2017

MULTI-OBJECT DETECTION AND TRACKING IN SENSOR NETWORKS AND VIDEO SEQUENCES

Guohua Ren

MULTI-OBJECT DETECTION AND TRACKING IN SENSOR NETWORKS AND VIDEO SEQUENCES

by

GUOHUA REN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2017

ACKNOWLEDGEMENTS

I would like to express special appreciation and thanks to my advisor Professor Dr. Ioannis D. Schizas: You have been a great mentor to me. I'm grateful to you for continually guiding me and letting me working as a research assistant throughout my doctoral level study. Thank you for putting the effort and time on evaluating this dissertation. Your advice on both my research work and career has been priceless. I wish someday I could be as enthusiastic, energetic a researcher as you are.

I would also like to thank my committee members, Dr. Jonathan Bredow, Dr. Michael T. Manry, Dr. Qilian Liang and Dr. Saibun Tjuatja. Thank you for taking your time to serve as my committee members.

I am also grateful to my friends, Jia, Akshay, Kazi, Joy, Shaoshu, etc. I've learnt a lot of things from the talks and discussions I had with you.

A special thanks for my parents and my brother. My parents have been working so hard to provide unconditional love and support to my brother and me. My brother has been a best friend to me, thanks for your understanding and advice.

May 3, 2017

ABSTRACT

MULTI-OBJECT DETECTION AND TRACKING
IN SENSOR NETWORKS AND VIDEO SEQUENCES

Guohua Ren, Ph.D.

The University of Texas at Arlington, 2017

Supervising Professor: Ioannis D. Schizas

Firstly the problem of tracking multiple objects using observations acquired at spatially scattered sensors is considered here. Sensors are measuring a sort of signal attenuation from the present targets/sources. Multiple moving targets may die and be born at some point of the monitored period while the states of the sources, e.g. temperature of a fire source, CO/CO2 density of gas source, are also changing with time. In the case of targets tracking, radar signals sent out by sensors and later bounced back from the surface of the targets are measured at sensors, and the task is to find out the true position/velocity information hidden in the sensor measurements. While in the scenario where sources are present in the sensed field, the aforementioned signal attenuation is generally not available, so the task is to estimate the states of the corresponding sources and in the meanwhile recovering the unknown sensing observation matrix.

Concretely, in this thesis a framework is put forth where norm-one regularized factorization is employed to decompose the sensor observation data covariance matrix into sparse factors whose support facilitates recovery of sensors that acquire informative measurements about the targets. This novel sensors-to-targets association scheme is integrated with par-

ticle filtering mechanisms to perform accurate tracking. Precisely, distributed optimization techniques are employed to associate targets with sensors, and Kalman/particle filtering is integrated to perform target tracking using only the sensors selected by the sparse decomposition scheme. Different from existing alternatives, the novel algorithm can efficiently track and associate targets with sensors even in noisy settings.

As for the multi-source tracking scenario, two different sensing architectures are studied: i) A fusion-center based topology where sensors have a limited power budget; and ii) an ad hoc architecture where sensors collaborate with neighboring nodes enabling in-network processing. A novel source-to-sensor association scheme and tracking is introduced by enhancing the standard Kalman filtering minimization formulation with norm-one regularization terms. In the fusion-based topology a pertinent transmission power constraint is introduced, while coordinate descent techniques are employed to recover the unknown sparse observation matrix, select pertinent sensors and subsequently track the source states. In the ad hoc topology, the centralized minimization problem is written in a separable way and the alternating direction method of multipliers is utilized to construct an in-network algorithmic tracking and association framework.

The problem of distributed tracking of multiple targets is tackled by exploiting sensor mobility and the presence of sparsity in the sensor data covariance matrix. Sparse matrix decomposition relying on norm-one/two regularization is integrated with a kinematic framework to identify informative sensors, associate them with the targets and enable them to follow closely the moving targets. Coordinate descent techniques are employed to determine in a distributed way the target-informative sensors, while the modified barrier method is employed to minimize proper error covariance matrices acquired by extended Kalman filtering. Different from existing approaches which force all sensors to move, here local updating recursive rules are obtained only for the target-informative sensors that can update their location and follow closely the corresponding targets while staying connected.

Lastly, we extend out tracking scheme to tackle the problem of tracking multiple objects in a sequence of frames (video). The task of identifying objects is formulated as the process of factorizing a properly defined kernel covariance matrix into sparse factors. The support of these factors will point to the indices of the pixels that form each object. A coordinate descent approach is utilized to determine the sparse factors, and extract the object pixels. A centroid pixel is estimated for each object which is subsequently tracked via Kalman filtering. A novel interplay between the sparse kernel covariance factorization scheme along with Kalman filtering is proposed to enable joint object detection and tracking, while a divide and conquer strategy is put forth to reduce computational complexity and enable real-time tracking. Extensive numerical tests on both synthetic data and thermal video sequences demonstrate the effectiveness of the novel approach and superior tracking performance compared to existing alternatives.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

In recent years, there has been increasing interest on sensor network (SN) systems due to the expanding needs on civilian and military surveillance and monitoring, e.g., air pollution level tracking, forrest fire monitoring, operation of unmanned aerial vehicles (UAVs) for reconnaissance and national defense systems. The availability of cheaper, smaller and intelligent sensor nodes make SN even more popular. These sensors are equipped with wireless interfaces e.g., signal emitter and/or receiver, thus they can communicate with each other and cooperate on measuring and monitoring fields or parameters of interest. We mainly focus on two important applications of SN, multiple target tracking and environmental sources monitoring here.

Object detection and tracking in videos is another active research area in both military and civilian surveillance systems. Different camera sensors provide different data modes to utilize. Recently, un-cooled thermal sensors have become affordable, as a result, there is an increasing interest in utilizing thermal sensors to facilitate video monitoring tasks. In this work, we aim at proposing novel methodologies to process thermal videos, thus precisely identifying and localizing moving objects present in the acquired video sequence.

1.1 Goals of the thesis

1.1.1 Distributed Spatio-Temporal Association and Tracking of Multiple Targets Using Multiple Sensors

Sensor networks allow the collection and distributed processing of information in challenging environments whose structure is not known and is dynamically changing with time, e.g. battlefields. In such harsh environments both equipment and infrastructure, as well as humans, are prone to threats that may be generated due to malicious attacks, functional failures and even human errors. Threats can be quite unpredictable both spatially and temporally, since they could happen anywhere anytime within a setting that consists of heterogeneous units, e.g,. communication units, sensing units and humans. Effective and fast target detection and tracking is really essential to avoid any potential negative effects.

