-1|k-1)] \\ &+W(k)[H(k)F(k-1)\tilde{\mathbf{x}}(k-1|k-1) \\ &+H(k)\mathbf{v}(k-1) + \mathbf{w}(k)]. \end{aligned} \tag{29}$$

The first-order vector Taylor series expansion of (20) around $\hat{\mathbf{x}}(k-1|k-1)$ is

$$\mathbf{x}(k) = f\left[\hat{\mathbf{x}}(k-1|k-1)\right] + F(k-1)\tilde{\mathbf{x}}(k-1|k-1) + \mathbf{v}(k-1). \tag{30}$$

Subtracting (29) from (30), the estimation error at time $k$ can be expressed as

$$\begin{aligned}\tilde{\mathbf{x}}(k|k) &= \mathbf{x}(k) - \hat{\mathbf{x}}(k|k)\\ &= [I_{n_\mathbf{x}} - W(k)H(k)]F(k-1)\tilde{\mathbf{x}}(k-1|k-1)\\ &\quad + [I_{n_\mathbf{x}} - W(k)H(k)]\mathbf{v}(k-1) + W(k)\mathbf{w}(k),\end{aligned} \tag{31}$$

where $I_{n_\mathbf{x}}$ is the $n_\mathbf{x}$-dimensional identity matrix and $n_\mathbf{x}$ is the dimension of the state vector $\mathbf{x}$.

Following the discussion above, the estimation errors from the two sensors $i$ and $j$ are

$$\begin{aligned}\tilde{\mathbf{x}}^i(k|k) &= \mathbf{x}^i(k) - \hat{\mathbf{x}}^i(k|k)\\ &= [I_{n^i_\mathbf{x}} - W^i(k)H^i(k)]F^i(k-1)\tilde{\mathbf{x}}^i(k-1|k-1)\\ &\quad + [I_{n^i_\mathbf{x}} - W^i(k)H^i(k)]\mathbf{v}^i(k-1) + W^i(k)\mathbf{w}^i(k),\end{aligned} \tag{32}$$

$$\begin{aligned}\tilde{\mathbf{x}}^j(k|k) &= \mathbf{x}^j(k) - \hat{\mathbf{x}}^j(k|k)\\ &= h[\mathbf{x}^i(k)] - \hat{\mathbf{x}}^j(k|k)\\ &= [I_{n^j_\mathbf{x}} - W^j(k)H^j(k)]F^j(k-1)\tilde{\mathbf{x}}^j(k-1|k-1)\\ &\quad + [I_{n^j_\mathbf{x}} - W^j(k)H^j(k)]\mathbf{v}^j(k-1) + W^j(k)\mathbf{w}^j(k).\end{aligned} \tag{33}$$

Note that $\mathbf{x}^j(k)$ in (33) is $\alpha[\mathbf{x}^i(k)]$ (5); that is, there is a common truth in (32) and (33).

Then, the estimation errors' cross-covariance (of dimension $n^i_\mathbf{x} \times n^j_\mathbf{x}$) is

$$\begin{aligned}P^{ij}(k|k) &= E[\tilde{\mathbf{x}}^i(k|k)\tilde{\mathbf{x}}^j(k|k)']\\ &= [I - W^i(k)H^i(k)]F^i(k-1)P^{ij}(k-1|k-1)\\ &\quad \times F^j(k-1)'[I - W^j(k)H^j(k)]'\\ &\quad + [I - W^i(k)H^i(k)]Q^{ij}(k-1)[I - W^j(k)H^j(k)]'\\ &= [I - W^i(k)H^i(k)]\{F^i(k-1)P^{ij}(k-1|k-1)F^j(k-1)'\\ &\quad + Q^{ij}(k-1)\}[I - W^j(k)H^j(k)]'.\end{aligned} \tag{34}$$

## B. Asynchronous Case

In this case, the cross-covariance between the estimation errors is, based on [1, eq. (9.3.2-5)] and the previous discussion about the synchronous case,

$$\begin{aligned}P^{ij}(t_k|t_k) &= [I - \chi^i(t_k)W^i(t_k)H^i(t_k)]\\ &\quad \cdot \{F^i(t_k, t_{k-1}, \hat{\mathbf{x}}^i(t_{k-1}|t^i(t_{k-1})))P^{ij}(t_{k-1}|t_{k-1})\\ &\quad \times F^j(t_k, t_{k-1}, \hat{\mathbf{x}}^j(t_{k-1}|t^j(t_{k-1})))'\\ &\quad + Q^{ij}(t_k, t_{k-1})\}[I - \chi^j(t_k)W^j(t_k)H^j(t_k)]',\end{aligned} \tag{35}$$

where $t^i(t_{k-1})$ and $t^j(t_{k-1})$ are the most recent times prior to $t_{k-1}$ at which LT $i$ and LT $j$ sent information to the FC, respectively, and

$$\begin{aligned}&F^i(t_k, t_{k-1}, \hat{\mathbf{x}}^i(t_{k-1}|t^i(t_{k-1})))\\ &\quad = \left[\nabla_x f^i(t_k, t_{k-1}, x)'\right]'|_{x=\hat{\mathbf{x}}^i(t_{k-1}|t^i(t_{k-1})))},\end{aligned} \tag{36}$$

$$\begin{aligned}&F^j(t_k, t_{k-1}, \hat{\mathbf{x}}^j(t_{k-1}|t^j(t_{k-1})))\\ &\quad = \left[\nabla_x f^j(t_k, t_{k-1}, x)'\right]'|_{x=\hat{\mathbf{x}}^j(t_{k-1}|t^j(t_{k-1})))}.\end{aligned} \tag{37}$$

In (35),

$$\chi^i(t_k) = \begin{cases}1, & \text{if sensor } i \text{ has a measurement at time } t_k,\\ 0, & \text{others,}\end{cases} \tag{38}$$

similarly for sensor $j$.

With the following assumptions for the previous fused estimate $\hat{\mathbf{x}}(t_{k-1}|t_{k-1})$:

$$F^i(t_k, t_{k-1}, \hat{\mathbf{x}}^i(t_{k-1}|t^i(t_{k-1}))) \approx F^i(t_k, t_{k-1}, \hat{\mathbf{x}}(t_{k-1}|t_{k-1})), \tag{39}$$

$$F^j(t_k, t_{k-1}, \hat{\mathbf{x}}^j(t_{k-1}|t^j(t_{k-1}))) \approx F^j(t_k, t_{k-1}, \hat{\mathbf{x}}(t_{k-1}|t_{k-1})), \tag{40}$$

equation (35) becomes

$$\begin{aligned}P^{ij}(t_k|t_k) &= [I - \chi^i(t_k)W^i(t_k)H^i(t_k)]\\ &\quad \cdot \{F^i(t_k, t_{k-1}, \hat{\mathbf{x}}(t_{k-1}|t_{k-1}))P^{ij}(t_{k-1}|t_{k-1})\\ &\quad \times F^j(t_k, t_{k-1}, \hat{\mathbf{x}}(t_{k-1}|t_{k-1}))'\\ &\quad + Q^{ij}(t_k, t_{k-1})\}[I - \chi^j(t_k)W^j(t_k)H^j(t_k)]'.\end{aligned} \tag{41}$$

If the fusion is at a time when there is an updated state only from $\text{LT}_i$, it will use a prediction to that time from $\text{LT}_j$. Note that although the fusion can be carried out on demand, the cross-covariance calculation needs to run with full rate.

# IV. THE STATE MODELS FOR THE ACTIVE AND PASSIVE SENSORS

In the $\xi$–$\eta$ space, an active sensor located at $[\xi^\text{a} \;\; \eta^\text{a}]$ with range and azimuth angle measurements (without time arguments, for simplicity)

$$r = \sqrt{(\xi - \xi^\text{a})^2 + (\eta - \eta^\text{a})^2} + w^\text{r}, \tag{42}$$

$$\theta^\text{a} = \tan^{-1}[(\eta - \eta^\text{a})/(\xi - \xi^\text{a})] + w^\text{a} \tag{43}$$

and a passive sensor located at $[\xi^\text{p} \;\; \eta^\text{p}]$ with bearing measurements only

$$\theta^\text{p} = \tan^{-1}[(\eta - \eta^\text{p})/(\xi - \xi^\text{p})] + w^\text{p} \tag{44}$$

are considered for the T2TF for a 2D target. The measurement noises $w^\text{r}$, $w^\text{a}$, and $w^\text{p}$ are assumed to be independent zero-mean white Gaussian with corresponding standard deviations $\sigma^\text{r}$, $\sigma^\text{a}$, and $\sigma^\text{p}$.

The active sensor's measurements in polar coordinates are transformed into Cartesian coordinates with an unbiased transformation [5, Sec. 10.4.3]. Given (42) and (43), the unbiased transformed measurement vector is

$$\mathbf{z}^{\mathrm{C}} = \begin{bmatrix} \xi^{\mathrm{C}} \\ \eta^{\mathrm{C}} \end{bmatrix} = \begin{bmatrix} b_1^{-1} r\cos(\theta^{\mathrm{a}}) + \xi^{\mathrm{a}} \\ b_1^{-1} r\sin(\theta^{\mathrm{a}}) + \eta^{\mathrm{a}} \end{bmatrix}, \tag{45}$$

where

$$b_1 = \mathrm{e}^{-\sigma^{\mathrm{a}}/2}. \tag{46}$$

The transformed active sensor noise vector $\mathbf{w}_{\mathrm{C}}$ has the covariance matrix $\mathbf{R}_{\mathrm{C}}$ with elements

$$\mathbf{R}_{\mathrm{C}}(1,1) = b_1^{-2} r^2 \cos(\theta^{\mathrm{a}}) + 0.5(r^2 + \sigma^{\mathrm{a}})(1 + b_1^4 \cos(\theta^{\mathrm{a}})), \tag{47}$$

$$\mathbf{R}_{\mathrm{C}}(2,2) = b_1^{-2} r^2 \sin(\theta^{\mathrm{a}}) + 0.5(r^2 + \sigma^{\mathrm{a}})(1 - b_1^4 \cos(\theta^{\mathrm{a}})), \tag{48}$$

$$\begin{aligned} \mathbf{R}_{\mathrm{C}}(1,2) &= \mathbf{R}_{\mathrm{C}}(2,1) \\ &= (0.5 b_1^{-2} r^2 + 0.5(r^2 + \sigma^{\mathrm{a}}) b_1^4 - r^2)\sin(2\theta^{\mathrm{a}}). \end{aligned} \tag{49}$$

For the active sensor, a nearly coordinated turn model [7] is used for tracking along with an EKF. The 5D state vector[3] includes position, velocity, and turn rate $\Omega$, that is,

$$\mathbf{x}^{\mathrm{a}} = [\xi \;\; \dot{\xi} \;\; \eta \;\; \dot{\eta} \;\; \Omega] \tag{50}$$

with the discretized dynamic model to be

$$\mathbf{x}^{\mathrm{a}}(t_{l+1}^{\mathrm{a}}) = f^{\mathrm{a}}[\mathbf{x}^{\mathrm{a}}(t_l^{\mathrm{a}})] + \mathbf{v}^{\mathrm{a}}[\mathbf{x}^{\mathrm{a}}(t_l^{\mathrm{a}})], \tag{51}$$

$$\mathbf{z}^{\mathrm{C}} = H^{\mathrm{a}} \mathbf{x}^{\mathrm{a}}(t_l^{\mathrm{a}}) + \mathbf{w}^{\mathrm{C}}(t_l^{\mathrm{a}}), \tag{52}$$

where

$$\begin{aligned} &f^{\mathrm{a}}[\mathbf{x}^{\mathrm{a}}(t_l^{\mathrm{a}})] \\ &= \begin{bmatrix} \xi(t_l^{\mathrm{a}}) + T^{\mathrm{a}} \dot{\xi}(t_l^{\mathrm{a}}) - (T^{\mathrm{a}})^2 \Omega(t_l^{\mathrm{a}}) \dot{\eta}(t_l^{\mathrm{a}})/2 \\ \dot{\xi}(t_l^{\mathrm{a}}) - T^{\mathrm{a}} \Omega(t_l^{\mathrm{a}}) \dot{\eta}(t_l^{\mathrm{a}}) - (T^{\mathrm{a}})^2 \Omega(t_l^{\mathrm{a}})^2 \dot{\xi}(t_l^{\mathrm{a}})/2 \\ \eta(t_l^{\mathrm{a}}) + T^{\mathrm{a}} \dot{\eta}(t_l^{\mathrm{a}}) + (T^{\mathrm{a}})^2 \Omega(t_l^{\mathrm{a}}) \dot{\xi}(t_l^{\mathrm{a}})/2 \\ \dot{\eta}(t_l^{\mathrm{a}}) + T^{\mathrm{a}} \Omega(t_l^{\mathrm{a}}) \dot{\xi}(t_l^{\mathrm{a}}) - (T^{\mathrm{a}})^2 \Omega(t_l^{\mathrm{a}})^2 \dot{\eta}(t_l^{\mathrm{a}})/2 \\ \Omega(t_l^{\mathrm{a}}) \end{bmatrix}, \end{aligned} \tag{53}$$

$$H^{\mathrm{a}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{54}$$

The process noise vector is target state dependent and its covariance matrix is discussed in [3]. The continuous-time process noise "intensities" are the power spectral densities that need to be chosen in the design of the process noise covariance matrix.

[3]Here, the superscripts $i$ and $j$ used in the previous text are replaced by "a" and "p" to indicate the states are from one active sensor and one passive sensor, respectively.

The passive sensor uses a Kalman filter based on a continuous-time Wiener process acceleration model with a state vector involving the angle, angle rate, and angle acceleration:

$$\mathbf{x}^{\mathrm{p}} = [\theta \;\; \dot{\theta} \;\; \ddot{\theta}]. \tag{55}$$

The discretized dynamic model is

$$\mathbf{x}^{\mathrm{p}}(t_m^{\mathrm{p}} + 1) = F^{\mathrm{p}} \mathbf{x}^{\mathrm{p}}(t_m^{\mathrm{p}}) + \mathbf{v}^{\mathrm{p}}(t_m^{\mathrm{p}}), \tag{56}$$

$$\mathbf{z}^{\mathrm{p}} = H^{\mathrm{p}} \mathbf{x}^{\mathrm{p}}(t_m^{\mathrm{p}}) + \mathbf{w}^{\mathrm{p}}(t_m^{\mathrm{p}}), \tag{57}$$

where

$$F^{\mathrm{p}} = \begin{bmatrix} 1 & T^{\mathrm{p}} & (T^{\mathrm{p}})^2/2 \\ 0 & 1 & T^{\mathrm{p}} \\ 0 & 0 & 1 \end{bmatrix}, \tag{58}$$

$$H^{\mathrm{p}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{59}$$

The process noise covariance matrix of the passive tracker's model at time $k$ has the relationship with active process noise covariance matrix shown in (10). The state vector (50) and the state vector (55) have a nonlinear relationship

$$\mathbf{x}^{\mathrm{p}} = \alpha\,[\mathbf{x}^{\mathrm{a}}] \tag{60}$$

with explicit expressions

$$\theta = \operatorname{atan}\left(\frac{\eta - \eta^{\mathrm{p}}}{\xi - \xi^{\mathrm{p}}}\right), \tag{61}$$

$$\dot{\theta} = \frac{v\sin(\phi)}{r^{\mathrm{p}}}, \tag{62}$$

$$\ddot{\theta} = \frac{v\cos(\phi)\Omega}{r^{\mathrm{p}}}, \tag{63}$$

where $v$ is the target speed

$$v = \sqrt{\dot{\xi}^2 + \dot{\eta}^2}, \tag{64}$$

$r^{\mathrm{p}}$ is the range with respect to the passive sensor's location

$$r^{\mathrm{p}} = \sqrt{(\xi - \xi^{\mathrm{p}})^2 + (\eta - \eta^{\mathrm{p}})^2}, \tag{65}$$

and $\phi$ is the difference between velocity angle and position azimuth angle:

$$\phi = \operatorname{atan}\left(\frac{\dot{\eta}}{\dot{\xi}}\right) - \operatorname{atan}\left(\frac{\eta - \eta^{\mathrm{p}}}{\xi - \xi^{\mathrm{p}}}\right). \tag{66}$$

## V. THE LMMSE ESTIMATOR FOR HETEROGENEOUS T2TF

### A. Synchronous Case

The LMMSE estimation for heterogeneous T2TF is carried out (omitting the time arguments) with

$$\hat{\mathbf{x}}^i = \hat{\mathbf{x}}^{\mathrm{a}}(1:4) = [\hat{\xi} \;\; \hat{\dot{\xi}} \;\; \hat{\eta} \;\; \hat{\dot{\eta}}]' \tag{67}$$

as the active sensor's track and

$$\hat{\mathbf{x}}^j = \hat{\mathbf{x}}^{\mathrm{p}}(1:2) = [\hat{\theta} \;\; \hat{\theta}]' \tag{68}$$

as the passive sensor's track.

The fused track estimate is obtained by (derivation can be found in [1, Sec. 9.2.3])

$$\hat{\mathbf{x}}^i_{\mathrm{LMMSE}} = \hat{\mathbf{x}}^i + P_{\mathbf{xz}} P_{\mathbf{zz}}^{-1} [\hat{\mathbf{x}}^j - g[\hat{\mathbf{x}}^i]], \tag{69}$$

where $g[\cdot]$ is the nonlinear relationship between the states from the two sensors. Here, we use $g$ rather than $\alpha$ to avoid ambiguity in calculating the different Jacobians. The corresponding fused covariance matrix is

$$P^i_{\mathrm{LMMSE}} = P^i - P_{\mathbf{xz}} P_{\mathbf{zz}}^{-1} P'_{\mathbf{xz}}, \tag{70}$$

where

$$P_{\mathbf{xz}} \approx P^i (G^i)' - P^{ij}, \tag{71}$$

$$P_{\mathbf{zz}} \approx P^i - G^i P^{ij} - P^{ji} (G^i)' + G^i P^i (G^i)', \tag{72}$$

and

$$G^i = [\nabla^i_{\mathbf{x}} g(\mathbf{x}^j)']'|_{\mathbf{x}^i = \hat{\mathbf{x}}^i} \tag{73}$$

is the Jacobian evaluated at the estimate from sensor $i$.