A necessary step towards multi-target tracking is to associate sensors with targets across space and time. Targets present in the sensed field affect only a small portion of the deployed sensor networks (SNs). Thus, given the limited resources, it is pertinent to identify the sensors that acquire informative observations about the targets and use only those which provide this information. We characterize such sensors as 'target-informative' sensors in this manuscript. Many existing tracking techniques require all sensors to be active [1,26,82,84,133] which may be resource-consuming given the locality of the targets and the fact that only a few sensors bear information about the field targets. To this end, a decentralized algorithmic framework is developed here that does not require a central fusion center and it can associate sensors with targets combined with tracking.

An algorithmic framework is proposed here that associates targets with the sensors which acquire informative measurements about these targets, and subsequently performs tracking using only these informative sensors. Note that existing data association schemes [31,42,52,79,83,118] match measurements with targets across time and rely on

probabilistic models. Differently, the sensors-targets association task here is relying only on the acquired sensor data and no probabilistic models are adopted. Specifically, sensors which are positioned close to the same target, acquire data measurements that tend to be correlated, *no matter* what the underlying physical model is. Such correlations induce a sparse structure (presence of many zeros) in the sensor data covariance matrix. Sparsity is an attribute found in many natural and man-made signals, and it has been exploited in a wide range of applications including sparse regression, sub-Nyquist sampling and statistical inference, e.g., see [25, 114].

To facilitate association of sensor measurements with targets a pertinent framework is derived to analyze the sensor data covariance into sparse factors whose support (position of the nonzero entries) will indicate subsets of sensors sensing the same target. Different from [43], [51], [68], [69], [115], [137], the matrix factorization scheme developed here does not require a central fusion center and does not impose structural requirements to the unknown factors such as orthogonality and/or positivity of the factor entries. The idea of covariance sparse factorization was also discussed in [101]. However, the work in [101] is dealing with stationary settings where the targets/sources present in the field are static and immobile, while linear data models are considered not pertinent for tracking applications. Here the framework in [42] is generalized in *nonlinear* highly *dynamic* and time-varying settings where sensors acquire information about multiple *moving* targets whose number may also be changing in time.

There is a plethora of strategies which address the multi-target-tracking problem, for example see the partial list [8, 9, 61, 71, 76–78, 119] and references therein. The tracking process here is carried out via particle filtering (PF) [5, 29, 41] due to its flexibility to handle nonlinear and/or nonGaussian scenarios. PF will be combined here with the aforementioned sparse factorization scheme to cope with the time-varying settings and perform real-time association of sensors and targets. A common assumption present in existing

multi-target schemes, e.g., [31, 42, 52, 79, 83, 118], is that sensor measurements contain information about just *one* target. Here we relax this assumption given that sensors may be sensing multiple targets at the same time among which one of them is closer to the sensor than the rest.

1.1.2 Distributed Spatio-Temporal Association and Tracking of Multiple Sources in Sensor Network

The task of tracking simultaneously many sources using sensor measurements at spatially scattered locations is extremely useful in a number of applications varying from surveillance to environmental monitoring [3]. The majority of existing tracking approaches, such as the network schemes in see e.g., [1], [27, 28, 70, 82, 84, 133], extend standard techniques such as Kalman filtering or particle filtering, see e.g., [4, 29]. The aforementioned approaches are developed under the assumption that the sensing model parameters are available. Such an assumption enables sensors to identify which sources they sense, which can further simplify the tracking process. However, in many settings it is not known which sensors observe each of the underlying sources in a monitored field, while the signal attenuation from a source to a sensor is unavailable giving rise to an observation model with unknown parameters. In such settings a source-to-sensor association scheme is essential.

Alternative Kalman filtering schemes have been designed for settings where there is uncertainly in the state and observation model parameters, e.g., [73, 98], [124], [126], [59], [90]. Different from these approaches, here no a priori information is available about the observation model parameter values, i.e., the sensing matrix entries. In practice, sources present in the monitored field are localized and affect only a small percentage of the sensors present in the sensor network (SN). Interestingly, such a localized structured can be translated to a sensing matrix which has a large number of negligible (or zero) entries, i.e., a sparse matrix. Sparsity is exploited here to recover the unknown sparse sensing matrix in

the measurement model, while tracking the different source states. To this end, norm-one regularization techniques, see e.g., [114, 131], will be employed to enhance the standard Kalman filter framework. The idea of sparsity has been exploited in the context of tracking [22, 60, 122], though the difference with respect to the present setting is that sparsity is present in the source states and not in the sensing matrix.

In practice, sources present in the monitored field are localized and affect only a small percentage of the sensors present in the sensor network (SN). For instance, ground vibrating sources produce signals that undergo an exponential attenuation as they propagate in the ground. Such signals can be sensed in the measurements of sensors located a few meters away from the sources [116]. Interestingly, such a localized structured can be translated to a sensing matrix which has a large number of negligible (or zero) entries, i.e., a sparse matrix. Sparsity in the sensing matrix will be exploited here to jointly recover the sensing matrix and obtain tracking estimates for the, *not necessarily sparse*, field source states. The minimization formulation for the Kalman filter/smoother, see e.g., [4], will be enhanced with a pertinent norm-one regularization term. The sparsity-inducing terms will enable associating sources with sensors, and thus identify the sensors that acquire informative observations about the sources and use only those subsequently for tracking. Similar to the multi-target tracking strategy, here the sparsity-inducing schemes will rely only on sensor measurements to associate sources with sensors.

Two different network topologies of complementary nature are considered here. A fusion center (FC) based topology is treated first in which a fusion-center is responsible for processing the sensor data and carrying out the association and tracking. Sparsity is combined with the introduction of a power constraint that enables utilization of a small percentage of sensors that are source-informative while a transmission power budget from the sensors to the fusion center is not exceeded. The resulting novel constrained minimization formulation is tackled here via coordinate descent tools, see e.g., [10]. Power considera-

tions in estimation and tracking have been considered in estimation and tracking [23], [56], though without taking into consideration issues such as source-to-sensor association and unknown model parameters.

The requirement for a more scalable and failure-resilient sensing architecture, while compromising computational speed, leads to tackle the novel norm-one regularized Kalman minimization framework in an ad hoc sensing topology. After reformulating the latter minimization problem in a separable form, the alternating direction method of multipliers (ADMM) combined with block coordinate descent, see e.g., [10, 15], is utilized to obtain an in-network algorithmic scheme that is capable of associating sensors with sources while tracking the source states.

1.1.3 Exploiting Sensor Mobility and Covariance Sparsity For Distributed Tracking of Multiple Sparse Targets

In recent years, potential applications of sensor networks (SN) have expanded due to the low cost of the sensing units, their ability to cover large areas and the robustness distributed processing offers. One characteristic exploited more and more in sensor networks is sensor mobility and the design of kinematic rules that control sensor movement. Sensor mobility adds extra flexibility to a sensor network making it capable of covering larger areas, as well as being more energy efficient and robust [107]. Mobile sensors have been extensively utilized in target tracking applications to enable sensors to closely follow the moving target(s) and provide accurate target location estimates [21, 80, 86]. The aforementioned approaches require all the sensors to keep active [86], [80], [21], which may lead to excessive resource consumption despite the targets' locality and the fact that in practice a small portion of sensors may possess useful information about the present targets. The aim here is to design an adaptive scheme that exploits mobility and covariance sparsity to

associate targets with sensors, and then properly determine kinematic strategies only for the informative sensors which will closely follow the field targets.

In the absence of sensor mobility, there has been a plethora of approaches for tracking multiple targets while associating targets' with sensor measurements. Existing works [42, 52, 85, 118] associate measurements acquired at static sensors with targets across time and rely heavily on probability models. A distributed Kalman filtering scheme is proposed in [20] relying on information diffusion strategies. In [20] only neighboring sensors collaborate, though all sensors in the network are utilized to track a single source while sensors have fixed locations. A different approach is followed in [99], where consensus-averaging is employed across the whole sensor network and all the sensors are forced to be active irrespective of the quality of their measurements. In [70], a related single-target distributed tracking approach is proposed, in which extended Kalman filtering is employed for tracking. A probability model is assumed to determine informative sensors which may lead to instability due to its dependence on the tracking estimates. Different in this paper, distributed tracking of multiple targets will be considered, while sensor mobility will be exploited, and combined with a sensor-to-target association scheme for selecting target-informative sensors without the need of relying on model parameters and state estimators that maybe inaccurate and result divergence. It should be pointed out that the distributed characterization here is referring to the fact that i) only neighboring sensors need to communicate with each other and collaborate for multi-target tracking; while ii) processing will take place in a few head sensors and will not involve all sensors in the network but only those sensors that bear information about the moving targets.

When tracking multiple-targets with mobile sensors, the approach in [80] proposed an active sensing model, whereas the target-sensor association is based on a nearest neighbor rule which heavily relies on the accuracy of the state estimator while a central processing center is required. The scheme in [66] tackled the problem of moving sensors using a

flock control law where all sensors are utilized, while the targets are some of the moving sensors whose position is known. The approach in [36] is utilizing clustering and neural networks to move sensors under the assumption that target locations are available. The scheme in [37], designs a Kalman filtering approach with gradient descent based kinematic rules under the assumption that it is known which targets every sensor observes bypassing in that way the essential sensor-to-target association step. These schemes involve movement of all sensors at every time instant leading to resource-demanding algorithms that do not exploit spatial locality of the field targets.

Measurements corresponding to sensors which are close to the same target tend to be statistically correlated. Given that targets are spatially localized and affect small portions of the sensor network, an approximately sparse sensor data covariance matrix is emerging. Sparsity (presence of a many zero entries in a vector or matrix), has been exploited in a wide range of applications including sparse regression, and statistical inference, e.g., see [114, 137]. The problem of associating targets to sensors, as well as determining the sensors with target-informative measurements is formulated here as the task of decomposing a matrix into sparse factors. The sparse matrix factorization techniques in [93, 101] are integrated here with proper sensor kinematic strategies and tracking techniques to exploit sensor mobility. Note that in [93, 101], a stationary (immobile) sensor network is considered where sensors have fixed locations. Tracking in [93, 101] is performed by immobile sensors, whereas here tracking is generalized to a mobile network with the more challenging task of designing and integrating with multi-target tracking, sensor kinematic strategies that improve tracking accuracy while preserving local sensor network connectivity. Norm-one and norm-two regularization mechanisms are employed to formulate a pertinent minimization framework that recovers sparse covariance factors, while estimates the number of targets on the field. Coordinate descent techniques [10, 112] are employed to derive local updating recursions that allow sensors to associate with targets.

Different from the aforementioned tracking schemes using sensor mobility, here only the target-informative sensors will be enabled to move at every time-instant and track closely the moving targets. Thus, only target-affected portions of the sensor network will be used for tracking the moving targets, potentially resulting better resource consumption and prolongation of the network lifetime. Kinematic rules will be designed by minimizing proper error covariance matrices obtained by extended Kalman filtering recursions [63] used to track each of the targets. The minimization will be performed under connectivity constraints that ensure the moving sensors stay connected and are able to communicate. The modified barrier method [10, pg. 423] is employed to solve a pertinent constrained minimization problem and obtain distributed kinematic rules that the mobile sensors can apply locally without the need of a central controller. In contrast to existing approaches, the novel framework identifies and controls the movement *only* of target-informative sensors allowing for accurate tracking.

1.1.4 Regularized Kernel Matrix Decomposition in Thermal Video Multi-Object Tracking

Tracking of moving objects in videos is a fundamental problem in computer vision, and a plethora of work has been put forth to address the tracking problem using RGB (red, green, blue) cameras, see e.g., [108], [14], [74], [57]. Nonetheless, there are many challenges that still need to be addressed such as object/camera motion, varying appearances of the objects, different illumination conditions and occlusions. Further, the presence of a changing number of multiple objects in a frame sequence makes tracking still an extremely challenging problem.

Recently, un-cooled thermal sensors have become affordable and achieve improved resolution capability [35]. Further, there is an increasing interest in utilizing thermal sensors to facilitate vision tasks, such as face recognition, and human-robot interaction, [109],

[24]. Moreover, in moving object tracking applications like outdoor surveillance, where usually the background temperature is largely different from moving objects, thermal imaging becomes crucial in detecting and tracking those objects that radiate thermal energy such as humans, animal or vehicles. It is noteworthy that thermal imaging is not affected by shadow and light illumination, which normally is a bottleneck for RGB or other visible cameras, rendering it more suitable for moving object tracking in both daytime and nighttime, [88].

Thermal cameras output corresponds usually to gray scale imaging, which results a lower data processing complexity, in contrast to the triple data load produced by RGB cameras. Also, there are some research efforts that propose fusion of thermal and RGB visible data, e.g, [48], [45], [34]. The work in [34] relies on the contour saliency map, to fuse together object locations and contours from both thermal and color sensors and eventually extract the object silhouette features, thus obtaining improved tracking performance. However, the method is computationally expensive since it aims at constructing a complete object contour. In [48], data fusion is implemented to fuse thermal and visible data, resulting in an illumination-invariant face image. In the latter work, decision fusion combines the matching score generated from individual face recognition models. Indeed, modal fusion enables better tracking performance since more data is utilized. However, in many practical scenarios where only one of the imaging modalities is available to use, tracking systems can benefit from the utility of thermal data due to the computational cost savings introduced.