## B. Asynchronous Case

In the asynchronous case, the fusion is carried out at the times given by the union of the different sampling times of the two sensors. Since not all the LTs' communications are available to the FC at the fusion time, predictions of the LTs' latest estimates (prior or at the fusion time) are used for LMMSE estimation. The fused track estimate (extension of (69)) at time $t_k$ is obtained by

$$\begin{aligned}\hat{\mathbf{x}}_{\mathrm{LMMSE}}(t_k) = {} & f^i\left[t_k, t^i(t_k), \hat{\mathbf{x}}^i[t^i(t_k)|t^i(t_k)]\right] \\ & + P_{\mathbf{xz}} P_{\mathbf{xz}}^{-1} \Big\{ f^j\left[t_k, t^j(t_k), \hat{\mathbf{x}}^j[t^j(t_k)|t^j(t_k)]\right] \\ & - g\Big[ f^i\left[t_k, t^i(t_k), \hat{\mathbf{x}}^i[t^i(t_k)|t^i(t_k)]\right] \Big] \Big\}, \end{aligned} \tag{74}$$

where $t^i(t_k)$ and $t^j(t_k)$ are the latest times up to and including $t_k$ at which LT $i$ and LT $j$ sent information to the FC. In (74), the latest estimates (or the predication if needed) are used. The corresponding fused covariance matrix calculation is carried out (based on (70)) in terms of the LTs' latest covariance matrices

$$P^i_{\mathrm{LMMSE}}(t_k) = P^i(t_k|t^i(t_k)) - P_{\mathbf{xz}} P_{\mathbf{zz}}^{-1} P'_{\mathbf{xz}}, \tag{75}$$

where

$$P_{\mathbf{xz}} \approx P^i(t_k|t^i(t_k)) G^i(t_k|t^i(t_k))' - P^{ij}(t_k|t_k), \tag{76}$$

$$\begin{aligned} P_{\mathbf{zz}} \approx {} & P^j(t_k|t^i(t_k)) - G^i(t_k|t^i(t_k)) P^{ij}(t_k|t_k) \\ & - P^{ji}(t_k|t_k) G^i(t_k|t^i(t_k))' \\ & + G^i(t_k|t^i(t_k)) P^i(t_k|t^i(t_k)) G^i(t_k|t^i(t_k))', \end{aligned} \tag{77}$$

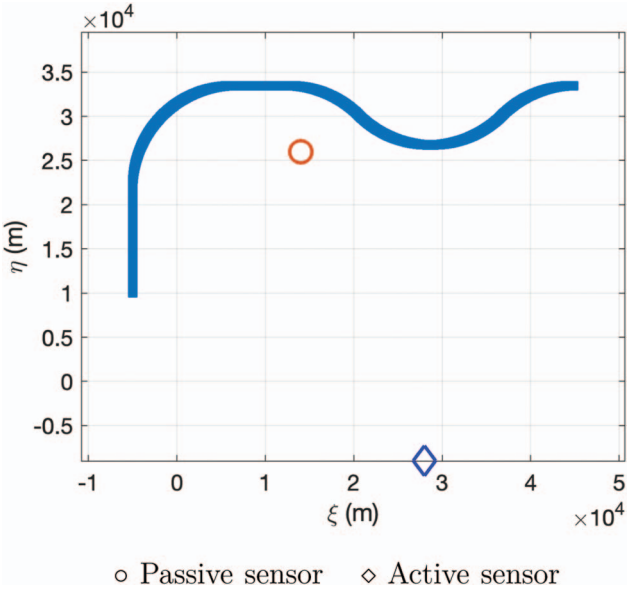



Fig. 2. Target trajectory on the $\xi$–$\eta$ plane.

and

$$G^i(t_k|t^i(t_k)) = [\nabla^i_{\mathbf{x}} g(\mathbf{x}^j)']'|_{\mathbf{x}^i = f^i\left[t_k, t^i(t_k), \hat{\mathbf{x}}^i[t^i(t_k)|t^i(t_k)]\right]} \tag{78}$$

is the Jacobian evaluated at the prediction/estimate from LT $i$. The covariance matrix from LT $i$ is

$$\begin{aligned} & P^i(t_k|t^i(t_k)) \\ & = \begin{cases} P^i(t^i(t_k)|t^i(t_k)), & \text{if } t_k = t^i(t_k), \\ F^i\left[t_k, t^i(t_k), \hat{\mathbf{x}}^i[t^i(t_k)|t^i(t_k)]\right] P^i(t^i(t_k)|t^i(t_k)) & \\ \quad \cdot F^i\left[t_k, t^i(t_k), \hat{\mathbf{x}}^i[t^i(t_k)|t^i(t_k)]\right]' + Q^i(t_k, t^i(t_k)), & \text{others.} \end{cases} \end{aligned} \tag{79}$$

# VI. SIMULATION RESULTS

In the simulation, a target moving in a plane is considered with initial position $[-5 \;\; 10]$ km and the initial velocity $[0 \quad 200]$ m/s in Cartesian coordinates. The tar-

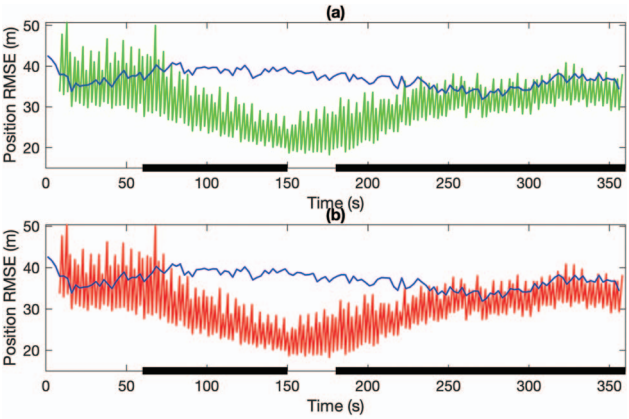



Fig. 3. Position RMSEs from 500 MC runs.

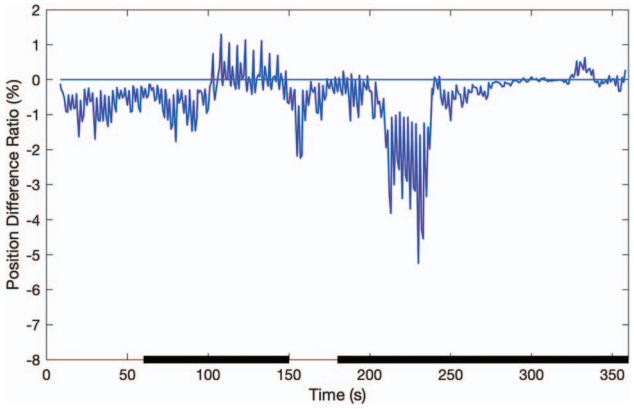



Fig. 4. Position RMSE difference ratio of the fused estimate (with cross-covariance minus without cross-covariance divided by without cross-covariance).

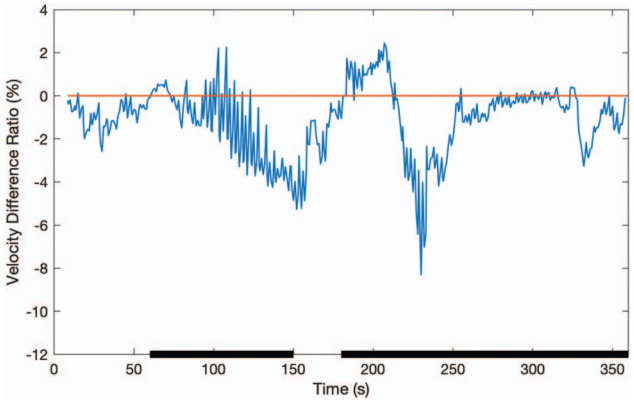



Fig. 6. Velocity RMSE difference ratio of the fused estimate (with cross-covariance minus without cross-covariance divided by without cross-covariance).

get keeps going straight for 60 s and makes a right turn at 1 rad/s for 90 s. Then, it keeps straight for another 30 s and turns to the right with turn rate 1 rad/s lasting for 45 s, followed by a left turn at 1 rad/s for 90 s, and finally makes a right turn at 1 rad/s for 45 s. The trajectory of the target is shown in Fig. 2.

The active sensor is located at [28 −9] km with measurement noise standard deviations $\sigma^{\mathrm{r}} = 20$ m and $\sigma^{\mathrm{a}} = 1$ mrad. The passive sensor is located at [14 26] km with angle measurement noise $\sigma^{\mathrm{p}} = 1$ mrad. The sampling intervals are $T^{\mathrm{a}} = 2.5$ s (for the active sensor) and $T^{\mathrm{p}} = 1$ s (for the passive sensor). Five hundred MC runs are made in the simulation to obtain the results. Maneuver duration is highlighted on the time axis in the following figures.

Fig. 3 shows the root mean square errors (RMSEs) of the position vector from the active sensor, for both fusion without cross-covariance and fusion with cross-covariance. The difference ratio between the latter two is compared (RMSE of fusion with cross-covariance minus RMSE of fusion without cross-covariance divided by the latter one) in Fig. 4. Similarly, the velocity vector RMSEs are shown in Fig. 5 with the difference comparison shown in Fig. 6. The negative differences shown in Figs. 3 and 5 indicate better performance of the fusion with cross-covariance as it achieves smaller RMSE. For position, the fusion with cross-covariance has RMSE reduction up to 6% (MSE reduction 12%); for velocity, the fusion with cross-covariance has RMSE reduction up to 8% (MSE reduction 16%). The difference ratio depends on the maneuvers but is not only limited to that since the maneuvers are not obvious to the passive sensor. The performance is sensitive to the geometry of the target's trajectory and the sensor positions. The CTF using the original measurements sequentially from different sensors is compared with the proposed fusion approach with simulation results shown in Figs. 7 and 8 for position and velocity, respectively. As shown in [3], the CTF using interacting multiple model (IMM) cannot "see" the maneuvers at the times when there is only a passive sensor measurement—the CTF IMM performs worse than the heterogeneous T2TF. For the case considered in this work, the FC used one EKF only. The CTF is sometimes worse (e.g., when the target starts maneuvering after 60 s) since the passive measurements (angle only) used in CTF-EKF cannot

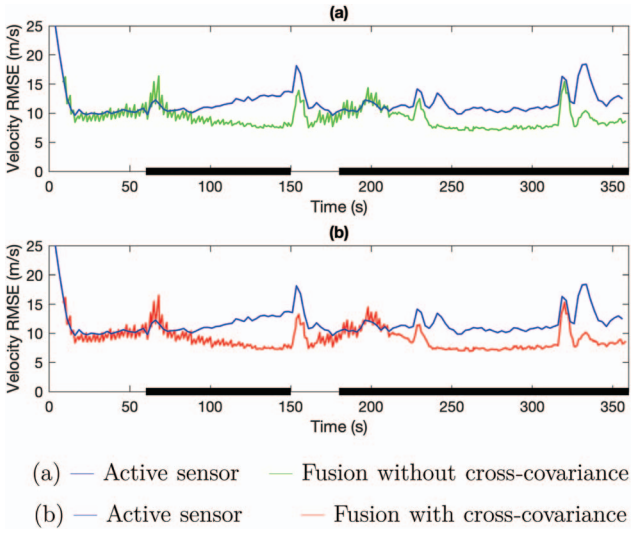



Fig. 5. Velocity RMSEs from 500 MC runs.

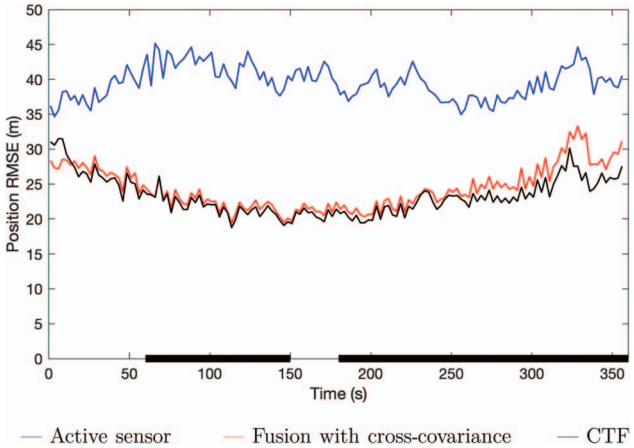



Fig. 7. Position RMSEs of the fused estimate.

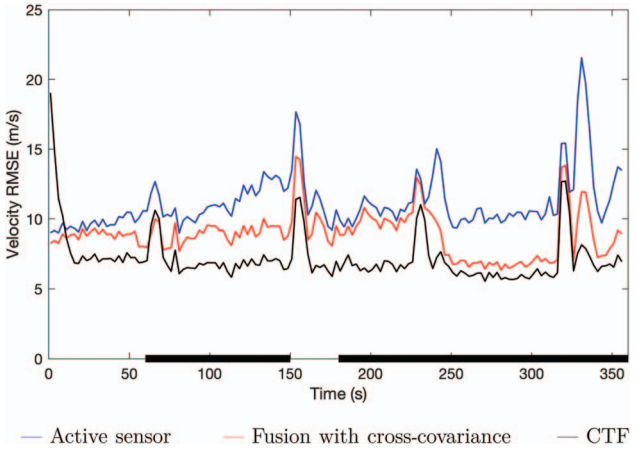



Fig. 8. Velocity RMSEs of the fused estimate.

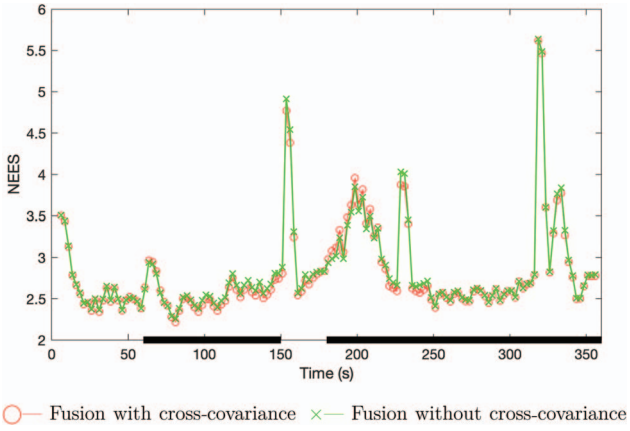
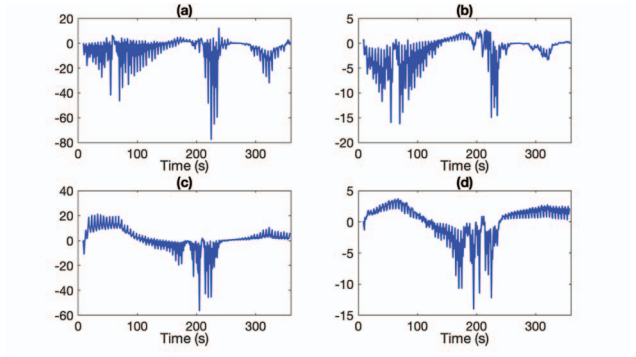



Fig. 10. NEES from 500 runs.

provide sufficient information on the target's velocity and maneuvers. The IMM is not used at the FC since the cross-covariance needs to be weighted based on the model probabilities and the relationship of different state models in Cartesian space and in angle space. In this work, this information is not available at the FC.

For the asynchronous, heterogeneous, and nonlinear case considered, the fusion with cross-covariance yields the actual variance, which is sometimes larger and sometimes smaller than the variance obtained under the (inaccurate) assumption of independence between the estimation errors. The variance differences (variance of fusion with cross-covariance minus variance of fusion without cross-covariance) for each component in the Cartesian state vector are shown in Fig. 9. Neglecting the cross-covariance between the estimation errors makes the fusion sometimes optimistic and sometimes pessimistic. The normalized estimation error squared (NEES) with the maneuver duration highlighted is shown in Fig. 10. Due to the facts that (i) the target is maneuvering (i.e., its motion uncertainty is deterministic rather than a stochastic white noise process) and (ii) the local trackers are running asynchronously, the system is not expected to be consistent with the ideal NEES of 4. For the white noise-driven motion model as in [6], the fusion with cross-covariance was shown to be consistent.



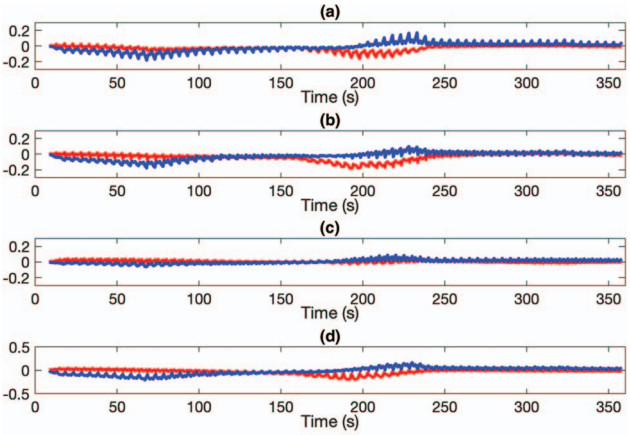

(a) Variance difference of $\xi$ $(\mathrm{m})^2$ (b) Variance difference of $\dot{\xi}$ $(\mathrm{m/s})^2$
(c) Variance difference of $\eta$ $(\mathrm{m})^2$ (d) Variance difference of $\dot{\eta}$ $(\mathrm{m/s})^2$

Fig. 9. Variance difference of elements from the state vector in Cartesian coordinates.

The cross-correlation coefficients between the elements in (67) and (68) are shown in Fig. 11. The cross-correlation coefficients depend on the geometry of the two sensors and the target as well as the maneuvers of the target. It can be seen from Fig. 9 that some of the cross-correlation coefficients are positive and some of them are negative. This confirms the results in [3], which were obtained numerically through an MC investigation.

## VII. SUMMARY AND CONCLUSIONS

In this work, we derived the cross-covariance (for both the synchronous and asynchronous cases) between the local estimation errors of heterogeneous tracks from local sensors with different state models. The simulation




(a) — $\hat{\xi}$ and $\hat{\theta}$ — $\hat{\eta}$ and $\hat{\theta}$
(b) — $\hat{\xi}$ and $\dot{\hat{\theta}}$ — $\hat{\eta}$ and $\dot{\hat{\theta}}$
(c) — $\dot{\hat{\xi}}$ and $\hat{\theta}$ — $\dot{\hat{\eta}}$ and $\hat{\theta}$
(d) — $\dot{\hat{\xi}}$ and $\dot{\hat{\theta}}$ — $\dot{\hat{\eta}}$ and $\dot{\hat{\theta}}$

Fig. 11. Cross-correlation coefficients between the elements from two state vectors.

results from a scenario with one passive tracker and one active tracker show the performance of heterogeneous T2TF with cross-covariance. It can be seen that with the cross-covariance, the T2TF can achieve improved performance with lower RMSE and better statistical efficiency. The cross-correlation coefficients are sometimes positive and sometimes negative, which confirms the results obtain in [3] through an MC investigation.

REFERENCES


[1] Y. Bar-Shalom, P. Willett, and X. Tian
*Tracking and Data Fusion*. Storrs, CT: YBS Publishing, 2011.

[2] X. Tian and Y. Bar-Shalom,
"On Algorithms for Asynchronous Track-to-Track Fusion,"
in *Proc. Int. Conf. Inf. Fusion*, Edinburgh, UK, July 2010.