In this work we propose a novel approach to perform joint detection and tracking of multiple moving objects in thermal videos. Having no prior information on the objects present in the video frames, the object detection problem is formulated as the problem of factorizing a kernel covariance matrix into sparse factors. The pixels consisting of an object will be determined by estimating the support of these sparse factors and employing clus-

tering of the nonzero entries to separate individual objects. Each object will be tracked via alternative implementation of Kalman filtering and the proposed kernel matrix sparse factorization scheme. The idea of sparse covariance factorization was first explored in [101] to determine informative sensors in a network. However, in [101], linear data models are considered which is not the case in the video object tracking setting considered here. Further, the approach in [101] focuses in detecting stationary and static sources, whereas in the proposed work here nonlinear inter-pixel correlations are extracted and utilized along with multiple object dynamics to achieve accurate multi-object tracking. Coordinate descent techniques [112], [10] are employed to decompose the formulated kernel covariance matrix in a recursive way. Moreover, the implementation of computationally efficient 'divide-and-conquer' based schemes mitigate the high computational burden of factorizing large kernel covariance matrices resulting from frames having large dimensions and acquired at fast rates. The Kalman filter [63] is further combined with the aforementioned kernel sparse factorization scheme to allow precise tracking of the detected objects in videos.

1.2 Prior work

Single-target tracking techniques have been developed for SNs using consensus-averaging techniques [28, 72] combined with the skeleton of particle filtering, e.g., see [29]. Further, extended Kalman filtering (EKF) for tracking a single-target is combined with a probabilistic framework for selecting sensors in [70]; an EKF for distributed multi-target tracking is considered in [91]. Data association and particle filtering have been applied in multi-target tracking applications where the measurements from a *single* sensor are used, while association takes place in time to determine which measurements contain information about a target [31, 42, 52]. Probabilistic models on the number of targets and the target-measurement assignments are also employed in [79] to perform multi-target track-

ing in single-sensor settings. Improved particle sampling techniques for single sensor settings are considered in [130], where particles corresponding to closely spaced targets are sampled jointly. The latter approaches require the availability of a probabilistic data model which is utilized to associate measurements acquired across time with the targets present. A centralized algorithm, that relies on Markov chain Monte Carlo (MCMC) tools, performs data association on measurements acquired at a single-sensor across time in polynomial time [85]. The previous framework is extended to a network of sensors in [83]. Again the data association performs matching among temporal measurements and targets. Other centralized approaches that perform data association in time utilize Monte Carlo filtering, see e.g., [118] and [42].

A distributed algorithm that combines joint probabilistic data association with Kalman filtering has been developed in [99]. Though, some limitations are that linear Gaussian measurements models are assumed which are not always suitable for tracking applications, e.g. in low SNR environments and/or when the sensor observations are bearing and range (see e.g. [78]), while the consensus-averaging methods [125] employed, force all sensors to be active and be used in the tracking process despite the fact that some of them may have low quality observations. A different approach is followed in [105] where multiple fusion centers are present in the sensor network and evaluate the posterior Cramer-Rao lower bound that requires knowledge of the underlying data model. Then, as long as the fusion centers know which targets they track, then they can select the sensors which result the smallest Cramer-Rao lower bound. The novel algorithm proposed here does not require linear data models to operate and furthermore does not assume that sensors are aware of which targets they track. A related distributed approach for tracking a *single* target is also proposed in [70]. The latter approach utilizes extended Kalman filtering, while assuming a probabilistic model to determine the sensors that are closely located to the target. Further, the scheme in [70] relies on the tracking algorithm target position estimates and leads to

instability in noisy environments as will be demonstrated via numerical tests. Finally, work has been done in sensor scheduling and tracking in [38, 64] where the focus is to determine at which time-intervals a sensor operates and when it should be idle. Further, the approach in [132] assumes the availability of the target position to activate sensors using tree-based structures in the network topology.

A number of Kalman filtering based schemes were studied when there is uncertainly in the state and observation model parameters. In [73, 98], known state and observation model parameters with some additive error is considered. A robust Kalman filter is developed in [124], which relies on the assumption that an uncertainty norm in the state and measurement models can be upper bounded. The latter work is extended in [126] to incorporate uncertainty with an upper bounded norm in the state and measurement noise covariance matrices. Robust Kalman filtering approaches have also been developed in sensor network settings. The work in [59] considers uncertainties in the measurement model introduced by a unknown sensor-to-fusion center channel. The channel follows a probabilistic on-off model, which is assumed known, and incorporated in the Kalman filter, to decide whether to use or drop measurements. Similarly, the work in [90] considers the design of Kalman filtering techniques in the presence of noise covariance matrix uncertainties with a bounded norm for a fusion-center based multi-sensor setting.

Single target tracking using mobile sensor networks has been studied for a variety of different scenarios, [67, 134, 136]. Most of these approaches control the movement of *all* sensors by minimizing the estimation error covariance, [134], [21], while the approach in [136] manages sensor mobility based on a Bayesian estimation model and restricting sensors to move only on a grid of locations. A path planning strategy for a setting involving a fixed-location target and a single moving sensor is designed in [81] by maximizing the determinant of the Fisher information matrix corresponding to the configuration. In [32], an approach is proposed for controlling the trajectories of multiple UAVs by minimizing

the localization uncertainty for a fixed-location target setting where the target is emitting a radio signal. The work in [65] rigorously presents how sensor mobility can increase spatial resolution when tracking a target with mobile sensors.

Various approaches has been put forth in the realm of video object tracking, *e.g.*, in [135], a sparse representation technique is utilized to extract features for the video objects. Compressed feature vectors are first obtained by the sparse representation technique, then a Bayes binary classifier is designed to track the object. A subspace model is learned in [12] to model the object of interest in videos, though it is an offline tracking approach. [88] proposed to use a particle filter to track object motion features preprocessed from the Wigner distribution. Support vector machines and Kalman filtering are combined toward identifying and tracking pedestrians in [127]. In [128], a scheme is developed to detect the pedestrian head, and pedestrian legs which are later tracked by local search. The aforementioned approaches are limited in the sense that cannot jointly detect and track multiple objects, while they have to impose certain pixel intensity thresholding or statistical/structural assumptions for the objects present.

1.3 Advantages of the proposed algorithms

In contrast to the aforementioned approaches, our proposed multi-target tracking framework exhibits several advantages: i) no fusion center is required to implement target-sensor association ii) only sensor observation data is acquired in our proposed scheme; iii) it can handle non-Gaussian noise and nonlinear sensor observations; iv) only a small portion of sensors will be included in the tracking process.