[3] T. Yuan, Y. Bar-Shalom, and X. Tian
"Heterogeneous Track-to-Track Fusion,"
*J. Adv. Inf. Fusion*, vol. 6, no. 2, pp. 131–149, Dec. 2011.

[4] M. Mallick, K. C. Chang, S. Arulampalam, and Y. Yan,
"Heterogeneous Track-to-Track Fusion in 3D Using IRST Sensor and Air MTI Radar," *IEEE Trans. on Aerospace Engineering Systems*, vol. 55, no. 6, pp. 3062–3079, Dec. 2019

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

[6] K. Yang, Y. Bar-Shalom, and P. Willett,
"Track-to-Track Fusion with Crosscovariances from Radar and IR/EO Sensor," in *Proc. Int. Conf. Inf. Fusion*, Ottawa, Canada, July 2019.

[7] M. R. Morelande and N. J. Gordon
"Target Tracking Through a Coordinated Turn," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. (ICASSP '05)*, 2005, vol. 4, pp. iv/21–iv/24.



**Kaipei Yang** received the B.S. degree from the Northwestern Polytechnical University, Xi'an, China, in 2014, and the Ph.D. degree from the University of Connecticut, Storrs, CT, USA, in 2019. She is currently an Assistant Research Professor with the Department of Electrical and Computer Engineering, University of Connecticut. Her research interests include statistical signal processing, estimation theory, and information fusion. She gained experience in autonomous driving vehicles while working at NIO, San Jose, CA, USA, in 2018.

**Yaakov Bar-Shalom** received the B.S. and M.S. degrees in electrical engineering from the Technion, Haifa, Israel, in 1963 and 1967, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 1970. From 1970 to 1976, he was with Systems Control, Inc., Palo Alto, CA, USA. He is currently a Board of Trustees Distinguished Professor with the Department of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering with the University of Connecticut, Storrs, CT, USA. His current research interests include estimation theory, target tracking, and data fusion. He has authored or coauthored more than 550 papers and book chapters, and coauthored/edited 8 books, including *Tracking and Data Fusion* (YBS Publishing, 2011). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He was an Associate Editor for the *IEEE Transactions on Automatic Control and Automatica*. He was General Chairman of the 1985 ACC, Chairman of the Conference Activities Board of the IEEE CSS and member of its Board of Governors, General Chairman of FUSION 2000, President of ISIF in 2000 and 2002, and Vice President for Publications during 2004–2013. In 1987, he was the recipient of the IEEE CSS Distinguished Member Award. Since 1995, he is a Distinguished Lecturer of the IEEE Aerospace and Electronic Systems Society. He was the corecipient of the M. Barry Carlton Award for the best paper in the *IEEE Transactions on Aerospace and Electronic Systems* in 1995 and 2000. In 2002, he was the recipient of the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008, he was the recipient of the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012 the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in aerospace engineering based on the citations of his work. He was the recipient of the 2015 ISIF Award for a Lifetime of Excellence in Information Fusion. This award has been renamed in 2016 as the Yaakov Bar-Shalom Award for a Lifetime of Excellence in Information Fusion.

**Peter Willett** received the B.A.Sc. (engineering science) degree from the University of Toronto, Toronto, Canada, in 1982, and the Ph.D. degree from Princeton University, Princeton, NJ, USA, in 1986. He has been a faculty member with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA, since 1986. Since 1998, he has been a Professor, and since 2003 an IEEE Fellow. His primary areas of research include statistical signal processing, detection, machine learning, communications, data fusion, and tracking. He has authored or coauthored more than 650 papers on these topics. He was Editor-in-Chief for the *IEEE Signal Processing Letters* from 2014 to 2016. He was Editor-in-Chief for the *IEEE Transactions on Aerospace and Electronic Systems* from 2006 to 2011, and then AESS Vice President for Publications (2012–2014). He was a member of the IEEE AESS Board of Governors (2005–2010, 2011–2016) and of the IEEE Signal Processing Society's Sensor-Array and Multichannel (SAM) technical committee (and Chair 2015–2016).

# The Role of Information Fusion in Transfer Learning of Obscure Human Activities During Night

**ANWAAR ULHAQ**

**Human actions are often tightly coupled with their context that can play an important role in their modeling and understating. However, adverse lighting conditions and clutter can easily disrupt the visual context during night, especially in outdoor environments. This situation makes it difficult for any autonomous system to detect or classify actions. Various works have proposed contextual enhancement of available imagery to improve performance. However, no study articulates the most suitable type of contextual enhancement. In this study, we try to evaluate the role of information fusion in enhancing the visual context. We are interested in knowing whether fusion can enhance the performance of the autonomous system or it is just visually appealing. Our evaluation framework is based on transfer learning using deep convolutional neural networks. Experimental results show that contextual enhancement based on 1) the fused contextual information and 2) its colorization significantly enhances the performance of automated action recognition.**

Manuscript received November 20, 2019; revised February 14, 2020; released for publication June 30, 2020.

The author is with the School of Computing and Mathematics, Charles Sturt University, Port Macquarie, NSW 2444, Australia (E-mail: aulhaq@csu.edu.au).

1557-6418/20/$17.00 

## I. INTRODUCTION

Human action recognition is a challenging computer vision problem. Different challenging scenarios are considered in the literature like action by large groups [1], group actions [2], recognizing actions in crowd [3], actions inside movies [4], single- and two-person action recognition [5], action recognition from the side of a video [6], actions across different viewpoints [7]–[9], and occluded actions [10]. However, these approaches assume that the action dataset is captured at daytime under clear context and reasonable lighting conditions. Their performance will decline if available data are adversely affected by diverse lighting conditions and cluttered context. Fig. 1 illustrates two such scenarios of adverse lighting conditions at night. Targets (actors) are visible with dim and hazy context in infrared (IR) imagery. In contrast, background context is clear with hidden or vague targets in visible (VIS) imagery. In this paper, we want to explore if context is enhanced, how it contributes to the automated recognition through machine vision.

Visual context is valuable a priori knowledge in terms of modeling action instances. Therefore, contextual action recognition is addressed by various researchers. The context of scenes is utilized for recognizing events by Li and Fei-Fei [11]; however, it uses only the static images. Contextual action recognition is presented by Marszalek et al. [12], which is based on the bag-of-features framework. It considered the annotated actions in movies and with script mining for visual learning. A similar technique [13] extracts the overall object-based context by detectors and their descriptors with supervised learning. Modeling of scene and object context is designed by Jiang et al. [14] for the Hollywood2 action dataset. These approaches aim at action recognition in high-resolution videos. Hierarchical attention and context modeling for group activity recognition is considered in recent works [15], [16]. However, achieving the same objectives in night-time imagery is cumbersome due to clutter and low-lighting conditions.

Human action activity recognition in a single spectrum is discussed in [17] and [18], which perform recognition in IR spectrum. However, these approaches ignore action contexts that are poorly captured by IR sensors. These approaches, therefore, cannot be classified as contextual action recognition approaches. In this paper, we build upon the idea of [19]–[21] and further evaluate the role of contextual information fusion in recognizing human actions.

Moreover, night-time imagery lacks color information that provides great help to human visual perception. Due to unnatural appearance and IR imagery limitations, multi-sensor systems and color information are integrated for better contextual awareness [22]. Another approach to optimize these systems is to introduce pseudo-color information [23], [24]. A recent study about the perceptual evaluation [25] of such

Fig. 1. Two different scenarios of visual context of actions captured by two different sensors: low-light VIS and IR sensors. Each sensor has its limitation that affects recognition performance. It means contextual improvement can play a positive role in improvement of detection capability of autonomous systems.

color-transformed multispectral systems concludes that pseudo-colorization better illustrates the gist of a night scene by improving the fixation behavior of human eye compared to large-scale imagery.

We address the following research question: Can accuracy of automated action recognition be increased by context enhancement through information fusion and transferring knowledge from daytime image data to night-time data?

This paper claims the following contributions: 1) It evaluates the role of information fusion in transfer learning of activity recognition at night-time. To the best of our knowledge, it is the first work that evaluates such a problem. 2) It explores how transfer learning can be better utilized (frozen or fine-tuned) for transferring knowledge from different domains.

The paper is organized as follows: Section 2 presents the related work, Section 3 illustrates how the context enhancement of multisensor videos is possible, and Section 4 discusses the transfer learning framework and the action filter. Experimental results are discussed in Section 5. The conclusion and references are provided at the end.

## II. PRIOR WORK

Human action recognition is now a well-researched area. There are various methodologies that can be categorized on the basis of the scenario used. The performances of these approaches vary in different circumstances and challenges. One of such challenges is the action context. An action-scene context is acquired through movie-script mining by Liu et al. [1] for realistic action recognition in movies. Spatiotemporal action context was utilized by Han et al. [13] based on space–time features. Similarly, [26] employs convolutional neural networks (CNNs) for contextual action recognition. However, these approaches use high-resolution action datasets for which the extraction of spatiotemporal interest points is straightforward.

Recently, deep CNNs [27] have achieved significant success in object detection and classification. In particular, CNNs trained on the large datasets such as ImageNet have been shown to learn general-purpose image descriptors for a number of vision tasks. A recent trend has been observed about the use of deep feature learning. Various pretrained convolutional network (ConvNet) models are publicly available. In the same spirit, 3D ConvNets [6], [28] were proposed for different types of video analysis tasks, especially action recognition. Instead of using fully connected layers, activations from convolutional layers of the network have achieved superior results.

However, recognition of human actions in low-quality night-time videos is not well-explored area of research and very few approaches can be cited in this category. The utility of thermal imagery is analyzed by Li and Gong [29] for human action recognition. This approach is built upon the histogram of oriented gradients and nearest-neighbor classification.

A similar work [30] uses gait energy images. However, it is limited to walking activity, which is easier to recognize. IR image super-resolution is proposed for enhancement by Du et al. [31]. Deep VIS and thermal image fusion [32] was used for enhanced pedestrian visibility. However, such work cannot be categorized as action recognition work. Fourier transform is a great tool to analyze response of patterns of interest in the frequency domain. Such matching is efficient and faster than matching based on spatial templates. In addition, it combines target classification and detection (localization) simultaneously. Inspired by this this idea, a contextual action recognition approach based on 3D fast Fourier transform and contextual cues was proposed in [19] and [20].

In this paper, we present robust action recognition, which can deal with low-quality night-time video sequences. In case of night-time videos, we consider the registered videos collected from low-light VIS and IR spectra. We enhance the context through video fusion [33]. Our action recognition approach is based on space–time interest point detection and frequency-domain correlation analysis and can detect and classify human actions in a robust manner.

## III. CONTEXT ENHANCEMENT OF NIGHT-TIME VIDEOS

In this section, we discuss the motivation behind contextual enhancement of night-time video sequences, video fusion, and colorization for context enhancement.

### A. Motivation

The aim of context enhancement is a preprocessing step to give day-like appearance to night-time videos.

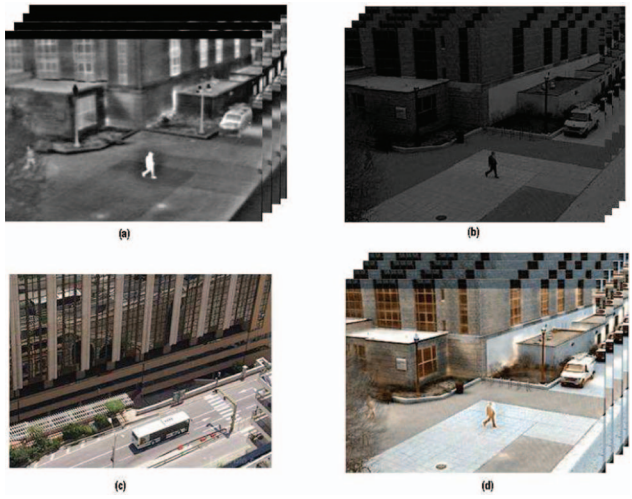


Fig. 2. Color-transfer-based video fusion method: (a) an IR video sequence; (b) low-light VIS domain video sequence; (c) a source color image for the purpose of color transfer; and (d) a color-fused video generated from (a), (b), and (c). Image adapted from [22].

It involves video fusion applied on registered video streams collected from IR and VIS spectra. Context enhancement helps to reveal a camouflaged target and to assist target localization [34]. Here, we present and discuss context enhancement briefly.

### B. Context Enhancement Through Video Fusion and Colorization

The objective of employing video fusion is to generate a single enhanced video from complementary videos that is more suitable for the purpose of human visual perception, action, and context recognition. If we denote $A$ as IR video sequence and $B$ as a VIS video sequence, we intend to generate another video sequence $C$ by fusing visual information from $A$ and $B$. Fig. 2 gives illustrations of video fusion results.

There is extensive literature on contextual enhancement of nigh-time imagery. However, we selected automatic color-transfer-based video fusion (FACE) [22] as it enhances video context by color transfer from a source image because it enhances context through fusion and colorization simultaneously. An example scenario adapted from this work can be seen in Fig. 2.

## IV. MULTIDOMAIN ACTION RECOGNITION VIA TRANSFER LEARNING

Transfer learning is a machine learning methodology where a model developed for a task in one domain is reused as the starting baseline for learning a more specific model on another task in the other domain. Let us define domain and task for better understanding of transfer learning.

Let $D$ denote the domain; we can define it as a two-element tuple (a finite ordered list) consisting of feature space, $X$, and marginal probability, $P(X)$, where $X$ is a sample data point. Therefore, we can write the domain mathematically as $D = (X, P(X))$.

A task, $T$, then can be defined as a two-element tuple of the label space, $Y$, and objective function, $O$, denoted as $P(Y|X)$ from a probabilistic viewpoint. Thus, a task can be defined as $T = (Y, P(Y|X)) = (Y, O)$. We can define transfer learning as follows.

Given a source domain $D_s$ with source task $T_s$, and similarly, a target domain $D_t$ with target task $T_t$, transfer learning has the objective to learn the target conditional probability distribution $P(Y_T|X_T)$ in $D_T$ with the knowledge transferred from $D_s$ and $T_s$, where $D_s \neq D_t$ and $T_s \neq T_t$. Usually in such scenarios, the number of labeled target examples is exponentially smaller than the number of labeled source examples. We have a similar scenario as the majority of action datasets and trained models have daytime-captured data, while target night-vision data are scarce. Our problem in this study, however, is not the design of effective transfer learning but to evaluate the suitability of data for transfer learning.

We will use transfer learning to extract knowledge from the already trained 3D CNN [2] for action recognition in one domain (daytime) and would use it to learn actions in the other domain (night time). This pretrained network on the UCF101 action dataset has eight convolutions, five maximum pooling, and two fully connected layers, followed by a softmax output layer.

All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. After training all the network layers, we extract fc6 layer features from the trained network and call them C3D features. To extract these features, we follow the guidelines of [2]. Each video is split into clips, and each of the 16 frames is passed to the C3D network to extract fc6 activations. These fc6 activations are averaged to form a 4096-dimensional video descriptor followed by an L2 normalization. These representations are known as C3D video features.

In transfer learning, we will train a base network and then copy its first $n$ layers to the first $n$ layers of a target network. In our case, $n = 4$. The remaining layers of the target network will then randomly be initialized and trained toward the target task. Rather than freezing the transferred layers, we train all the layers. It is inspired by the recent work by Yosinski et al. [35].

### A. Action Classification

To apply transfer learning for the action classification task, we developed four separate networks based on the 3D CNN model, C3D network [2]. The first network was trained on daytime image data UCF101 [36]. Model A is 3D CNN trained on IR video sequences (InfAR data, and other available night IR data), Model B is 3D CNN trained on the night-time VIS spectrum only, and

Model C is 3D CNN trained on fused video sequences and color-enhanced video sequences.

### B. Action Detection

Action detection is more challenging compared to simple classification as it not only classifies the action but also provides its location. To achieve action detection, we use 3D feature-based zero-aliasing maximum-margin correlation filter as described below.

*1) Action-02MCF: 3D Feature-Based Zero-Aliasing Maximum-Margin Correlation Filter:*

The motivation of using correlation filter compared to end-to-end classification is simultaneous localization and detection of action instances. We train 3D correlation filters on fine-tuned features described in the previous section. These filters can be synthesized by calculating Fourier transfer of fine-tuned features. Correlation filters were initially developed in the seminal work of [37], which is a way of learning a template/filter in the frequency domain that, when correlated with a set of training signals, gives a desired response (correlation peak). A general correlation filter $h$ can be expressed as

$$h = arg_h min \sum_{i=1}^{N} ||h \otimes x_i - g_i||^2, \tag{1}$$

where $\otimes$ denotes the cross-correlation of the vector versions of the input signal $x_i$ and the template $h$, and $g_i$ is the vector version of the desired correlation output. If $N$ denotes the training feature vectors, $x_i$ denotes the $i$th feature vector.

Correlation filters are generally 2D as these filters work well on images. In our previous work [38], we have extended correlation filters in 3D for action recognition. Therefore, only a brief description of their optimization criteria is presented here as the complete design of Action-02MCF correlation filters is described in [38].

The correlation filter design problem is often considered as an optimization problem. If $N$ denotes the training feature vectors of length $M$, we can write the multiobjective function of the proposed correlation filter as follows:

$$\begin{aligned} h = \min_{\hat{h}} \Bigg( & \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{M} || \hat{f}_i^k \otimes \hat{h}_i^k \\ & - \hat{g}_i||_2^2, \lambda \sum_{k=1}^{M} ||h^k||_2^2 + C \sum_{i=1}^{N} \xi_i \Bigg), \\ s.t. \quad & y_i \left( \sum_{k=1}^{M} \cdot \hat{h}^{k'} \cdot \hat{f}_i^k \right) \geqslant u_i \xi_i \end{aligned}$$

Here, $\hat{f}$ is a feature vector, $\hat{h}$ is a frequency-domain filter, $\xi_i$ is a penalty term to penalize the training samples on the wrong side of the margin, $\lambda$ is a regularization parameter, whereas $C > 0$ denotes a trade-off parameter, $y_i$ is a class label (1: positive class; $-1$: negative class), and $u_i$ is the minimum peak value set to 1 for $N$ training samples and an $M$ number of features.

The regularization parameter $\lambda$ serves as a degree of importance that is given to misclassifications. So, intuitively, the larger the $\lambda$ grows, the fewer the wrongly classified examples are allowed (or the higher the price they pay in the loss function). Then, when $\lambda$ tends to infinity, the solution tends to the hard margin (allowing no misclassification). When $\lambda$ tends to 0 (without being 0), more misclassifications are allowed.

$C$ parameter controls the trade-off between achieving a low error on the training data and minimizing the norm of the weights and it tells the optimization how much misclassification to be avoided for each training example. For large values of $C$, the optimization will choose a smaller margin hyperplane if that hyperplane does a better work of achieving all the training points classified correctly. Conversely, a very small value of $C$ will cause the optimizer to look for a larger margin separating hyperplane, even if that hyperplane misclassifies more points.

*2) Notation:* Vectors are represented by the lowercase letters $f$, matrices are represented by uppercase letters $F$, ˆ represents variables in the frequency domain, and $t$ represents its transpose.

We extracted C3D features and named them Model-0-fc. We call these models as the pretrained model and extracted features as pretrained features. Model-A-fc was extracted from 3D CNN trained on IR video sequences (InfAR data, and other available night IR data), Model-B-fc was extracted from 3D CNN trained on the night-time VIS spectrum only, and, finally, Model-C-fc was extracted from 3D CNN trained on fused video sequences and color-enhanced video sequences.

## V. EXPERIMENTAL RESULTS AND DISCUSSION

This section describes our experimental data, setup, results, and performance comparison with discussion.

### A. Action Dataset and Experimental Setup

In the absence of any benchmark night-vision (NV) action dataset, we have recorded the NV action dataset using two different cameras. One of them is an IR camera, Raytheon Thermal IR-2000B, and the other is a low-light VIS camera, Panasonic WV-CP470. The thermal and visual videos are registered before the fusion process. In addition to these videos, this dataset includes 20 video sequences collected from the TNO image fusion dataset [25], Eden Project dataset [39], and Ohio State University thermal dataset. This dataset comprises eight action categories, including walking, wave1, wave2, stand-up, sit-down, clapping, pick-up, and running performed by different actors. It also includes videos from the IR action dataset [40], which contains 12 common human actions with IR video sequences. All action

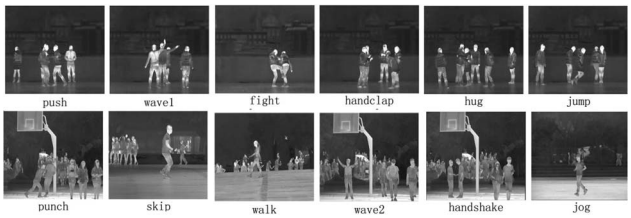



Fig. 3. Sample IR video instances for 12 action classes. Action categories include fight, handclasp, handshake, hug, jog, jump, punch, push, skip, walk, wave1, and wave2.

instances are displayed in Fig. 3. Sample actions include one-hand wave (wave1), multiple-hand wave (wave2), handclasp, jog, jump, walk, skip, hug, push, handshake, punch, and fighting action, each containing 50 video sequences, with 25 fp/s and resolution of 293 × 256. These actions are performed by 40 different actors. Each video clip lasts about 4 s on average. Some of these videos illustrate interactions between multiple actors.

### B. Experiment No. 1: The Role of Information Fusion in Terms of Error Rate

In this experiment, we intended to validate the significance of multisensor fusion data. We created training and validation sets with a ratio of 70:30. The base network was trained on IR data only as actions are more visible in the IR than in the VIS spectrum. The second network is prepared after transferring the first three layers of the C3D network and remaining layers of the fused dataset. Error rate is calculated for both training and validation sets. The experiment is shown in Fig. 5. It shows that in the case of fused data, the error rate is much lower than that of single-domain data for both training and validation sets.

### C. Experiment No. 2: The Role of Information Fusion in Terms of Recognition Accuracy

This experiment checks the classification accuracy. For validation, the leave-one-out cross-validation strategy is used. The results are shown in Table 1 in terms of recognition accuracy and the layers used during transfer learning. We experimented with different versions of our recognition framework to know the impact of information fusion on recognition performance. First, a baseline is developed as discussed in the transfer learning section. It is based on the daytime video action dataset, and we used the knowledge extracted from this network to fine-tune other networks. This network was the C3D network pretrained on the UCF101 dataset. Second, we fine-tuned other networks as described earlier in the transfer learning section. In addition, to know the effect of different layers in transfer learning, we fine-tuned different versions of each network to quantify the learning transferred from the base network.

Table 1
Average Recognition Accuracy of Three Different Models Versus Number of Layers Transferred from the Baseline Network

| Model used | No. of layers transferred | Recognition accuracy |
|---|---|---|
| Model A | 3, 4, 5 | 0.96, 0.91, 0.87 |
| Model B | 3, 4, 5 | 0.72, 0.71, 0.69 |
| Model C | 3, 4, 5 | 0.98, 0.93, 0.87 |

It shows that best recognition for each model is achieved if only three convolution layers are transferred as after this dataset specificity started increasing.

In this experiment, Model A is 3D CNN trained on IR video sequences (InfAR data, and other available night IR data), Model B is 3D CNN trained on the nighttime VIS spectrum only, and Model C is 3D CNN trained on fused video sequences and color-enhanced video sequences.

We calculated the average recognition accuracy for each case against the NV dataset and the results are displayed in Table 1. We found that network that uses both color and context information fusion alongside motion cues outperforms others. It demonstrated that fused information is significantly important in the action recognition process. Contextual information also plays an important role, especially in actions that involve full-body motion. Therefore, an information fusion of motion, color, and contextual cues can enhance action recognition performance.

### D. Experiment No. 3: Filter Performance for Action Detection and Localization

For this experiment, the training of Action-02MCF filters is performed for each action category. During the testing phase, a test action video is correlated with the synthesized filter to find the correlation peak.

To measure the detection and localization performance of the proposed filter, we utilize the probability of detection versus false alarms per second (FA/s). A performance metric denoted as $P$ is utilized, which is equal to the integration of a receiver operating characteristic (ROC) curve from 0 to 5 FA/s. An ideal ROC curve must have $P = 5$. To evaluate this performance,

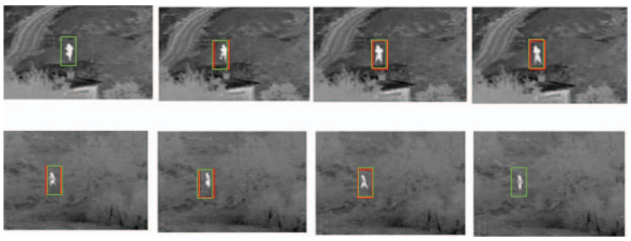


Fig. 4. Action instance detection in three NV action instances, where the red bounding box is the actual ground truth, while the green bounding box shows detection by 3D SDCF.

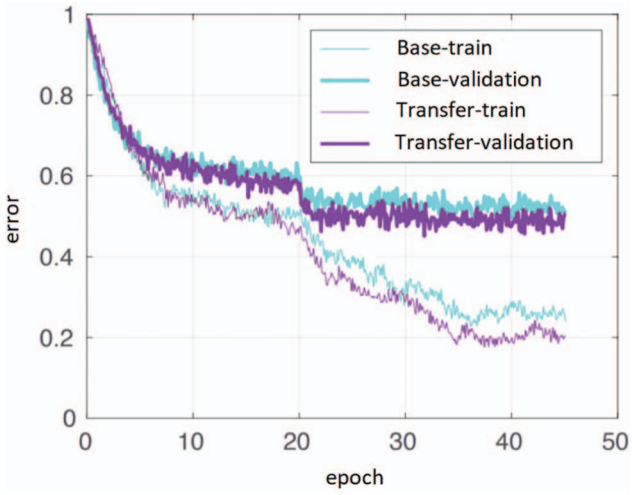



Fig. 5. The plot for the effect of transfer learning on the C3D network. The base C3D network trained on the NV dataset shows a high error rate in both training and validation sets compared to the C3D transferred network that is fine-tuned on the knowledge transferred by the fused dataset.

we applied the proposed filter to each test video and varied the threshold of the detection to generate ROC curves. The detection is labeled a true positive detection if the ground truth and the center of the bounding box lie within three frames of each other and the Euclidean distance is $\leq$8 pixels in the spatial domain to keep a >50% bounding box overlap for each action. We then plot the values of the performance metric $P$ against all actions and perform a comparison with a similar approach,

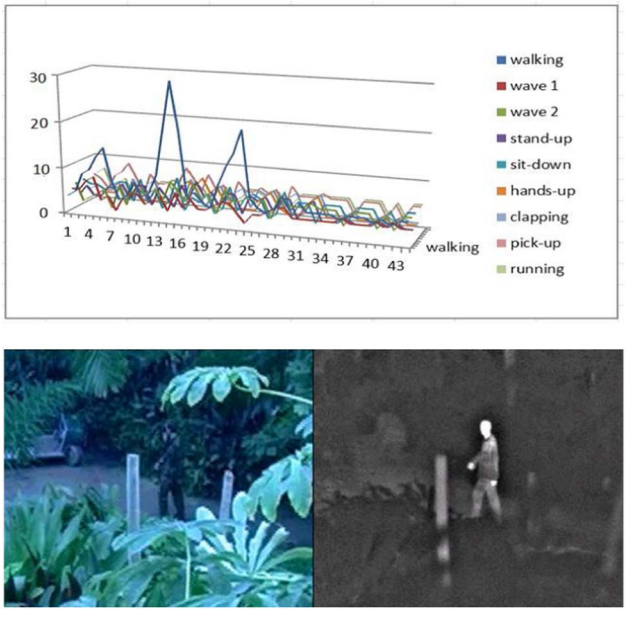



Fig. 6. Top: the plot of PSR by correlating the trained walking-Action-02MCF with a night-time video sequence. As visible from the plot, PSRs produced by walking-Action-02MCF are comparatively much higher than the responses by action filters for other actions. Bottom: a representative frame from the respective videos (both domains).

Action-DCCF filter [19]. The corresponding filter is selected due to similarity and code availability. Fig. 4 displays the sample detections with the original and estimated bounding boxes.

### E. Experiment No. 4: Quantitative Evaluation of Filter Robustness

We use another quantitative metric, named peak-to-sidelobe ratio (PSR) described in [41], which calculates the ratio of peak response to local surrounding response. Fig. 6 plots PSRs for walking action present in the test video sequence (sample frame displayed) using Action-03MCF for the walking action trained on the NV dataset.

## VI. CONCLUSION

In this paper, we explored and discussed the role of information fusion for automated action recognition. We use deep ConvNets for action recognition and used transfer learning to learn and transfer knowledge from a pretrained action network. In addition, we included an action-detection framework based on robust feature-based space–time action recognition called Action-02MCF. Experiments were conducted to know the effects of transfer learning, number of layers in transfer learning, and information fusion on improving the performance of action recognition at night time. We discovered that information fusion enhances action recognition performance as it improves the contextual information of night-vision data.

## REFERENCES

[1] J. Liu, J. Luo, and M. Shah
"Recognizing realistic actions from videos 'in the wild'," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1996–2003.

[2] S. Gong and T. Xiang
"Recognition of group activities using dynamic probabilistic networks," in
*Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 742–749.

[3] P. Siva and T. Xiang
"Action detection in crowd," in
*Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 1–11.

[4] I. Laptev, M. Marsza?ek, C. Schmid, and B. Rozenfeld
"Learning realistic human actions from movies," in
*Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[5] S. Islam, T. Qasim, M. Yasir, N. Bhatti, H. Mahmood, and M. Zia
"Single- and two-person action recognition based on silhouette shape and optical point descriptors,"
*Signal Image Video Process.*, vol. 12, no. 5, pp. 853–860, 2018.

[6] L. Pei, M. Ye, X. Zhao, T. Xiang, and T. Li
"Learning spatio-temporal features for action recognition from the side of the video,"
*Signal Image Video Process.*, vol. 10, no. 1, pp. 199–206, 2016.

[7] A. Ulhaq, X. S. Yin, J. He, and Y. Zhang
"On space–time filtering framework for matching human actions across different viewpoints,"

*IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1230–1242, 2018.

[8] A. Ulhaq, I. Gondal, and M. Murshed
"On dynamic scene geometry for view-invariant action matching," in
*Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3305–3312.

[9] A.-U. Haq, I. Gondal, and M. Murshed
"On temporal order invariance for view-invariant action recognition,"
*IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 203–211, 2012.

[10] D. Weinland, M. Özuysal, and P. Fua
"Making action recognition robust to occlusions and viewpoint changes," in
*Computer Vision–ECCV*. Berlin, Germany: Springer, 2010, pp. 635–648.

[11] L.-J. Li and L. Fei-Fei
"What, where and who? Classifying events by scene and object recognition," in
*Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[12] M. Marszalek, I. Laptev, and C. Schmid
"Actions in context," in
*Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2929–2936.

[13] D. Han, L. Bo, and C. Sminchisescu
"Selection and context for action recognition," in
*Proc. Int. Conf. Comput. Vis.*, 2009, vol. 9, pp. 1933–1940.

[14] Y.-G. Jiang, Z. Li, and S.-F. Chang
"Modeling scene and object contexts for human action retrieval with few examples,"
*IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 674–681, 2011.

[15] L. Kong, J. Qin, D. Huang, Y. Wang, and L. Van Gool
"Hierarchical attention and context modeling for group activity recognition," in
*Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 1328–1332.

[16] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes
"Hierarchical attention network for action segmentation,"
*Pattern Recognit. Lett.*, vol. 131, pp. 442–448, 2020.

[17] J. Han and B. Bhanu
"Human activity recognition in thermal infrared imagery," in
*Proc. IEEE Comput. Vis. Pattern Recognit. Workshop*, 2005, pp. 17–17.

[18] J. F. Li and W. G. Gong
"Application of thermal infrared imagery in human action recognition,"
*Adv. Mater. Res.*, vol. 121, pp. 368–372, 2010.

[19] H. Anwaar, G. Iqbal, and M. Murshed
"Contextual action recognition in multi-sensor nighttime video sequences," in
*Proc. Digit. Image Comput. Techn. Appl.*, 2011, pp. 256–261.

[20] A. Ulhaq
"Action recognition in the dark via deep representation learning," in
*Proc. IEEE Int. Conf. Image Process., Appl. Syst.*, 2018, pp. 131–136.

[21] G. I. Haq Anwaar and M. Manzur
"Action recognition using spatio-temporal distance classifier correlation filter," in
*Proc. Int. Conf. Digit. Image Comput. Techn. Appl.*, 2011, pp. 474–479.

[22] A. Ulhaq, X. Yin, J. He, and Y. Zhang
"FACE: fully automated context enhancement for night-time video sequences,"
*J. Vis. Commun. Image Representation*, vol. 40, pp. 682–693, 2016.

[23] A. Ulhaq, I. Gondal, and M. Murshed
"Scarf: semi-automatic colorization and reliable image fusion," in
*Proc. Int. Conf. Digit. Image Comput. Techn. Appl.*, 2010, pp. 435–440.

[24] A. Ulhaq, A. Mirza, and S. Qamar
"An optimized image fusion algorithm for night-time surveillance and navigation," in
*Proc. IEEE Symp. Emerg. Technol.*, 2005, pp. 138–143.

[25] A. Toet, M. J. de Jong, M. A. Hogervorst, and I. T. Hooge
"Perceptual evaluation of colorized nighttime imagery," in
*Proc. IS&T/SPIE Electron. Imaging*, 2014, pp. 901412–901412.

[26] G. Gkioxari, R. Girshick, and J. Malik
"Contextual action recognition with R*CNN," in
*Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1080–1088.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton
"Imagenet classification with deep convolutional neural networks," in
*Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] S. Ji, W. Xu, M. Yang, and K. Yu
"3D convolutional neural networks for human action recognition,"
*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, 2013.

[29] J. F. Li and W. G. Gong
"Application of thermal infrared imagery in human action recognition,"
*Adv. Mater. Res.*, vol. 121, pp. 368–372, 2010.

[30] J. Han and B. Bhanu
"Human activity recognition in thermal infrared imagery," in
*Proc. IEEE Comput. Vis. Pattern Recognit. Workshop*, 2005, pp. 17–17.

[31] J. Du, H. Zhou, K. Qian, W. Tan, Z. Zhang, L. Gu, and Y. Yu
"RGB-IR cross input and sub-pixel upsampling network for infrared image superresolution,"
*Sensors*, vol. 20, no. 1, p. 281, 2020.

[32] I. Shopovska, L. Jovanov, and W. Philips
"Deep visible and thermal image fusion for enhanced pedestrian visibility,"
*Sensors*, vol. 19, no. 17, p. 3727, 2019.

[33] H. Anwaar, G. Iqbal, and M. Murshed
"Automated multi-sensor color video fusion for nighttime video surveillance," in
*Proc. IEEE Symp. Comput. Commun.*, 2010, pp. 529–534.

[34] P. Shah, B. C. S. Reddy, S. N. Merchant, and U. B. Desai
"Context enhancement to reveal a camouflaged target and to assist target localization by fusion of multispectral surveillance videos,"
*Signal Image Video Process.*, vol. 7, no. 3, pp. 537–552, 2013.

[35] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson
"How transferable are features in deep neural networks?" in
*Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[36] K. Soomro, A. R. Zamir, and M. Shah
"UCF101: a dataset of 101 human action classes from videos in the wild.," CRCV-TR-12-01, November, 2012.

[37] C. F. Hester and D. Casasent
"Multivariant technique for multiclass pattern recognition,"
*Appl. Opt.*, vol. 19, no. 11, pp. 1758–1761, 1980.

[38] A. Ulhaq, X. Yin, Y. Zhang, and I. Gondal
"Action-02MCF: a robust space–time correlation filter for action recognition in clutter and adverse lighting conditions," in
*Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.* Berlin, Germany: Springer, 2016, pp. 465–476.

[39] J. Lewis, S. Nikolov, A. Loza, E. F. Canga, N. Cvejic, J. Li, A. Cardinali, C. Canagarajah, D. Bull, T. Riley, D. Hickman,

and M. I. Smith
"The Eden Project multisensory data set," The Online Resource for Research in Image Fusion (ImageFusion.org), 2006.

[40] C. Gao, Y. Du, J. Liu, J. Lv, L. Yang, D. Meng, and A. G. Hauptmann
"InfAR dataset: infrared action recognition at different times,"
*Neurocomputing*, vol. 212, pp. 36–47, 2016.

[41] B. V. K. V. Kumar, A. Mahalanobis, and R. D. Juday
*Correlation Pattern Recognition*. New York, NY, USA: Cambridge University Press, 2005.

**Anwaar Ulhaq** is serving as a lecturer in School of Computing and Mathematics and deputy leader in Machine Vision and Digital Health Research Group at Charles Sturt University, New South Wales, Australia. Anwaar holds PhD (Artificial Intelligence) from Monash University, Australia. He has also worked a research fellow at Institute for Sustainable Industries & Liveable Cities, Victoria University, Australia. His research interests include signal and image processing, deep learning, data analytics and computer vision.