When tracking underlying sources with model parameter uncertainty, our proposed FC-based method utilizes just a sensor subset to jointly recover the unknown sensing matrix

while respecting a limited power budget; our distributed method extends the FC-based scheme to a more scalable and stable architecture.

In the scenario that mobile sensor network is employed, our proposed scheme heirs the benefits from our target-sensor association algorithm which potentially save sensor deployment cost and prolong the lifespan of the whole SN, moreover, it exhibits favorable tracking performance after combining our proposed sensor kinematic strategy.

As for the video object tracking sector, compared to existing works, our approach works with the original video data directly, without the need of extracting certain object features or any prior knowledge of object location or search window size.

CHAPTER 2

Distributed Spatio-Temporal Association and Tracking of Multiple Targets Using Multiple Sensors

2.1 Problem formulation

Consider an ad-hoc multi-sensor network with a total number of m sensors. Each sensor is able to communicate with its single-hop neighboring sensors which are within its range. The single-hop neighborhood for sensor j will be denoted by $\mathcal{N}_j$, while the sensor network (SN) is modeled as an undirected graph and the inter-sensor links are assumed to be symmetric [see dashed lines (single hop) in Fig. 2.1]. The connectivity information of the SN is summarized by the $m \times m$ adjacency matrix $\mathbf{E}$ whose (j, j')th entry will be 1 if sensors j and j' are connected and zero otherwise. Sensors monitor a field on which an unknown and time-varying number of multiple moving targets is present. The targets on the field are sensed via measurements $x_j(t)$ acquired at sensor j and time instant t. For instance, in Fig. 2.1, there are two targets whose location is denoted by the red and green stars. The targets are moving at spatially different locations in the field affecting different parts of the SN. A general setting is considered where new targets are sensed at a given time instance, while other targets maybe becoming inactive (e.g., they are eliminated in a tactical environment). This leads to a setting where the number of targets is time-varying.

Sensors measure the intensity of signals received from the different moving targets on the field. Sensor j acquires a *scalar* measurement at time instant t that adheres to the following model

$$\mathbf{x}_j(t) = \sum\nolimits_{\rho=1}^{R} b_\rho(t) d_{j,\rho}^{-2}(t) + w_j(t),\ j = 1, \ldots, m \tag{2.1}$$

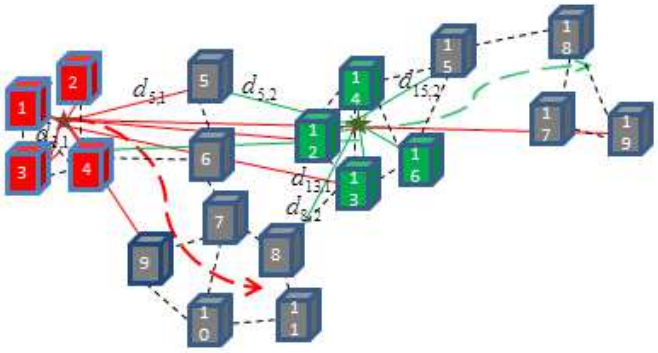

Figure 2.1. Tracking multiple targets in a sensor network..

where $b_\rho(t)$ denotes the intensity of a signal *emitted* by the ρth target, while $d_{j,\rho}(t)$ denotes the distance between the ρth target and sensor j at time t. The number R corresponds to the total number of different targets that move through the field over the lifetime of the SN, while $w_j(t)$ denotes the zero-mean temporally white sensing noise with variance equal to σ_w^2. Note that (4.3) is formulated assuming that the targets act as transmitters. The signal $b_\rho(t)$ emitted from target ρ propagates via free-space to arrive at sensor j attenuated as $b_\rho(t)/d_{j,\rho}^2(t)$. From wireless transmission (see [40, Ch. 2]), it is known that signals emitted from different targets and propagating via free-space are superimposed in the way described in (1), while the additive noise corresponds to random perturbations generated by the sensors' electronic components. Each of the $b_\rho(t)$ signals emitted by a moving target can be the result of, e.g., a radar signal impinging on the ρ target surface and then bouncing back. Thus, $b_\rho(t)$ could be viewed as the signal resulting after the radar signal has bounced back from target ρ surface. If the radar signal has intensity $\beta(t)$, then the intensity of the signal emitted by the target would be proportional to $\beta_\rho(t) \sim \frac{\beta(t)}{d_\rho^2(t)}$, where $d_\rho(t)$ denotes the distance of the radar from target ρ. This would give rise to fourth-order distance terms in (4.3), however here since $\beta_\rho(t)$ corresponds to the signal emitted by target ρ second-order distance terms appear in (4.3). Assuming that each sensor will receive one reflection of the

bounced radar signal,then the superimposing argument given earlier can be used here to derive (4.3).

Assuming that targets are sufficiently separated in space, the intensity of the signals bouncing back from the target surfaces are considered to be uncorrelated. Note from (4.3) that among the summands $b_\rho(t)d_{j,\rho}^{-2}(t)$ some have very small amplitude when sensor j is far from target ρ, whereas others have large amplitude when sensor j is close to target ρ [$d_{j,\rho}(t)$ is small]. Here it is assumed that among these summands in (4.3) only one has strong amplitude whereas the rest are negligible. This pertains to a setting where only one target, say the ρth target, is close to sensor j whereas the rest are sufficiently far thus their impact is very small. This can be realized when targets are well separated in space. This is a more 'relaxed' version of the common assumption that sensor measurements in multi-target tracking contain information about just *one* target [31,42,52,79,83,118]. The intensity $b_\rho(t)$ will be nonzero only for the interval for which a target is sensed by the sensors, otherwise will be zero deactivating target ρ in (4.3). For instance, if a target is sensed moving within interval $[t_1, t_2]$, then $b_\rho(t) = 0$ for $t < t_1$ and $t > t_2$. At a given time t a subset of the targets, say of cardinality $r(t)$, will be active ($b_\rho(t) \neq 0$) in (4.3). The distance term $d_{j,\rho}(t)$ is equal to $\|\mathbf{p}_j - \mathbf{p}_\rho(t)\|$, where $\|\cdot\|$ denotes the Euclidean norm, $\mathbf{p}_j \in \mathbb{R}^{2\times 1}$ is the fixed and available position of sensor j, while $\mathbf{p}_\rho(t) := [p_{\rho,x_1}(t), p_{\rho,x_2}(t), \ldots, p_{\rho,x_K}(t)]^T \in \mathbb{R}^{K\times 1}$ denotes the unknown ρth target position in a K-dimensional manifold.