# A Nonparametric Bayesian Compressive Sensing Classification

**RUILONG CHEN**
**MATTHEW HAWES**
**LYUDMILA MIHAYLOVA**

**This paper presents a novel nonparametric backpropagation Bayesian compressive sensing (BBCS) classification approach. While the state-of-the-art parametric classifiers such as logistic regression require model training and can result in inadequate models, the developed approach does not require model training. It is combined with a column-based subspace sampling process and can deal efficiently with uncertainties and highly computational tasks. Validation on a publicly available vehicle logo dataset shows that the proposed classifier can achieve up to 98% recognition accuracy as compared with the state-of-the-art nonparametric classifiers. Compared with the generic Bayesian compressive sensing classification, the proposed approach decreases the mean number of misclassifications by 87% along with 68% reduction of the computational time. The robustness of the BBCS approach is demonstrated over scene recognition tasks, and its outperformance over the AlexNet convolutional neural network algorithm is demonstrated in noisy conditions. The proposed BBCS approach is generic and can be used in different areas; for example, it has shown robustness over the CIFAR-10 dataset.**

Manuscript received January 22, 2019; revised April 13, 2019 and November 7, 2019; released for publication June 30, 2020.

Associate Editor: Marcus Baum.

The authors are with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (E-mail: chen8131928@gmail.com, hawes.matthewblair@gmail.com, l.s.mihaylova@sheffield.ac.uk).

We appreciate the support of the "SETA project: An open, sustainable, ubiquitous data and service for efficient, effective, safe, resilient mobility in metropolitan areas" funded from the European Union's Horizon 2020 research and innovation program under grant agreement no. 688082.

1557-6418/20/$17.00 © 2020 JAIF

## I. INTRODUCTION

A number of parametric classifiers such as the linear support vector machine (SVM) [1]–[4] and logistic regression [5] have been developed for vehicle logo recognition (VLR) and traffic scene recognition (TSR). Deep learning models such as convolutional neural networks (CNNs) and capsule networks have been applied to VLR [6], [7]. These parametric classifiers assume a functional distribution of the data [8]. The relationship between the label and the input data is modeled using a fixed number of parameters. An advantage of parametric classifiers is that once the number of parameters is determined, it would not change later as nonparametric methods do. However, in practice, parametric classifiers could result in an inadequately trained model due to inappropriate assumptions of prior distributions, leading to inappropriate predictions in the testing phase [8], [9].

On the contrary, nonparametric classifiers do not make assumptions about the distribution representing the data [8]. They do not have a model with a fixed number of parameters. Instead, the number of parameters increases with the size of the training dataset [10]. This in turn increases the computational complexity.

The $K$-nearest neighbor ($K$NN) approach is a commonly used nonparametric approach that is often used for classification [11], [12]. However, the $K$NN approach is not robust to outliers and to data with high dimensionality. This is because the shortest distance is not necessarily the best match to the testing data, especially when the number of training data is limited [8], [13]. Besides, the $K$NN approach has been shown to be vulnerable to noise effects [5].

A nonparametric classification approach based on sparse representation proposed by Wright et al. [14] has proven to be more accurate than the linear SVM and the $K$NN classifier for face recognition. The sparse representation classifier (SRC) [14] assumes that the testing data can be represented as a linear combination of the training dataset. A weight vector is generated with each element representing a corresponding coefficient in the linear combination. By splitting the weights according to their associated classes (with the remaining set to be zero valued), the weights in the correct class should reconstruct the original data with a minimum error. However, the high computational costs of the SRC can be a problem. In addition, the SRC works only when the system is under-determined [15]. In practice, this criterion cannot be met when there is a lack of training data.

Recently, the Bayesian compressive sensing (BCS) [16] approach has been efficiently applied to synthetic aperture radar target classification [17], image reconstruction [18], [19] and phonetic classification [20]. The Bayesian approach could potentially provide an alternative to the $l_1$-norm minimization for optimizing the linear combination coefficients required for the classification framework. Similarly to Zhou et al. [21], by comparing the magnitudes of the coefficients, the testing

data can then be classified by assigning them to the class whose coefficients have the highest $l_2$-norm magnitude.

The methods proposed in [22] and [23] map the data into a reduced dimensional space, using principal component analysis (PCA). However, these new latent spaces are different from the original space and make the original data difficult to interpret. To combat this issue, a column-based subspace sampling data representation can be used [24]–[26]. In this case, it is still possible to work in the original space, just with fewer data points.

In order to cope with various sources of uncertainties that many of the existing classification algorithms face, this paper proposes a new solution that provides robustness to insufficient training data and to noises. The key contributions of this work can be summarized as follows:

1) A new backpropagation BCS (BBCS) classifier is developed that represents efficiently the data and solves the classification problem as an optimization problem. The Euclidean distance between the constructed testing data and the original testing data is minimized. This process increases the recognition accuracy.

The BBCS incorporates a data reduction process that further decreases the computational costs. The column-based subspace sampling representation selects informative data points from the dataset. Compared with the PCA that transforms the original data into a new latent space, the column-based subspace sampling method chooses the best data directly from the original space. This process significantly decreases the computational costs and facilitates the interpretation in this reduced dimensional space.

2) The developed BBCS approach is validated and evaluated over noisy data and compared with state-of-the-art nonparametric classifiers: the *K*NN algorithm, the SRC, and the BCS algorithm. The BBCS is more robust than the *K*NN classifier. Compared with the BCS, the proposed approach decreases the mean number of misclassifications by 87% and reduces the computational cost compared with the SRC algorithm.

The rest of this paper is organized as follows. Section II introduces the general sparse representation classification framework. Section III presents the BCS approach. Section IV introduces the developed backpropagation BCS classifier approach and the column-based subspace sampling method. Section V presents performance validation on VLR and discussions of the results. Sections VI and VII present performance validation on vehicle scene recognition and the CIFAR 10 dataset. Section VIII summarizes the findings. The appendices contain the full derivation of the marginal likelihood function and its maximization.

## II. CLASSIFICATION FRAMEWORK BASED ON SPARSE REPRESENTATION

The SRC, BCS classifier, and BBCS classifier assume that the testing data $\mathbf{x}^* \in \mathbb{R}^{M\times 1}$ can be represented as a linear combination of the training samples $\mathbf{X} \in \mathbb{R}^{M\times N}$, where $M$ is the length of the vector data and $N$ gives the number of entries in the training dataset. When applying to images, each image is represented by an image feature vector rather than by pixels of the raw image. Therefore, $M$ refers to the length of the feature vector representing the image. Feature-based methods such as the scale-invariant feature transform (SIFT) [27] and CNN [28] can represent an image using a vector rather than a matrix representation.

A testing image denoted by image feature $\mathbf{x}^*$ is represented with the linear model

$$\mathbf{x}^* = \mathbf{X}\mathbf{w} + \mathbf{z}, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^{N\times 1}$ is a weight vector controlling the contribution of each image feature in the training dataset to the linear combination representing the testing image feature, $\mathbf{z} \in \mathbb{R}^{M\times 1}$ is a bounded noise term with $||\mathbf{z}||_2 \leqslant \epsilon$, $||\cdot||_2$ is the $l_2$-norm, and $\epsilon$ is a small positive constant. The solution to equation (1), $\mathbf{w}$, is obtained by minimizing the $l_2$-norm:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}(||\mathbf{w}||_2), \quad s.t.\ ||\mathbf{x}^* - \mathbf{X}\mathbf{w}||_2 \leqslant \epsilon, \tag{2}$$

where $\hat{\mathbf{w}} \in \mathbb{R}^{N\times 1}$ is the estimated weight vector. However, when $N > M$, equation (1) corresponds to an under-determined system and there is no unique solution by using conventional methods [14], [29].

The SRC classification method [14] assumes that a testing image feature can be sufficiently represented by a dictionary for its corresponding class. Therefore, the solution is naturally sparse as coefficients for unrelated classes are zero valued. For instance, if there are 20 classes, only approximately 5% of the coefficients in $\hat{\mathbf{w}}$ will have nonzero values [14]. In fact, the sparser the recovered $\mathbf{w}$ is, the easier it is to accurately classify the testing image feature $\mathbf{x}^*$ [14]. This motivates the use of the $l_0$-norm to find the sparest solution for $\mathbf{w}$ in equation (1).

However, $l_0$-norm minimization is an NP-hard problem. Instead, an $l_1$-norm minimization is typically used as an approximation [15], [30], [31], giving

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}}(||\mathbf{w}||_1), \quad s.t.\ ||\mathbf{x}^* - \mathbf{X}\mathbf{w}||_2 \leqslant \epsilon. \tag{3}$$

The solution to the $l_1$-minimization in equation (3) can be found by linear programming methods such as the basis pursuit [32] or the orthogonal matching pursuit [33] methods. The solution to equation (1) gives the optimal $\mathbf{w}$ for classification purposes in the SRC [14].

## III. BAYESIAN COMPRESSIVE SENSING

The BCS method [16] provides an alternative to the $l_1$-norm minimization method by incorporating prior knowledge within the Bayesian framework. Since the testing image feature can be represented as a linear combination (1) of the training images, the relative importance of each training image feature is controlled by the weight vector $\mathbf{w}$. The vector $\mathbf{w}$ can be separated into $\mathbf{w}_v$ and $\mathbf{w}_e$, where $\mathbf{w}_v$ contains the significant weights and $\mathbf{w}_e$

the remaining negligible weights. Hence, $\mathbf{w} = \mathbf{w}_v + \mathbf{w}_e$ and equation (1) can be written as

$$\mathbf{x}^* = \mathbf{X}\mathbf{w}_v + \mathbf{X}\mathbf{w}_e + \mathbf{z}. \tag{4}$$

Both $\mathbf{X}\mathbf{w}_e$ and $\mathbf{z}$ can be approximated as zero-mean Gaussian noises [16], allowing equation (4) to be written as

$$\mathbf{x}^* = \mathbf{X}\mathbf{w}_v + \mathbf{n}, \tag{5}$$

where $\mathbf{n} = \mathbf{X}\mathbf{w}_e + \mathbf{z}$. The variance of $\mathbf{n}$ is then given by $\mathbf{\Sigma}_n = \sigma^2\mathbf{I}_M$, where $\mathbf{I}_M$ is an identity matrix of size $M \times M$. Note that each entry in $\mathbf{n}$ has the same variance $\sigma^2$ and hence the likelihood function can be given by

$$p(\mathbf{x}^*|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-M/2} \exp\left\{-\frac{||\mathbf{x}^* - \mathbf{X}\mathbf{w}||_2^2}{2\sigma^2}\right\}, \tag{6}$$

rather than in the standard multivariate form that includes the covariance matrix $\mathbf{\Sigma}_n$. In equation (6) and in the following equations, the subscript $v$ of $\mathbf{w}$ is dropped for conciseness.

The elements of $\mathbf{w}$ are assumed to have a zero-mean Gaussian distribution. This is given by

$$\begin{aligned} p(\mathbf{w}|\boldsymbol{\alpha}) &= \prod_{i=1}^{N} N(w_i|0, \alpha_i^{-1}) \\ &= \prod_{i=1}^{N} (2\pi\alpha_i^{-1})^{-1/2} \exp\left\{-\frac{1}{2}\alpha_i w_i^2\right\} \\ &= (2\pi)^{-N/2} |\mathbf{A}|^{1/2} \exp\left\{-\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w}\right\}, \end{aligned} \tag{7}$$

where $\mathbf{A} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N)$ and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^{\mathrm{T}}$, $\alpha_i$ is a precision value, and $|\cdot|$ denotes the determinant. Furthermore, Gamma hierarchical priors are considered over $\alpha_i$ and $\sigma^2$:

$$p(\boldsymbol{\alpha}) = \prod_{i=1}^{N} \mathrm{Gamma}(\alpha_i|a, b), \tag{8}$$

$$p(\sigma^2) = \mathrm{Gamma}(\sigma^2|c, d), \tag{9}$$

where $a, b, c$, and $d$ are shape and scale parameters.

The overall prior over $\mathbf{w}$ can be evaluated by marginalizing over the hyperparameters $\boldsymbol{\alpha}$:

$$p(\mathbf{w}|a, b) = \prod_{i=1}^{N} \int_0^{\infty} N(w_i|0, \alpha_i^{-1})\mathrm{Gamma}(\alpha_i|a, b)d\alpha_i. \tag{10}$$

Since the prior of $\mathbf{w}$ is assumed to be a zero-mean Gaussian distribution that conjugates to a Gamma prior, the probability density $p(\mathbf{w}|a, b)$ corresponds to the Student's $t$-distribution [34]. This achieves sparsity as the Student's $t$-distribution can be strongly peaked at $w_i = 0$ with appropriate choices of $a$ and $b$ [16], [34].

Combining the likelihood function and the prior given by equations (6) and (7), respectively, the posterior distribution of the weights can be found from

$$p(\mathbf{w}|\mathbf{x}^*, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\mathbf{x}^*|\mathbf{w}, \sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2)}. \tag{11}$$

As the likelihood function and prior are both Gaussian, the posterior distribution over $\mathbf{w}$ is also a Gaussian distribution:

$$\begin{aligned} p(\mathbf{w}|\mathbf{x}^*, \boldsymbol{\alpha}, \sigma^2) &= N(\mathbf{w}|\boldsymbol{\mu}, \mathbf{\Sigma}), \\ &= (2\pi)^{-N/2}|\mathbf{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{w} - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{w} - \boldsymbol{\mu})\right\}, \end{aligned} \tag{12}$$

where the mean vector and covariance matrix, respectively, are given by

$$\boldsymbol{\mu} = \sigma^{-2}\mathbf{\Sigma}\mathbf{X}^{\mathrm{T}}\mathbf{x}^* \tag{13}$$

and

$$\mathbf{\Sigma} = (\mathbf{A} + \sigma^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}. \tag{14}$$

Note that $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are dependent on $\sigma^2$ and $\boldsymbol{\alpha}$. Therefore, the goal is to find the posterior probability density function over all the unknown parameters given the training image features and the testing image feature. This means finding the values for $\mathbf{w}$, $\boldsymbol{\alpha}$, and $\sigma^2$ that maximize the following posterior probability density function:

$$p(\mathbf{w}, \boldsymbol{\alpha}, \sigma^2|\mathbf{x}^*) = p(\mathbf{w}|\mathbf{x}^*, \boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha}, \sigma^2|\mathbf{x}^*). \tag{15}$$

Finding the optimal $\mathbf{w}$, $\boldsymbol{\alpha}$, and $\sigma^2$ involves two steps. First, for the current values of $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, the values of $\boldsymbol{\alpha}$ and $\sigma^2$ are calculated to maximize $p(\boldsymbol{\alpha}, \sigma^2|\mathbf{x}^*)$. Then, these values are substituted to re-evaluate $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$. This process is then repeated until a convergence criterion is met. In the first step, $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are fixed then, and maximizing equation (15) is equivalent to maximizing

$$p(\boldsymbol{\alpha}, \sigma^2|\mathbf{x}^*) = \frac{p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})p(\sigma^2)}{p(\mathbf{x}^*)}, \tag{16}$$

where the denominator is independent of $\boldsymbol{\alpha}$ and $\sigma^2$. Therefore, only $p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2)p(\boldsymbol{\alpha})p(\sigma^2)$ has to be maximized. Furthermore, by selecting $a$, $b$, $c$, and $d$ to be small positive values, there are flat, uninformative priors over $\alpha$ and $\sigma^2$ [34]. Maximizing equation (16) is approximately equal to maximizing the marginal likelihood:

$$p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2) = \int p(\mathbf{x}^*|\mathbf{w}, \sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w}, \tag{17}$$

with $p(\mathbf{x}^*|\mathbf{w}, \sigma^2)$ and $p(\mathbf{w}|\boldsymbol{\alpha})$ being given in equations (6) and (7), respectively. The full derivation of the marginal likelihood function is given in Appendix A.

Equation (17) is a convolution of two zero-mean Gaussians and the logarithm of the result gives

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}, \sigma^2) &= \ln\left(p(\mathbf{x}^*|\boldsymbol{\alpha}, \sigma^2)\right) \\ &= \ln\left(N(\mathbf{x}^*|0, \mathbf{C})\right) \\ &= -\frac{1}{2}\left(M\ln(2\pi) + \ln|\mathbf{C}| + \mathbf{x}^{*T}\mathbf{C}^{-1}\mathbf{x}^*\right), \end{aligned} \tag{18}$$

where the $M \times M$ matrix $\mathbf{C}$ is given by

$$\mathbf{C} = \sigma^2 \mathbf{I}_M + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}}. \tag{19}$$

A type II maximum likelihood approximation is used to estimate $\boldsymbol{\alpha}$ and $\sigma^2$ [34], which gives

$$\alpha_i^{\text{new}} = \frac{1 - \alpha_i \Sigma_{ii}}{\mu_i^2}, \tag{20}$$

$$(\sigma^{\text{new}})^2 = \frac{||\mathbf{x}^* - \mathbf{X}\boldsymbol{\mu}||_2^2}{M - \sum_i^N (1 - \alpha_i \Sigma_{ii})}, \tag{21}$$

where $\Sigma_{ii}$ is the $i$th diagonal element of $\boldsymbol{\Sigma}$ in equation (14). The parameters $\boldsymbol{\alpha}$ and $\sigma^2$ are functions of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, while $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are functions of $\boldsymbol{\alpha}$ and $\sigma^2$. This leads to an iterative algorithm to update each variable until a convergence criterion has been met. The derivation of the update equations in (20) and (21) is provided in Appendix B.

## IV. THE PROPOSED BACKPROPAGATION BAYESIAN COMPRESSIVE SENSING CLASSIFIER AND COLUMN-BASED SUBSPACE SAMPLING

### A. Backpropagation Bayesian Compressive Sensing Classifier

Given that the training images in $\mathbf{X}$ belong to $K$ classes, where the class label $i \in \{1, 2, \ldots, K\}$, the training image features can be separated according to their labels. This gives $\mathbf{X} = [\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^i, \ldots, \mathbf{X}^K]$, where $\mathbf{X}^i$ contains all of the training image features belonging to the $i$th class. Suppose that there are $n_i$ samples in the $i$th class, then all of the training image feature vectors in the $i$th class are given by $\mathbf{X}^i = [\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_{n_i}^i]$. Notice that this process only separates the training image feature vectors by their labels, and the total number of training image feature vectors does not change. Hence, $\sum_i^K n_i = N$.

Therefore, an original testing image feature vector can be reconstructed by using the estimated weight vector $\hat{\mathbf{w}}$:

$$\tilde{\mathbf{x}}^* = [\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^K] \begin{bmatrix} \hat{\mathbf{w}}^1 \\ \hat{\mathbf{w}}^2 \\ \vdots \\ \hat{\mathbf{w}}^K \end{bmatrix}, \tag{22}$$

where $\tilde{\mathbf{x}}^*$ is an estimate of the original image feature vector $\mathbf{x}^*$ and $\hat{\mathbf{w}} = [[\hat{\mathbf{w}}^1]^{\mathrm{T}}, [\hat{\mathbf{w}}^2]^{\mathrm{T}}, \ldots, [\hat{\mathbf{w}}^K]^{\mathrm{T}}]^{\mathrm{T}}$. Based on the assumption that the testing image feature vector is a linear combination of a few image feature vectors from its corresponding class, nonzero-valued elements in $\hat{\mathbf{w}}$ should be only in $\hat{\mathbf{w}}^i$ if the testing image feature vector belongs to class $i$. The BCS approach [17], [20] assigns the testing image feature vector to class $i$ if it has the highest norm-2 magnitude of $\hat{\mathbf{w}}^i$.