Each target, say the ρth is characterized by a $2K \times 1$ state vector $\mathbf{s}_\rho(t)$ that contains at a given time t its location $\mathbf{p}_\rho(t)$ and the velocity $\mathbf{v}_\rho(t) := [v_{\rho,x_1}(t), \ldots, v_{\rho,x_K}(t)]^T$ at the K different spatial directions, i.e., $\mathbf{s}_\rho(t) := [\mathbf{p}_\rho^T(t), \mathbf{v}_\rho^T(t)]^T$. The target states evolve according to a general Markov model:

$$\mathbf{s}_\rho(t+1) = \mathbf{g}(\mathbf{s}_\rho(t), \mathbf{u}_\rho(t)), \tag{2.2}$$

where $\mathbf{g}(\cdot,\cdot) : \mathbb{R}^{2K\times 1} \times \mathbb{R}^{2K\times 1} \to \mathbb{R}^{2K\times 1}$ is family of nonlinear, nonsingular functions, while $\mathbf{u}_\rho(t)$ denotes the state noise. Details on the state model used for the numerical tests will be given in Sec. 2.6.

Fusing all sensor measurements in (4.3) on an $m \times 1$ vector we obtain the following measurement model

$$\mathbf{x}_t = \mathbf{D}_t\mathbf{b}_t + \mathbf{w}_t, \text{ where } \mathbf{b}_t := [b_1(t)\ b_2(t)\dots b_R(t)]^T, \tag{2.3}$$

while $\mathbf{D}_t$ is a $m \times R$ matrix with entries $\mathbf{D}_t(j,\rho) = d_{j,\rho}^{-2}(t)$ with $j = 1,\dots,m$ and $\rho = 1,\dots,R$. The noise $\mathbf{w}_t$ has covariance $\mathbf{\Sigma}_w = \sigma_w^2\mathbf{I}_m$, where $\mathbf{I}_m$ denotes the $m \times m$ identity matrix. Note that vector $\mathbf{x}_t$ is not stored somewhere and it is introduced here for notational purposes. Given that the entries of $\mathbf{b}_t$ are uncorrelated, it follows that the data covariance matrix is

$$\mathbf{\Sigma}_{x,t} = \mathbf{D}_t\mathbf{\Sigma}_b\mathbf{D}_t^T + \sigma_w^2\mathbf{I}_m = \bar{\mathbf{D}}_t\bar{\mathbf{D}}_t^T + \sigma_w^2\mathbf{I}_m, \tag{2.4}$$

where $\mathbf{\Sigma}_b$ is a diagonal matrix whose diagonal entries correspond to the variance of the entries in $\mathbf{b}_t$, while $\bar{\mathbf{D}}_t := \mathbf{B}_t\mathbf{\Sigma}_b^{1/2}$. Note that the matrix $\mathbf{D}_t$ is time-varying since the distance of the sensors from the targets is changing with time. Further, among the R entries in $\mathbf{b}_t$, there will be $r(t)$ nonzero entries corresponding to the active targets moving at the sensed field at t. Inactive targets at time instant t (either far away from sensors) will be represented by zero $b_\rho(t)$'s that will further zero out the corresponding columns in $\mathbf{D}_t$. Here it is assumed that once a target becomes inactive (i.e. $b_\rho(t) = 0$) it remains inactive.

The ρth column of $\mathbf{D}_t$ contains the distances of all sensors from target ρ at time t. For sensors close to target ρth, the corresponding distances, $d_{j,\rho}(t)$, will be relatively small, thus leading to relatively large entries $\mathbf{D}_t(j,\rho) = d_{j,\rho}^{-2}(t)$, compared to sensors that are further away. For example in Fig. 2.1, where the number of targets $R = 2$ and $K = 2$, target 1 (green star) will be close to sensors $\{1,2,3,4\}$, while target 2 will be close to sensors $\{12,13,14,16\}$. The measurements of the aforementioned sensors are expected to

be strongly affected by the target intensity signals and have a richer information content about the present targets compared to the remaining sensors that are further away. Since targets at a given instant t are very localized and close to a small percentage of sensors, many entries of any column, say the ρth, in $\mathbf{D}_t$ are expected to be close to zero giving rise to an approximately sparse matrix $\mathbf{D}_t$. Matrices $\mathbf{D}_t$ here are constantly changing, due to the presence of mobile targets. This is to be constrasted with the stationary setting in [101] where sources are immobile and the corresponding covariance matrix time-invariant.

Notice that the matrices $\mathbf{D}_t$ are not available since the targets' locations are not available. Further, someone may approximate $\mathbf{D}_t$ by first applying tracking techniques to estimate the targets' locations and subsequently the entries of $\mathbf{D}_t$. However, the presence of a time-varying number of multiple-targets and the fact that it is unknown which target corresponds to each measurement make our framework challenging and different than the one in [101]. Nonetheless, if there was a way to locate where the strong-amplitude and small-amplitude entries are located in the ρth column $\mathbf{D}_{t,\rho:}$ then we can identify which sensors are close and acquire informative observations about a specific target, say the ρth. This step of associating sensor measurements with targets will be of paramount importance before applying any tracking techniques. A spatio-temporal data association framework will be designed here that allows sensors to collaborate and determine which subsets of sensors acquire informative measurements about the $r(t)$ active targets at time instant t. This will be executed by employing sparsity-regularization techniques to estimate $\mathbf{D}_t$ and decompose it into sparse factors. Note that existing data association schemes [31, 42, 52, 79, 83, 118] match measurements with targets across time and rely on probabilistic models. The sensor-target association framework proposed here will then be integrated with particle filtering techniques that will be encountered across the different sets of informative sensors to track accurately the targets' positions. The goal here is twofold: i) determine the active targets

and identify the sensors that acquire informative measurements about them; and ii) perform tracking among the different subsets of target-informative sensors.

2.2 Spatio-Temporal Target-to-Sensor Data Association

2.2.1 Sparsity-Cognizant Minimization Framework

As mentioned earlier, at a given time instant, the number of targets present on the monitored field is unknown, as well as which sensors have informative measurements about a specific target. Let $\mathcal{T}_{\rho,t}$ denote the subset of sensors that are closely located to target ρ and whose measurements [see eq. (4.3)] are dominated by the ρth target component at time t. This implies that for all sensors $j \in \mathcal{T}_{\rho,t}$ the corresponding measurements will be approximately distributed as

$$\mathbf{x}_j(t) \sim \mathcal{N}(b_\rho(t) d_{j,\rho}^{-2}(t), \sigma_w^2), \tag{2.5}$$

since sensors $j \in \mathcal{T}_{\rho,t}$ are much closer to target ρ than the rest targets, resulting $b_\rho(t) d_{j,\rho}^{-2}(t)$ to be the dominant summand in (4.3). This stems directly from the assumption introduced in Sec. II that targets are sufficiently separated in space, which further implies that $d_{j,\rho} \ll d_{j,\rho'}$ for $\rho' \neq \rho$.