However, when there are training image feature vectors with no or a very small number of points of interest, most of the resulting feature vectors are zero valued. This would allow large weight values in $\hat{\mathbf{w}}$ without detrimentally affecting the likelihood value when evaluating equation (6). These inappropriately large weight values can lead to data being misclassified when using the $l_2$-norm of the weights as a classification mechanism. To overcome this problem, this work proposes a classification approach based on a backpropagation process as described below. Note that the backpropagation here is a reconstruction process, in which the weights are propagated back in order to reconstruct the input feature vector. This is different with the backpropagation process used in neural network.

The proposed approach reconstructs the testing image feature vector by a BCS process in which the image feature vectors are represented by equation (22). Similar to SRC, the weight vector $\hat{\mathbf{w}}$ is separated into $K$ vectors with each vector keeping the value in its corresponding weight locations and setting the remaining values to zero:

$$\begin{bmatrix} \hat{\mathbf{w}}^1 \\ \hat{\mathbf{w}}^2 \\ \vdots \\ \hat{\mathbf{w}}^K \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{w}}^1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{w}}^2 \\ \vdots \\ \mathbf{0} \end{bmatrix} + \cdots + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \hat{\mathbf{w}}^K \end{bmatrix},$$

$$\hat{\mathbf{w}} = \tilde{\mathbf{w}}^1 + \tilde{\mathbf{w}}^2 + \cdots + \tilde{\mathbf{w}}^K, \tag{23}$$

where $\tilde{\mathbf{w}}^i \in \mathbb{R}^{N \times 1}$ and $i \in \{1, 2, \ldots, K\}$. Each $\tilde{\mathbf{w}}^i$ is used to reconstruct the testing image feature $\mathbf{x}_{\text{cons}}^i$ as follows:

$$\mathbf{x}_{\text{cons}}^i = \mathbf{X}\tilde{\mathbf{w}}^i. \tag{24}$$

The testing image feature vector $\mathbf{x}^*$ is assigned to a class corresponding to the most similar reconstructed image feature vector. More specifically, if the testing image feature vector recovered by $\tilde{\mathbf{w}}^i$ has the highest similarity with the original testing image feature vector $\mathbf{x}^*$, then this testing image feature vector can be classified into the $i$th class. In order to compute the similarity between the image feature vector recovered by $\tilde{\mathbf{w}}^i$ and the original image feature vector $\mathbf{x}^*$, an error term is defined for each class:

$$\text{Err}(i) = ||\mathbf{x}^* - \mathbf{x}_{\text{cons}}^i||_2. \tag{25}$$

Then, the testing image feature vector can be classified into the class that gives the minimum error. SRC, BCS, and BBCS classifiers all need a dictionary composed by training data; hence, they are naturally inefficient for large datasets.

### B. Column-Based Subspace Sampling

Estimating the coefficients in equation (5) for BBCS can be time consuming when $\mathbf{X}$ is high dimensional. PCA can solve this problem by mapping the data into a lower dimensional data space. However, as the space has been altered, each entry can be difficult to interpret. The column-based subspace sampling method can

avoid these problems [24]. It selects the "best" subset of $h$ columns from $\mathbf{X}$, where $h < N$.

Let $\mathbf{X}_k$ represent the "best" rank-$k$ approximation to $\mathbf{X}$ by singular value decomposition. The output matrix $\mathbf{D} \in \mathbb{R}^{M\times h}$ consists of $h$ columns from $\mathbf{X}$ such that the inequality in equation (26) is valid for a probability of at least $1-\delta$.

$$||\mathbf{X} - \mathbf{D}\mathbf{D}^{+}\mathbf{X}||_F \leqslant (1+\rho)||\mathbf{X} - \mathbf{X}_k||_F, \qquad (26)$$

where $||\cdot||_F$ is the Frobenius norm, $\mathbf{D}^{+}$ is a Moore–Penrose generalized inverse of $\mathbf{D}$, $\rho$ is an error parameter, and $\delta$ is the failure probability.

Define a score for each column in "$\mathbf{X}$" in the following form:

$$\pi_j = \frac{1}{k}\sum_{\xi=1}^{k}(\mathbf{v}_j^{\xi})^2, \qquad (27)$$

where $\mathbf{v}_j^{\xi}(j = 1, 2, \ldots, N)$ is the $j$th coordinate of $\mathbf{v}^{\xi}$ and $\mathbf{v}^{\xi} \in \mathbb{R}^{N\times 1}$ $(\xi = 1, 2, \ldots, k)$ is the top right $k$ singular vectors of $\mathbf{X}$. A random sampling process is applied on $\mathbf{X}$ and the $j$th column of $\mathbf{X}$ is adopted with probability $\min\{1, h\pi_j\}$, where $h = O(k\log k/\rho^2)$. All the adopted columns then generate the target matrix $\mathbf{D}$, with $h$ examples to represent the original dataset. The detailed proof is given in [24] and [26].

## V. PERFORMANCE EVALUATION FOR VEHICLE LOGO RECOGNITION

The proposed BBCS can be used as a generic classifier. In this paper, we implemented for VLR. Recognizing vehicle logos and traffic scenes is of paramount importance for intelligent transportation systems, especially for traffic monitoring and management. The vehicle logo is one of its most distinguishable vehicle features [11] and as part of systems it can facilitate detecting fraudulent plates even when the observed logo is not available in the police security database [35]. As a result, this could give robust vehicle identification also in commercial investigations [1] and document retrieval systems [36]. VLR is also plays a crucial role in self-driving cars, traffic safety [37], and surveillance [38].

In this section, the open VLR dataset provided by Huang et al. [6] is used to evaluate the proposed classification approach. It has 10 categories and each category contains 1000 training images and 150 testing images. All images have a size of $70\times70$ pixels. Fig. 1 shows an example of the 10 vehicle categories by randomly choosing one image from each category in the training dataset.

The local descriptor SIFT [27] and the bag of words [39] model are applied in order to represent images before the classification. All SIFT interest points are clustered in order to generate a dictionary with $M$ words. In the representation stage, interest points from an image are replaced by their nearest words in the dictionary. This allows each image to be represented as a feature vector of length $M$, where $M$ is the number of centroids in the clustering process in the bag of words model. The value in each entry of the vector is the normalized frequency of each word that appeared in an image. Increasing $M$ gives more detailed information about the feature but increases the computation costs. Further details about representation models can be found in [40] and [41].

Fig. 1. Vehicle logo dataset.

The performance evaluation is conducted in MATLAB on a computer with the following specification: Intel CPU i5-4590 (3.4 GHz) and 8 GB of RAM. The open-source library VLFeat [42] is applied for extracting the SIFT features. A comparison is made with the SRC (implemented using CVX [43], [44]), BCS classifier, and $K$NN classifier. In our experiment, $K = 1$ achieves the best result for clear images. Different $K$ values influence the result when images are noisy, while the prior knowledge of images is unknown. Therefore, as it is commonly done in the literature [14], [25], here a value of $K = 1$ is selected for all considered examples. The performance of each method is evaluated in terms of accuracy (percentage of correctly classified images), the total number of misclassified images, and the computation time (to indicate the relative computational complexities).

### A. Classification Comparisons for Vehicle Logo Recognition

This subsection compares the performances of the classification methods when applied to the images that are provided in the dataset [6]. The simulation is repeated 30 times, and the average accuracy is found and given with the corresponding standard deviation. The computation time and number of misclassified images are also given as the mean results for all the simulation runs.

Table I shows that the BBCS classifier achieves the highest accuracy of 98.91%. Table I also indicates that the BCS classifier is less accurate than the SRC and BBCS classifier. For example, when $M = 300$, the BCS classifier incorrectly classifies 138 images, while this is reduced to 17 images for the BBCS classifier. In this case, the number of misclassifications is reduced by 88% without increasing the computational cost. For all the values of $M$ considered, there was a mean reduction in the number of misclassified logos of 87% for the BBCS classifier as compared to the BCS classifier. The computation times in Table I show that this improvement in

Table I
Nonparametric Classifiers' Comparison Using SIFT Descriptors with $M = 100, 200, 300, 400$, and 500

| Classifiers | *K*NN | SRC | BCS | BBCS |
|---|---|---|---|---|
| $M = 100$ Accuracy (%) | $98.29 \pm 0.36$ | **98.30** $\pm 0.44$ | $92.17 \pm 0.77$ | $98.24 \pm 0.32$ |
| Misclassified images | 25.65 | **25.50** | 117.45 | 26.40 |
| Time (s) | **0.97** | 6357 | 868 | 868 |
| $M = 200$ Accuracy (%) | $98.72 \pm 0.24$ | **98.73** $\pm 0.25$ | $91.36 \pm 0.54$ | $98.60 \pm 0.28$ |
| Misclassified images | 19.20 | **19.05** | 129.60 | 21 |
| Time (s) | **1.84** | 7804 | 2358 | 2358 |
| $M = 300$ Accuracy (%) | $98.63 \pm 0.27$ | $98.78 \pm 0.24$ | $90.77 \pm 0.75$ | **98.86** $\pm 0.22$ |
| Misclassified images | 20.55 | 18.30 | 138.45 | **17.10** |
| Time (s) | **2.70** | 8360 | 3120 | 3120 |
| $M = 400$ Accuracy (%) | $98.67 \pm 0.30$ | $98.83 \pm 0.23$ | $90.37 \pm 0.77$ | **98.91** $\pm 0.24$ |
| Misclassified images | 19.95 | 17.55 | 144.45 | **16.35** |
| Time (s) | **3.54** | 9116 | 3360 | 3360 |
| $M = 500$ Accuracy (%) | $98.74 \pm 0.23$ | **98.86** $\pm 0.19$ | $90.25 \pm 0.95$ | $98.84 \pm 0.25$ |
| Misclassified images | 18.90 | **17.10** | 146.25 | 17.40 |
| Time (s) | **4.17** | 9582 | 3497 | 3497 |

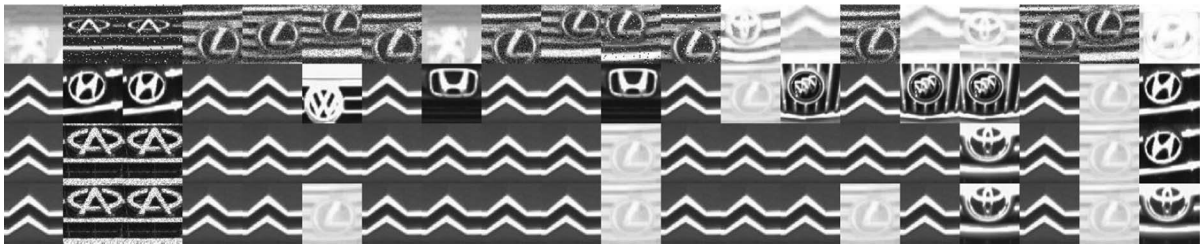


Fig. 2. The first row illustrates some challenge images, and the second, third, and fourth rows are the corresponding results classified by *K*NN, SRC, and BBCS, respectively.

classification accuracy comes without an increase in computational complexity.

The SRC and BBCS classifier give very similar classification accuracies. However, the BBCS classifier has a significant advantage in terms of computational costs. For the example, when $M = 300$, the proposed BBCS classifier reduces the computational cost by 63% when compared with the SRC while giving a slightly improved accuracy compared with the SRC algorithm. When comparing the computation times of the proposed BBCS classifier to the SRC, for all values of $M$ considered, there is a mean reduction in the computation time of 68%. It only takes about 2 s to recognize an image using the BBCS classifier (note, that the times in Table I are for classifying all images in the testing dataset). The computation times show that the *K*NN classifier is quicker than the proposed BBCS classification approach. However, later results will show that the *K*NN classifier is more vulnerable to the effects of noise than the BBCS approach.

According to these results, the computation times for the BCS and BBCS are the same. However, the accuracy is consistently lower for the BCS classifier as compared to the BBCS classifier. The accuracy of the other two classifiers considered in the comparison also outperforms the BCS-based method. As a result, the BCS-based classifier will not be considered further in this performance evaluation.

Fig. 2 shows 20 images (from the original testing dataset) that the *K*NN algorithm fails to satisfactorily classify. The first row gives the images that are under consideration and the second row gives the classification results from the *K*NN classifier. For comparison, the SRC and BBCS classification results are shown in rows 3 and 4, respectively. The relative performances of the three methods are also further summarized in Table II. Here, it can be seen that both methods outperform the *K*NN algorithm in terms of classification accuracy. The BBCS classifier gives the highest classification accuracy overall. Note that the 30 independent simulation runs are conducted with the final selected class being the most frequent overall.

## B. Classification Comparisons with Noise

In practice, it is unlikely that the logos being classified will be clearly visible. Hence, here different levels of Gaussian white noise are added to the training images and testing images in order to examine the performance

Table II
Accuracies Obtained Using Challenging Data

| Classifier | *K*NN | SRC | BBCS |
|---|---|---|---|
| Accuracy | 19.17% | 43.83% | **47**% |

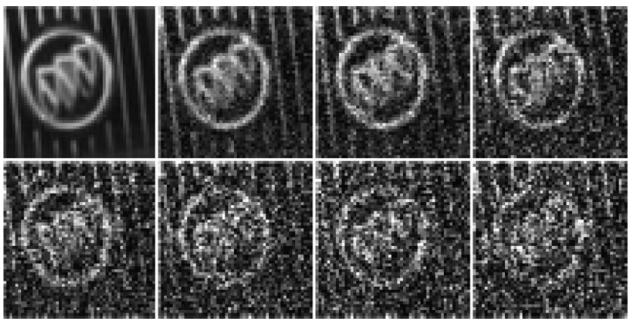


Fig. 3. An example of a training image and the effect by adding Gaussian white noise to image intensities with zero mean and variance values of 0.02, 0.05, 0.1, 0.15, 0.2, 0.25, and 0.3 from left to right, respectively.

of the classifiers. Due to computational costs, only $M = 300$ will be considered in this subsection and those that follow. This has been selected as a compromise between accuracy and computational costs.

Fig. 3 shows an example of a training image and the effects of adding noise with increasing values of variance. The intensities of all pixels in the image are normalized, giving values between 0 and 1. A white Gaussian noise is then added to each pixel, which varies the pixel intensity, with the effects of different variance levels being investigated. Normally, an image is considered highly contaminated if the variance of the Gaussian noise is above 0.2. The noise variance levels in the training and testing images are denoted as $\sigma^2_{\text{train}}$ and $\sigma^2_{\text{test}}$, respectively.

Ten independent classification simulation runs are then carried out using the noisy images and the mean accuracies are shown in Fig. 4. Although adding a small amount of noise to the training images can initially offer an improvement in terms of classification accuracy, there is a degradation in performance when it is is increased further.

According to the authors' experience, there are more SIFT features that could be detected in slightly noisy images. This results in a better image representation vector. It can be explained by the fact that the use of the small amount of noise preserves more edges than for clear images after the Gaussian smoothing process used in the SIFT algorithm. However, an increase of the noise level makes difficult to recover the image. As the noise variance is increased, less and less SIFT features can then be detected as the images are then severely damaged by the noise.

Fig. 4 shows that the *K*NN classifier is the most vulnerable to the effects of noise. It can be explained by the fact that the *K*NN classifier only calculates the Euclidean distance, while the other two allow for some error when modeling a testing image feature as a linear combination of the training image features. The performances of the BBCS classifier and the SRC are similar, while the BBCS classifier tends to be more accurate compared with the SRC when the training images are heavily contaminated by noise. For instance, when the

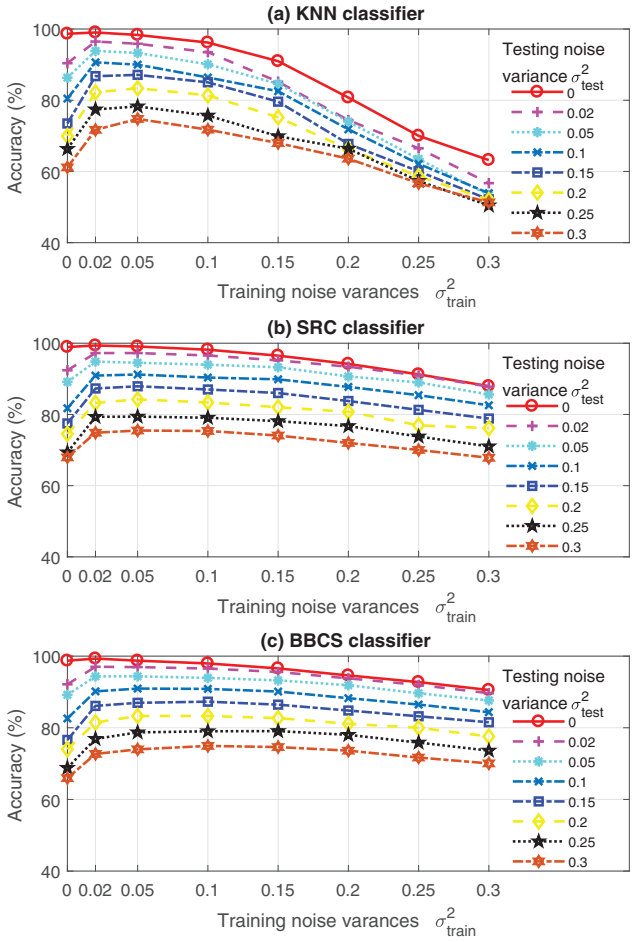



Fig. 4. Noise robustness comparisons for the *K*NN, SRC, and BBCS classifiers.

noise variances are 0.25 in the training and testing images, the BBCS classifier and the SRC achieve 75.87% and 73.79%, respectively. Furthermore, when the noise variances increase to 0.3, the BBCS classifier and the SRC can achieve 70.05% and 67.82%, respectively.