Next, we derive a technique to track the sensor subsets $\mathcal{T}_{\rho,t}$. In this way we will manage to associate sensor measurements with targets. Note that the data association here is spatio-temporal and will be done in a distributed manner and does not rely on specific probabilistic models. Different from [101], the subsets $\mathcal{T}_{\rho,t}$ are time-varying due to the moving targets. Thus, the stationary framework developed in [101] is generalized here to dynamic and time-varying settings. Note that the rank of the information component in the data covariance in (4.5), which is equal to $r(t)$, reveals the number of active sensed targets at time instant t, whereas the relatively strong-amplitude entries in each column $\{\mathbf{D}_{t,:\rho}\}_{\rho=1}^{R}$ reveal the members of a target-informative subset $\mathcal{T}_{\rho',t}$. With these properties in mind, it is

of interest to decompose the sensor data covariance matrix $\mathbf{\Sigma}_{x,t}$ into sparse factors whose nonzero entries will indicate where the strong-amplitude entries are in $\bar{\mathbf{D}}_t$.

Note that the sensor data covariance $\mathbf{\Sigma}_{x,t}$ is time-varying due to the changing number of targets and their movements, while in practical situations the ensemble covariance is not available. This is to be contrasted with the setting in [101], where the sensor data covariance matrix is time-invariant. The covariance matrix should be updated in a way that gives more emphasis to the more recent sensing data while forgets the old data gradually. This process is achieved by utilizing exponential weighing, and is a common technique in adaptive signal processing to estimate efficiently time-varying covariance matrices, see e.g., [100, 103]. Specifically, the covariance entries are estimated by

$$\hat{\mathbf{\Sigma}}_{x,t} = \frac{1-\gamma}{1-\gamma^{t+1}} \sum_{\tau=0}^{t} \gamma^{t-\tau} (\mathbf{x}_\tau - \bar{\mathbf{x}}_t)(\mathbf{x}_\tau - \bar{\mathbf{x}}_t)^T, \tag{2.6}$$

where $\gamma \in (0,1)$ denotes the forgetting factor that controls the 'memory' process and

$$\bar{\mathbf{x}}_t = \frac{1-\gamma}{1-\gamma^{t+1}} \sum_{\tau=0}^{t} \gamma^{t-\tau} x_\tau, \tag{2.7}$$

corresponds to an adaptive estimate for the data ensemble mean which is also time-varying. Note that $\gamma^{t-\tau}$ decreases as τ decreases (corresponding to past data), while for $\tau = t$ (present datum) the coefficient multiplying $\mathbf{x}_t$ is equal to one. Thus, the present datum is multiplied by the maximum possible value that $\gamma^{t-\tau}$ can reach. The scaling $(1-\gamma)(1-\gamma^{t+1})^{-1}$ in (4.6) and (4.7) is introduced here to ensure that the time-varying covariance and mean estimates $\hat{\mathbf{\Sigma}}_{x,t}$ and $\bar{\mathbf{x}}_t$ will be unbiased estimates of the ensemble quantities $\mathbf{\Sigma}_{x,t}$ and $\mathbb{E}[\mathbf{x}_t]$ respectively, in a time-invariant (stationary) setting, i.e.

$$\mathbb{E}[(1-\gamma)(1-\gamma^{t+1})^{-1}\hat{\mathbf{\Sigma}}_{x,t}] = \mathbf{\Sigma}_{x,t}, \text{ and } \mathbb{E}[\bar{\mathbf{x}}_t] = \mathbb{E}[\mathbf{x}_t].$$

Thus, the scaling introduced in (4.6) and (4.7) ensures that the obtained adaptive estimates are properly normalized to give unbiased estimates in a stationary setting which further implies good estimation in nonstationary settings too, see e.g., [100].

In order to adhere to the single-hop connectivity constraints summarized in the adjacency matrix $\mathbf{E}$, each sensor j is responsible for evaluating the 'single-hop' covariance entries $\hat{\mathbf{\Sigma}}_{x,t}(j,j')$ where $j' \in \mathcal{N}_j$. For example, sensor 5 in Fig. 2.1 will be able to evaluate only the single-hop covariance entries $\hat{\mathbf{\Sigma}}_{x,t}(5,5), \hat{\mathbf{\Sigma}}_{x,t}(5,6)$. The latter tasks involve the exchange of scalar measurements $x_j(t)$ between single-hop neighbors during time t. Thus, covariance entries that correspond to sensors more than one hop away will not be evaluated in the SN.

A standard least-squares based matrix factorization scheme would minimize the Frobenius norm-based cost $\|\hat{\mathbf{\Sigma}}_{x,t} - \mathbf{M}_t\mathbf{M}_t^T - \sigma^2\mathbf{I}_{m\times m}\|_F^2$ with respect to (wrt) the factor estimates in $\mathbf{M}_t \in \mathbb{R}^{m\times r}$. However, such a formulation does not account for the nearly sparse structure of $\bar{\mathbf{D}}_t$. In fact it assumes that the number r of factors (sensed targets) is available, while all covariance entries are available. The need for a framework that accounts for sparsity, unknown number of targets and single-hop connectivity is apparent. To this end, the following framework is put forth

$$\left(\hat{\mathbf{M}}_t, \{\hat{\sigma}_j\}_{j=1}^m\right) := \arg\min_{\mathbf{M}_t, \{\sigma_j\}_{j=1}^m} \|\mathbf{E} \odot \left(\hat{\mathbf{\Sigma}}_{x,t} - \mathbf{M}_t\mathbf{M}_t^T - \text{diag}(\sigma_{1,t}^2, \ldots, \sigma_{m,t}^2)\right)\|_F^2 + \sum_{\ell=1}^{L} \lambda_\ell \|\mathbf{M}_{t,:\ell}\|_1 + \phi \sum_{\ell=1}^{L} \|\mathbf{M}_{t,:\ell}\|_2^2, \quad (2.8)$$

where $\odot$ denotes the Hadamard operator (entry-wise matrix product), σ_j^2 is the local noise variance estimate at sensor j, while L is an upper bound for the number of active sensed targets $r(t)$ ($L \geq r(t)$) and $\mathbf{M}_{t,:\ell}$ denotes the ℓth column of $\mathbf{M}_t$. Although the sensing noise variance σ_w^2 is common across all sensors we introduce different noise variance estimates $\sigma_{j,t}^2$ to facilitate the development of a decentralized iterative minimization technique for (4.8). Matrix $\mathbf{M}_t \in \mathbb{R}^{m\times L}$ contains L columns that will estimate the sparse matrix columns of $\bar{\mathbf{D}}_t$, while L is selected sufficiently large to ensure that is an upper bound for the number of present targets $r(t)$.