## C. Column-Based Subspace Sampling

In this section, a reduced number of training images are used to evaluate the situation where the size of the dictionary is large. Table III shows the time and computational cost comparisons for different classifiers. Using the column-based subspace sampling method, the partial dictionary size is decreased to 20% and 10% (denoted as $p_1$ and $p_2$, respectively) when compared to the original dataset (denoted as $f$). The computational cost decreases about 6 times ($p_1$) and 11 times ($p_2$), while the accuracy drops slightly. The proposed BBCS approach requires an overall time of 500 and 277 s, respectively, which is 0.3 and 0.18 s per image. The experiments are performed over 1500 images. This could still be applied to real-time applications. Even though the computational cost of the proposed algorithm is still higher than the cost of the KNN algorithm, it is more robust than the *K*NN when applied to noisy images. Since

Table III
Comparisons Between Using the Full and Partial Dictionaries

| Classifiers | $K$**NN**$(f)$ | **SRC**$(f)$ | **BBCS**$(f)$ |
|---|---|---|---|
| Accuracy (%) | $98.63 \pm 0.27$ | $98.78 \pm 0.24$ | $\mathbf{98.86} \pm 0.22$ |
| Misclassified images | 26.33 | 18.30 | **17.10** |
| Time (s) | **2.70** | 8360 | 3120 |
| Classifiers | $K$**NN**$(p_1)$ | **SRC**$(p_1)$ | **BBCS**$(p_1)$ |
| Accuracy (%) | $97.32 \pm 0.47$ | $\mathbf{98.54} \pm 0.31$ | $98.24 \pm 0.35$ |
| Misclassified images | 40.20 | **21.83** | 26.83 |
| Time (s) | **0.25** | 1436 | 500 |
| Classifiers | $K$**NN**$(p_2)$ | **SRC**$(p_2)$ | **BBCS**$(p_2)$ |
| Accuracy (%) | $96.75 \pm 0.86$ | $\mathbf{97.49} \pm 0.61$ | $96.94 \pm 0.52$ |
| Misclassified images | 40.20 | **21.83** | 26.83 |
| Time (s) | **0.13** | 1170 | 277 |

10% data reduction does not decrease the accuracy significantly, the next experiments are performed with 10% data reduction as a trade-off between the computational cost and accuracy.

Fig. 5 shows the result of different classifiers when the dictionary size is decreased to 10% of the original size by the column-based subspace sampling method. When comparing the accuracies to those shown in Fig. 4, the accuracy of each classification method has been reduced

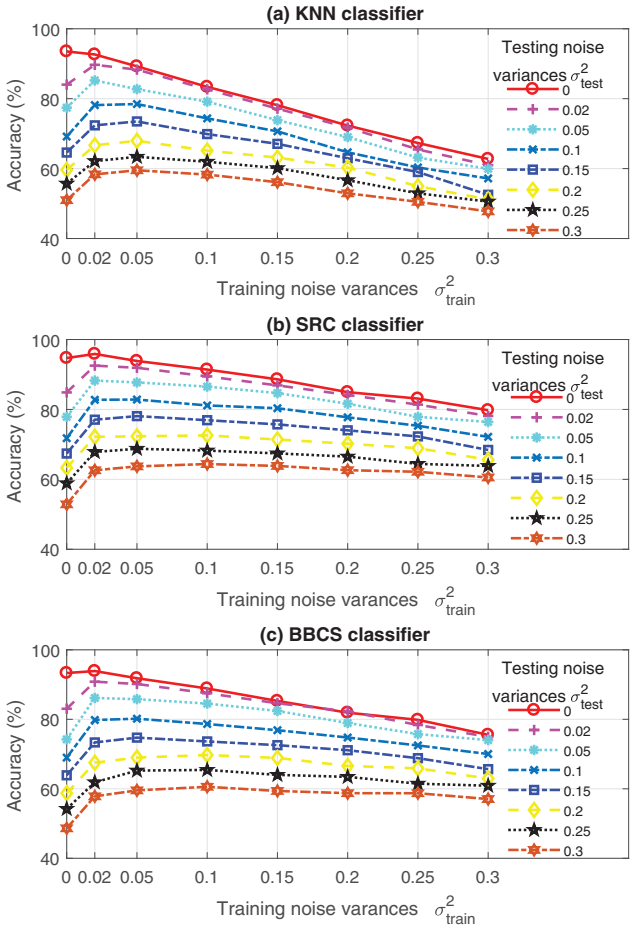



Fig. 5. Noise robustness comparisons when there are 10% training examples in each class using the column-based subspace sampling.

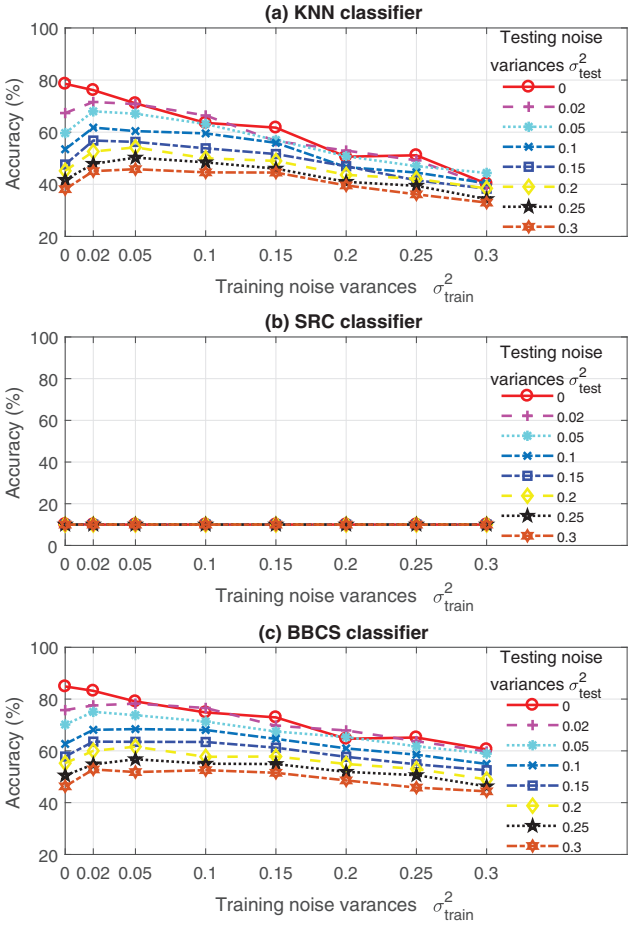



Fig. 6. Noise robustness comparisons when there are 1% training examples in each class using the column-based subspace sampling.

when compared to Fig. 5. Moreover, Fig. 5 shows that the $K$NN classifier is vulnerable to noise and the SRC is only marginally more accurate than the BBCS classifier, despite having previously been shown to be less computationally efficient. However, the computational cost is dropped as the dictionary size has decreased by a factor of 10.

The size of the training dataset is further decreased to only 1% selected images for each class in each of the 10 independent simulations, with the resulting classification accuracies being shown in Fig. 6. In this case, the accuracies of the $K$NN classifier are not as high as those of the BBCS algorithm, especially when the noise levels increase. The SRC does not work any more since $M > N$ and the system is no longer underdetermined. Note that the conventional compressive sensing framework (as used in the SRC) is specifically designed for systems that are under-determined [15]. This leads to a random guess that can only achieve 10% accuracy as there are 10 classes with equal number of logos in each class.

## VI. PERFORMANCE EVALUATION FOR SCENE RECOGNITION

The previous section considered the application of BBCS for VLR. TSR is a very similar topic in smart

cities. Here, the FM2 dataset [45] is considered. This dataset contains 6237 images from eight classes: highway, road, tunnel, tunnel exit, settlement, overpass, toll booth, and dense traffic. Seventy percent of the images are randomly chosen for the training stage and the rest 30% of images are for testing purposes. Fig. 7 illustrates some examples of the FM2 dataset.

A pre-trained CNN framework (AlexNet [28]) is used for feature extraction. Instead of using the original weights from the network that was trained on other images, this work replaces the last fully connected layer to 200 neurons and fine-tunes the weights based on traffic scene images. Hence, each image is represented by a vector of length 200. Note that the focus is on the classification method rather than on the image feature extraction.

The column-based subspace sampling representation is applied to each training group. Since each class has imbalanced training data, the experiment sets a maximum number of 200 to each class. When a class has more than 200 training images, the column-based subspace sampling method is applied to this class. A comparison with a recently developed deep learning approach, the CNN from [28], is performed, where the weights are trained for classification. Note that in CNN the classification is applied directly without using column-based subspace sampling. Since the parameters are fixed based on the whole training dataset, there is no need of retraining a network using a much smaller dataset. However, the results for *K*NN, BBCS, and SRC are achieved on the new dataset after the column-based subsampling.

Table IV shows the result from each classifier. Zero-mean Gaussian noises with different noise variances are applied on these training images and testing images. Without adding any noise, the CNN achieves the highest accuracy. However, when increasing the noise, the CNN becomes fragile. Similar research shows that when changing the intensity of even a single pixel, the classification result changes [46]. However, using the extracted features from CNN and applying them to other classifiers leads to better results. Increasing the noise level, the proposed BBCS achieves the best results. This is important as the real images are not always clear. Fig. 8 illustrates how different noise levels influence an image.

Table IV
Classifiers' Accuracy Comparisons Using Features Extracted by CNN Based on the FM2 Dataset

| Noise variance | **CNN** (%) | *K***NN** (%) | **SRC** (%) | **BBCS** (%) |
|---|---|---|---|---|
| 0 | **87.70** | 84.41 | 87.00 | 86.31 |
| 0.01 | 57.01 | 73.21 | 79.73 | **79.89** |
| 0.1 | 10.59 | 56.04 | 57.59 | **64.39** |
| 0.2 | 7.43 | 52.03 | 42.51 | **54.33** |

## VII. APPLICATION OF BBCS TO ALTERNATIVE DATASET

The proposed BBCS approach has the potential to be applied to other areas, not only to VLR and TSR. In this performance validation, the CIFAR-10 dataset [47] is used. This dataset consists of 50,000 training images and 10,000 testing images. Here, a CNN similar to [28] is trained on the new dataset. The network contains three convolution layers with 48, 96, and 192 3-by-3 kernels. Each convolutional layer is followed by a batch normalization layer and a max-pooling layer. Two fully connected layers are followed with 512 and 200 neurons, respectively. The ReLU non-linear function [28] is applied to all neurons, except the softmax being applied to the neurons in the last layer. The last fully connected layer is used as the feature. Hence, each image is represented by a vector of length 200.

The column-based subspace sampling is applied to each training group. This process picks 200 image feature vectors from 5000 image feature vectors in each group (4% of the original size). Hence, in order to avoid using all image feature vectors, the dictionary $\mathbf{X}$ is formed by only 2000 representative image feature vectors. Both the

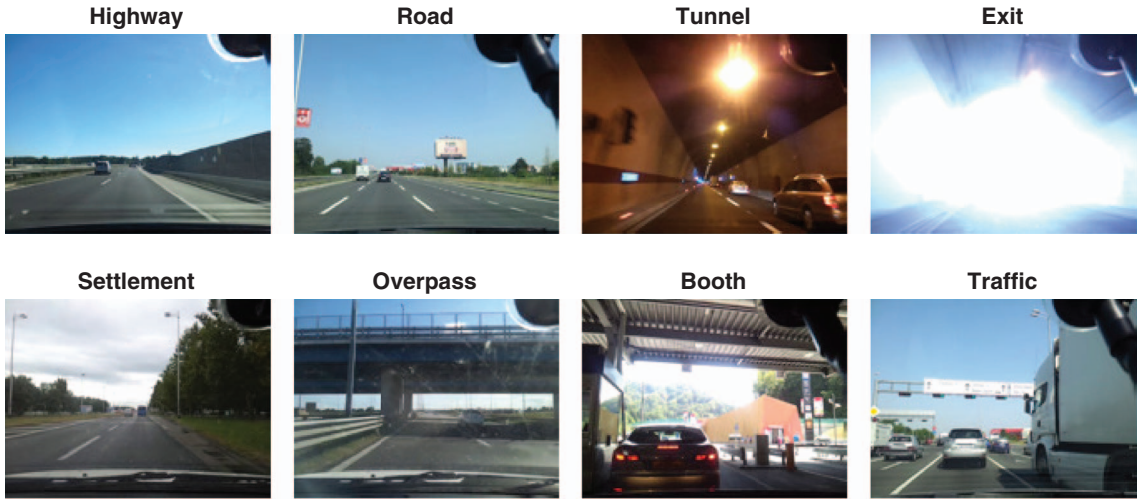



Fig. 7. Example of classes from the FM2 dataset.

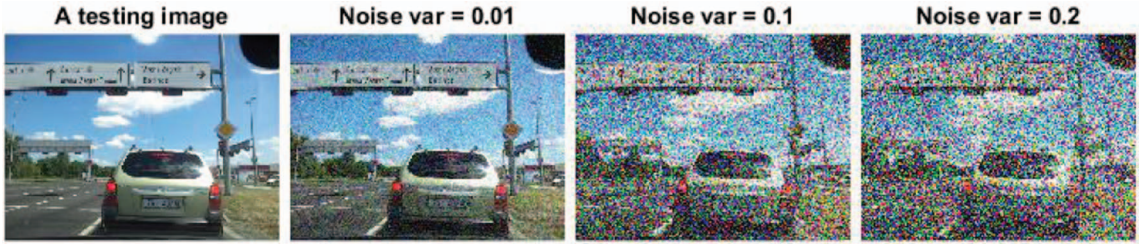



Fig. 8. An example of a traffic scene image with different levels of noises.

Table V
Classifiers' Accuracy Comparisons Using Features Extracted by CNN Based on the CIFAR-10 Dataset

| Noise variance | **CNN** (%) | *K***NN** (%) | **SRC** (%) | **BBCS** (%) |
|---|---|---|---|---|
| 0 | **81.87** | 68.79 | 78.53 | 73.40 |
| 0.01 | 47.60 | 52.77 | 52.51 | **58.36** |
| 0.02 | 36.37 | 42.39 | 43.80 | **46.98** |

CNN and BBCS approaches train the weights for classification. Similarly, in CNN the classification is applied directly without using column-based subspace sampling.

Table V gives the performance of each classifier. Zero-mean Gaussian noises with different noise variances are added on these training images and testing images. Note that here the noise level is lower than that in the VLR dataset. The reason for this is the images in the CIFAR-10 dataset are tiny color images. A small color image can be easily contaminated by adding up the noise effects from each channel. Fig. 9 illustrates the effect of the noise contamination. Similar to the TSR dataset, the result shows that the CNN classifier is not robust to noise. However, using the features extracted by the CNN and applying it to other classifiers could achieve better accuracy. This is important as clear images are not always guaranteed in real applications. Table V also shows that SRC should achieve good accuracy when the images are noise free, even if only 4% training images are applied. However, when the images are noisy, the BBCS algorithm achieves the best accuracy. Again, both BBCS and SRC perform better than the *K*NN algorithm.

## VIII. CONCLUSION

This paper proposes a novel nonparametric classification approach, namely the BBCS classifier. The novelty of the work has two main components: 1) the proposed back propagation process, and 2) the proposed column-based subspace sampling to reduce the size of the dataset and associated computation costs.

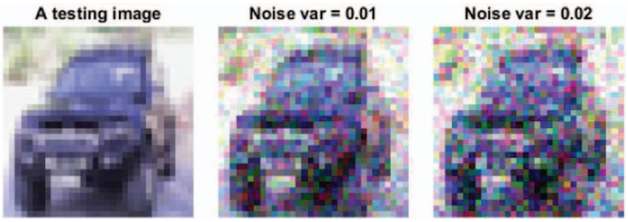



Fig. 9. An example of an image from the CIFAR 10 dataset with different levels of noises.

The developed approach relies on the constructing of the testing image feature using partial information from the weights estimated by BCS. Note, that for each class there is a corresponding reconstructed image feature. By comparing the reconstructed image feature with the testing image feature, the objects of interest are reconstructed and classified.

The proposed backpropagation process gives a significant reduction of the misclassification error. For VLR, the number of misclassified testing images reduces by 87% when compared with the BCS classifier. Compared with the SRC, the BBCS algorithm gives a similar recognition accuracy, while decreasing the mean computational cost by 68%. However, the SRC does not work when the training dataset is small while the BBCS algorithm shows accurate results in the same situation. Moreover, the proposed classifier and column-based subspace sampling have been shown to be robust to the effects of heavy noise, unlike the *K*NN classifier. The proposed approach is a general nonparametric classifier and is also validated on the TSR dataset and on the CIFAR-10 image dataset.

## APPENDIX A
## MARGINAL LIKELIHOOD MAXIMIZATION

The following gives a detailed derivation for the marginal likelihood in equation (17). By combining equations (6) and (7), the marginal likelihood can be expanded to

$$\begin{aligned} p(\mathbf{x}^*|\boldsymbol{\alpha},\sigma^2) &= \int p(\mathbf{x}^*|\mathbf{w},\sigma^2)\, p(\mathbf{w}|\boldsymbol{\alpha}) d\mathbf{w} \\ &= (2\pi\sigma^2)^{-M/2}(2\pi)^{-N/2}|\mathbf{A}|^{1/2} \\ &\times \int \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{x}^* - \mathbf{X}\mathbf{w}||_2^2 + \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w}\right\} \mathrm{d}\mathbf{w}. \end{aligned} \tag{A1}$$

In order to simplify equation (A1), define

$$\mathbb{Q} = \frac{1}{2}\left\{\frac{1}{\sigma^2}||\mathbf{x}^* - \mathbf{X}\mathbf{w}||_2^2 + \mathbf{w}^{\mathrm{T}}\mathbf{A}\mathbf{w}\right\}. \tag{A2}$$

Combining with equations (13) and (14), equation (A2) can be given as

$$\mathbb{Q} = \frac{1}{2}\left(\frac{\mathbf{x}^{*\mathrm{T}}\mathbf{x}^{*}}{\sigma^2} - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right) + \frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{w}-\boldsymbol{\mu}). \tag{A3}$$

In order to simplify equation (A3), we set

$$\mathbb{T} = \frac{1}{2}\left(\frac{\mathbf{x}^{*\mathrm{T}}\mathbf{x}^{*}}{\sigma^2} - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right). \tag{A4}$$

Therefore, the integral part on the right-hand side of equation (A1) is given by

$$\int \exp\{-\mathbb{Q}\}d\mathbf{w} = (2\pi)^{N/2}|\boldsymbol{\Sigma}|^{1/2}\exp\{-\mathbb{T}\}. \tag{A5}$$

Substituting equation (A5) back in equation (A1) gives

$$p(\mathbf{x}^{*}|\boldsymbol{\alpha},\sigma^2) = (2\pi\sigma^2)^{-M/2}|\mathbf{A}|^{1/2}|\boldsymbol{\Sigma}|^{1/2}\exp\{-\mathbb{T}\}. \tag{A6}$$

This can be further simplified by

$$p(\mathbf{x}^{*}|\boldsymbol{\alpha},\sigma^2) = (2\pi)^{-M/2}\frac{1}{\sigma^M}\frac{1}{|\mathbf{I}_N + \sigma^{-2}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}|^{1/2}}\exp\{-\mathbb{T}\}, \tag{A7}$$

where $\mathbf{I}_N = \mathbf{A}^{-1}\mathbf{A}$. Using the matrix determinant properties [48] that $|\mathbf{I}_N + \mathbf{D}^{\mathrm{T}}\mathbf{B}| = |\mathbf{I}_M + \mathbf{D}\mathbf{B}^{\mathrm{T}}|$ with $\mathbf{D} \in \mathbb{R}^{M\times N}$ and $\mathbf{B} \in \mathbb{R}^{M\times N}$, the above equation can be updated to

$$p(\mathbf{x}^{*}|\boldsymbol{\alpha},\sigma^2) = (2\pi)^{-M/2}\frac{1}{|\sigma^2\mathbf{I}_M + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}}|^{1/2}}\exp\{-\mathbb{T}\}. \tag{A8}$$

Recall that $\mathbb{T}$ is given in equation (A4) and it can be expressed as follows:

$$\mathbb{T} = \frac{1}{2}\left(\mathbf{x}^{*\mathrm{T}}\left[\sigma^{-2}\mathbf{I}_M - \sigma^{-2}\mathbf{X}(\mathbf{A}+\sigma^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\sigma^{-2}\right]\mathbf{x}^{*}\right). \tag{A9}$$

According the Woodbury inversion identity [34]

$$[\sigma^{-2}\mathbf{I}_M - \sigma^{-2}\mathbf{X}(\mathbf{A}+\sigma^{-2}\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\sigma^{-2}] = (\sigma^2\mathbf{I}_M + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}})^{-1}, \tag{A10}$$

equation (A9) can be expressed as

$$\mathbb{T} = \frac{1}{2}\left(\mathbf{x}^{*\mathrm{T}}(\sigma^2\mathbf{I}_M + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}})^{-1}\mathbf{x}^{*}\right). \tag{A11}$$

Therefore, equation (A8) can be given as

$$p(\mathbf{x}^{*}|\boldsymbol{\alpha},\sigma^2) = \frac{1}{\sqrt{(2\pi)^M|\mathbf{C}|}}\exp\left\{-\frac{1}{2}\mathbf{x}^{*T}\mathbf{C}^{-1}\mathbf{x}^{*}\right\}, \tag{A12}$$

which links back to equation (18), with the $M \times M$ matrix $\mathbf{C}$ given by

$$\mathbf{C} = \sigma^2\mathbf{I}_M + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^{\mathrm{T}}. \tag{A13}$$

## APPENDIX B
## EVIDENCE APPROXIMATION

This section presents the derivation of the marginal log-likelihood function and its maximization with respect to $\alpha_i$ and $\sigma^2$. We can express $\mathbb{T}$ from equation (A4) as follows:

$$\mathbb{T} = \frac{1}{2\sigma^2}||\mathbf{x}^{*} - \mathbf{X}\boldsymbol{\mu})||_2^2 + \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\mathbf{A}\boldsymbol{\mu}. \tag{B1}$$

Hence, taking the logarithm of the marginal likelihood given in equation (A6), the logarithm of the marginal likelihood can be obtained in the following form:

$$\mathcal{L}(\boldsymbol{\alpha},\sigma^2) = -\frac{M}{2}\ln\sigma^2 - \frac{M}{2}\ln(2\pi) + \frac{1}{2}\sum_{i=1}^{N}\ln\alpha_i + \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2\sigma^2}||\mathbf{x}^{*} - \mathbf{X}\boldsymbol{\mu})||_2^2 - \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\mathbf{A}\boldsymbol{\mu}. \tag{B2}$$

The procedure of maximizing equation (B2) with respect to $\alpha_i$ and $\sigma^2$ is known as the evidence approximation procedure.

Following the approach from [49], the derivative of $\ln|\boldsymbol{\Sigma}|$ with respect to $\alpha_i$ is

$$\frac{d}{d\alpha_i}\ln|\boldsymbol{\Sigma}| = \frac{d}{d\alpha_i} - \ln|\boldsymbol{\Sigma}|^{-1} = -\mathrm{Trace}\boldsymbol{\Sigma} = -\Sigma_{ii}, \tag{B3}$$

where $\Sigma_{ii}$ is the $i$th diagonal component of the posterior covariance matrix $\boldsymbol{\Sigma}$ and Trace is the trace of a matrix. Therefore, the derivative of $\mathcal{L}(\boldsymbol{\alpha},\sigma^2)$ from equation (B2) with respect to $\alpha_i$ is

$$\frac{d\mathcal{L}(\boldsymbol{\alpha},\sigma^2)}{d\alpha_i} = \frac{1}{2\alpha_i} - \frac{1}{2}\Sigma_{ii} - \frac{1}{2}\mu_i^2. \tag{B4}$$

Setting the derivative to zero gives equation (20).

In order to simplify $d\mathcal{L}(\boldsymbol{\alpha},\sigma^2)/d\sigma^2$, set $\beta = 1/\sigma^2$. Following the approach from [50], the derivative of $\ln|\boldsymbol{\Sigma}|$ with respect to $\beta$ is

$$\frac{d}{d\beta}\ln|\boldsymbol{\Sigma}| = \frac{d}{d\beta} - \ln|\boldsymbol{\Sigma}|^{-1} = -\mathrm{Trace}(\mathbf{I}_N - \boldsymbol{\Sigma}\mathbf{A})\beta^{-1}. \tag{B5}$$

Therefore, the derivative of $\mathcal{L}(\boldsymbol{\alpha},\sigma^2)$ from equation (B2) with respect to $\beta$ is

$$\frac{d\mathcal{L}(\boldsymbol{\alpha},\sigma^2)}{d\beta} = \frac{M}{2\beta} - \frac{1}{2}||\mathbf{x}^{*} - \mathbf{X}\boldsymbol{\mu})||_2^2 - \frac{1}{2}\mathrm{Trace}(\mathbf{I}_N - \boldsymbol{\Sigma}\mathbf{A})\beta^{-1}. \tag{B6}$$

Setting the derivative to zero gives equation (21).

## REFERENCES


[1] Y. Ou, H. Zheng, S. Chen, and J. Chen "Vehicle logo recognition based on a weighted spatial pyramid framework," in *Proc. 17th IEEE Int. Conf. Intell. Transp. Syst.*, Qingdao, China, Oct. 2014, pp. 1238–1244.

[2] Q. Sun, X. Lu, L. Chen, and H. Hu
"An improved vehicle logo recognition method for road surveillance images,"
in *Proc. 7th Int. Symp. Comput. Intell. Des.*, Hangzhou, China, Dec. 2014, pp. 373–376.

[3] D. Llorca, R. Arroyo, and M. Sotelo
"Vehicle logo recognition in traffic images using HOG features and SVM,"
in *Proc. 16th IEEE Int. Conf. Intell. Transp. Syst.*, The Hague, Netherlands, Oct. 2013, pp. 2229–2234.

[4] I. Sikirić, K. Brkić, J. Krapac, and S. Šegvić
"Robust traffic scene recognition with a limited descriptor length,"
in *Proc. CVPR Workshop Vis. Place Recognit. Changing Environ.*, Boston, MA, USA, Jun. 2015.

[5] R. Chen, M. Hawes, L. Mihaylova, J. Xiao, and W. Liu
"Vehicle logo recognition by spatial-SIFT combined with logistic regression,"
in *Proc. IEEE Int. Conf. Inf. Fusion*, Heidelberg, Germany, Jul. 2016, pp. 1228–1235.

[6] Y. Huang, R. Wu, Y. Sun, W. Wang, and X. Ding
"Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy,"
*IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1951–1960, 2015.

[7] R. Chen, M. A. Jalal, L. Mihaylova, and R. K. Moore
"Learning capsules for vehicle logo recognition,"
in *Proc. 21st Int. Conf. Inf. Fusion*, Jul. 2018, pp. 565–572.

[8] J. S. Sánchez, F. Pla, and F. J. Ferri
"On the use of neighbourhood-based non-parametric classifiers,"
*Pattern Recognit. Lett.*, vol. 18, no. 11, pp. 1179–1186, 1997.

[9] C. M. Bishop
*Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[10] K. P. Murphy
*Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[11] A. P. Psyllos, C.-N. E. Anagnostopoulos, and E. Kayafas
"Vehicle logo recognition using a SIFT-based enhanced matching scheme,"
*IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 322–328, 2010.

[12] H. Peng, X. Wang, H. Wang, and W. Yang
"Recognition of low-resolution logos in vehicle images based on statistical random sparse distribution,"
*IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 681–691, 2015.

[13] A. Hinneburg, C. C. Aggarwal, and D. A. Keim
, "What is the nearest neighbor in high dimensional spaces?"
in *Proc. Int. Conf. Very Large Data Bases*, San Francisco, CA, USA, Sept. 2000, pp. 506–515.

[14] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma
"Robust face recognition via sparse representation,"
*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009.

[15] D. L. Donoho
"For most large underdetermined systems of linear equations the minimal $l_1$-norm solution is also the sparsest solution,"
*Commun. Pure Appl. Math.*, vol. 59, no. 6, pp. 797–829, 2006.

[16] S. Ji, Y. Xue, and L. Carin
"Bayesian compressive sensing,"
*IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, 2008.

[17] X. Zhang, J. Qin, and G. Li
"SAR target classification using Bayesian compressive sensing with scattering centers features,"
*Prog. Electromagn. Res.*, vol. 136, pp. 385–407, 2013.

[18] Y. Zhang, Y. Li, Z. Wang, Z. Song, R. Lin, J. Qian, and J. Yao
"A fast image reconstruction method based on Bayesian compressed sensing for the undersampled AFM data with noise,"
*Meas. Sci. Technol.*, vol. 30, no. 2, p. 025402, Jan. 2019.

[19] Y. Huang, J. Paisley, Q. Lin, X. Ding, X. Fu, and X. Zhang
"Bayesian nonparametric dictionary learning for compressed sensing MRI,"
*IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5007–5019, Dec. 2014.

[20] T. N. Sainath, A. Carmi, D. Kanevsky, and B. Ramabhadran
"Bayesian compressive sensing for phonetic classification,"
in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Dallas, TX, USA, Mar. 2010, pp. 4370–4373.

[21] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf
"Learning with local and global consistency,"
in *Proc. Adv. Neural Inf. Process. Syst.*, Whistler, BC, Canada, Dec. 2003, pp. 321–328.

[22] M. Shi, T. Furon, and H. Jégou
"A group testing framework for similarity search in high-dimensional spaces,"
in *Proc. ACM Int. Conf. Multimedia*, New York, NY, USA, Nov. 2014, pp. 407–416.

[23] A. Iscen, M. Rabbat, and T. Furon
"Efficient large-scale similarity search using matrix factorization,"
in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 2073–2081.

[24] M. W. Mahoney and P. Drineas
"CUR matrix decompositions for improved data analysis,"
*Proc. Natl Acad. Sci. USA*, vol. 106, no. 3, pp. 697–702, 2009.

[25] E. Elhamifar, G. Sapiro, and R. Vidal
"See all by looking at a few: Sparse modeling for finding representative objects,"
in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 1600–1607.

[26] C. Boutsidis, M. W. Mahoney, and P. Drineas
"An improved approximation algorithm for the column subset selection problem,"
in *Proc. Annu. ACMSIAM Symp. Discrete Algorithms*, New York, NY, USA, Jan. 2009, pp. 968–977.

[27] D. G. Lowe
"Distinctive image features from scale-invariant keypoints,"
*Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton
"ImageNet classification with deep convolutional neural networks,"
in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1097–1105.

[29] S. A. Tesfamicael and F. Barzideh
"Bayesian inference and compressed sensing,"
*Bayesian Inference*, J. P. Tejedor, Ed. London, U.K.: IntechOpen, Nov. 2017.

[30] E. Amaldi and V. Kann
"On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems,"
*Theor. Comput. Sci.*, vol. 209, no. 1, pp. 237–260, 1998.

[31] E. J. Candès, J. Romberg, and T. Tao
"Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,"
*IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[32] S. S. Chen, D. L. Donoho, and M. A. Saunders
"Atomic decomposition by basis pursuit,"
*SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[33] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck
"Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit,"
*IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[34] M. E. Tipping
"Sparse Bayesian learning and the relevance vector machine,"
*J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[35] L. Figueiredo, I. Jesus, J. T. Machado, J. Ferreira, and J. M. De Carvalho
"Towards the development of intelligent transportation systems,"
in *Proc. IEEE Intell. Transp. Syst.*, Oakland, CA, USA, Aug. 2001, pp. 1206–1211.

[36] Z. Zhang, X. Wang, W. Anwar, and Z. L. Jiang
"A comparison of moments-based logo recognition methods,"
in *Proc. Abstr. Appl. Anal.*, vol. 2014, 2014, pp. 1–8.

[37] C. Y. Chen, W. Choi, and M. Chandraker
"Atomic scenes for scalable traffic scene recognition in monocular videos,"
in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Lake Placid, NY, USA, Mar. 2016, pp. 1–9.

[38] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. J. Russell, and J. Weber
"Automatic symbolic traffic scene analysis using belief networks,"
*Adv. Artif. Intell.*, vol. 94, 1994, pp. 966–972.

[39] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray
"Visual categorization with bags of keypoints,"
in *Proc. Workshop Statist. Learn. Comput. Vis.*, Prague, Czech Republic, May 2004, pp. 1–22.

[40] X. Zhen and L. Shao
"Action recognition via spatio-temporal local features: A comprehensive study,"
*Image Vis. Comput.*, vol. 50, pp. 1–13, 2016.

[41] U. L. Altintakan and A. Yazici
"Towards effective image classification using class-specific codebooks and distinctive local features,"
*IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 323–332, 2015.

[42] A. Vedaldi and B. Fulkerson
"VLFeat: An open and portable library of computer vision algorithms,"
in *Proc. 18th ACM Conf Int. Multimedia*, New York, NY, USA, Oct. 2010, pp. 1469–1472.

[43] M. Grant and S. Boyd
"CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[44] M. C. Grant and S. P. Boyd
"Graph implementations for nonsmooth convex programs,"
in *Recent Advances in Learning and Control*. Vincent D. Blondel, Stephen P. Boyd, and Hidenori Kimura, Eds. London, U.K.: Springer, 2008, pp. 95–110.

[45] I. Sikirić, K. Brkić, J. Krapac, and S. Šegvić
"Image representations on a budget: Traffic scene classification in a restricted bandwidth scenario,"
in *Proc. IEEE Intell. Veh. Symp.*, Dearborn, MI, USA, Jun. 2014, pp. 845–852.

[46] J. Su, D. V. Vargas, and K. Sakurai
"One pixel attack for fooling deep neural networks,"
*IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[47] A. Krizhevsky and G. Hinton
"Learning multiple layers of features from tiny images,"
Masters thesis, Dept. Computer Science, Univ. Toronto, 2009.

[48] M. Brookes
"The matrix reference manual," Imperial College London, London, U.K., 2005. [Online]. Available: http://www.ee.imperial.ac.uk/hp/staff/dmb/matrix/intro.html.

[49] D. J. MacKay
"Bayesian interpolation,"
*Neural Comput.*, vol. 4, no. 3, pp. 415–447, 1992.

[50] T. Fletcher
"Relevance vector machines explained," University College London, London, U.K., 2010. [Online]. Available: http://home.mit.bme.hu/horvath/IDA/RVM.pdf.

**Ruilong Chen** received the M.Sc. degree from the Department of Electronic and Electrical Engineering at the University of Sheffield, Sheffield, U.K., and the Ph.D. degree from the Department of Automatic Control and Systems Engineering, University of Sheffield, in 2013 and 2018, respectively. His research interests include machine learning, image processing, deep neural networks, image recognition, and object detection.

**Matthew Hawes** received the M.Eng. and Ph.D. degrees from the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K., in 2010 and 2014, respectively. Between 2014 and 2017, he was a Research Associate with the Department of Automatic Control and Systems Engineering, University of Sheffield, working on the EU-funded SETA project and the EPSRC-funded BTaRoT project. His research interests include array signal processing, machine learning, big data, modeling complex systems, data fusion, sequential Monte Carlo methods, and Markov chain Monte Carlo methods.

**Lyudmila Mihaylova** received the M.Eng., M.Sc., and Ph.D. degrees, all awarded from the Technical University of Sofia, Bulgaria. She is a Professor of Signal Processing and Control at the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K. Her research interests are in the areas of machine learning and autonomous systems with applications to navigation, surveillance, and sensor network systems. She has given a number of invited talks and tutorials, including the plenary talks for the JIC Smart Cities (Cairo, Egypt, 2019), NATO SET-262 AI 2018 (Hungary), Fusion 2017, and others. Prof. Mihaylova is an Associate Editor for the IEEE Transactions on Aerospace and Electronic Systems and Elsevier's *Signal Processing* journal. She was the President of the International Society of Information Fusion (ISIF) for two mandates: 20172018. She is on the Board of Directors of ISIF and a Senior IEEE member.

# INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

## 2020 BOARD OF DIRECTORS*

| 2018–2020 | 2019–2021 | 2020–2022 |
|---|---|---|
| Fredrik Gustafsson | Kathryn Laskey | Pieter De Villiers |
| X. Rong Li | Felix Govaers | Murat Efe |
| Zhansheng Duan | Simon Maskell | Wolfgang Koch |

*Board of Directors are elected by the members of ISIF for a three year term.

## PAST PRESIDENTS

| | | |
|---|---|---|
| Paulo Costa, 2019 | Joachim Biermann, 2011 | Xiao-Rong Li, 2003 |
| Lyudmila Mihaylova, 2018 | Stefano Coraluppi, 2010 | Yaakov Bar-Shalom, 2002 |
| Lyudmila Mihaylova, 2017 | Elisa Shahbazian, 2009 | Pramod Varshney, 2001 |
| Jean Dezert, 2016 | Darko Musicki, 2008 | Yaakov Bar-Shalom, 2000 |
| Darin Dunham, 2015 | Erik Blasch, 2007 | Jim Llinas, 1999 |
| Darin Dunham, 2014 | Pierre Valin, 2006 | Jim Llinas, 1998 |
| Wolfgang Koch, 2013 | W. Dale Blair, 2005 | |
| Roy Streit, 2012 | Chee Chong, 2004 | |

## SOCIETY VISION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied information fusion technologies.

## SOCIETY MISSION

**Advocate**

To advance the profession of fusion technologies, propose approaches for solving real-world problems, recognize emerging technologies, and foster the transfer of information.

**Serve**

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

**Communicate**

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

**Educate**

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

**Integrate**

Integrate ideas from various approaches for information fusion, and look for common threads and themes– look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

**Disseminate**

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

## Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors are encouraged to submit their manuscripts for peer review http://isif.org/journal.

## Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at http://jaif.msubmit.net. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.

Journal of Advances in Information Fusion Volume 15 Number 1 June 2020 ISSN 1557-6418