Sensor j will be responsible for updating the jth row in $\mathbf{M}_t$, namely $\mathbf{M}_{t,j:}$ for $j = 1, \ldots, m$. The adjacency matrix $\mathbf{E}$ in (4.8) along with the nature of the Hadamard operator ensure that only the available single-hop covariance entries will participate in the minimization formulation, while the updating recursions that will be obtained later for each sensor j to find $\hat{\mathbf{M}}_{t,j:}$ will require message exchanges only between single-hop neighbors. The first term in (4.8) accounts for the fact that the covariance assumes the structure in (4.5). The second (norm-one) term in (4.8) induces sparsity in the columns of $\mathbf{M}_t$ to account for the approximately sparse structure of $\bar{\mathbf{D}}_t$. Norm-one regularization is well known to affect sparsity in several estimation and regression problems [114, 137]. The larger the nonnegative sparsity-controlling coefficient λ_ρ is, the more zeros the estimated factor $\hat{\mathbf{M}}_{t,:\rho}$ will contain. The third term in (4.8), where $\phi \geq 0$, is present to adjust the number of nonzero columns of $\hat{\mathbf{M}}_t$ needed to accurately represent $\hat{\mathbf{\Sigma}}_{x,t}$. The number of nonzero columns in $\hat{\mathbf{M}}_t$ will be smaller than L and can be used as an estimate for the number of sensed targets $r(t)$ at time t.

Notice that the optimization formulation in (4.8) is also different from the one given in [101]. The difference is in the last two terms which are there to control the number of nonzero rows in matrix $\mathbf{M}_t$. In fact the number of nonzero rows in $\mathbf{M}_t$ will correspond to an estimate of the number of targets present in the field. The scheme in [101] works under the assumption that the number of sources is known. Another feature of the minimization formulation in (4.8), not present in [101], is the estimation of the sensing noise variances σ_j^2, which in general are unknown, and not available as is the case in [101].

2.2.2 Decentralized Algorithm

An iterative algorithm is proposed here to minimize numerically the cost in (4.8) derived using coordinate descent techniques [10, 112]. The approach followed here is to minimize the cost in (4.8) recursively wrt an entry of $\mathbf{M}_t$ or $\text{diag}(\sigma_1^2, \ldots, \sigma_m^2)$, while keep-

ing the remaining entries fixed. During one coordinate descent cycle all the entries of matrix $\mathbf{M}_t$ and $\text{diag}(\sigma_{1,t}^2, \ldots, \sigma_{m,t}^2)$ are updated. Sensor j is responsible for updating the entries $\{\mathbf{M}_t(j,\ell)\}_{\ell=1}^L$ and $\sigma_{j,t}^2$. Given the most recent updates $\hat{\mathbf{M}}_t^{k-1}$ and $\{\sigma_{j,t,k-1}^2\}$ at the end of coordinate cycle $k-1$, updates $\hat{\mathbf{M}}_t^k(j,\ell)$ at sensor j can be formed by differentiating (4.8) wrt $\mathbf{M}_t(j,\ell)$ while fixing the rest of the minimization variables to their most up-to-date values from cycle $k-1$. It turns out that (see Apdx. A) during coordinate cycle k, the update $\hat{\mathbf{M}}_t^k(j,\ell)$ can be obtained as the value that achieves the minimum possible cost in (4.8) (while fixing the rest of the variables) among the candidate values: i) $y=0$; ii) the real positive roots of the third-degree polynomial

$$4y^3 + 4\left[\sum_{i\in\mathcal{N}_j}[\hat{\mathbf{M}}_t^{k-1}(i,\ell)]^2 - \zeta_{t,\Sigma}^k(j,j,\ell) + 0.5\phi\right]y - \left[4\sum_{i\in\mathcal{N}_j}\zeta_{t,\Sigma}^k(j,\mu,\ell)\hat{\mathbf{M}}_t^{k-1}(i,\ell)\right] + \lambda_\ell = 0 \tag{2.9}$$

and iii) the real negative roots of the third-degree polynomial

$$4y^3 + 4\left[\sum_{i\in\mathcal{N}_j}[\hat{\mathbf{M}}_t^{k-1}(\mu,\ell)]^2 - \zeta_{t,\Sigma}^k(j,j,\ell) + 0.5\phi\right]y - \left[4\sum_{i\in\mathcal{N}_j}\zeta_{t,\Sigma}^k(j,i,\ell)\hat{\mathbf{M}}_t^{k-1}(i,\ell)\right] - \lambda_\ell = 0 \tag{2.10}$$

where

$$\zeta_{t,\Sigma}^k(j,i,\ell) := \hat{\mathbf{\Sigma}}_{x,t}(j,i) - \delta_{j,i}\hat{\sigma}_{j,t,k-1}^2 - \sum_{\ell'=1,\ell'\neq\ell}^{L}\hat{\mathbf{M}}_t^{k-1}(j,\ell')\hat{\mathbf{M}}_t^{k-1}(i,\ell') \tag{2.11}$$

while $\delta_{j,i}$ denotes the Kronecker delta, i.e., $\delta_{j,i}=1$ if $j=i$, and $\delta_{j,i}=0$ if $j\neq i$.

Further, the noise variance estimates across sensors can be updated during cycle k at time instant t as

$$\hat{\sigma}_{j,t,k}^2 = \hat{\mathbf{\Sigma}}_{x,t}(j,j) - \hat{\mathbf{M}}_{t,j:}^k(\hat{\mathbf{M}}_{t,j:}^k)^T, \quad j=1,\ldots,m. \tag{2.12}$$

The roots of (2.9) and (2.10) can be obtained using, companion matrices [50]. Sensor j can evaluate the coefficients of the polynomials in (2.9) and (2.10) by communicating only with its neighbors in $\mathcal{N}_j$. In detail, sensor j receives $\{\hat{\mathbf{M}}_t^{k-1}(i,1),\ldots,\hat{\mathbf{M}}_t^{k-1}(i,L)\}$ and

the latest measurements $\{x_i(t)\}$ from sensors $i \in \mathcal{N}_j$ to form the single-hop covariance updates $\hat{\mathbf{\Sigma}}_{x,t}(j,i)$ and subsequently evaluate $\zeta^k_{t,\Sigma}(j,i,\ell)$. Similarly, it sends to its neighbors the L scalar updates for the jth row of $\mathbf{M}_t$, namely $\{\hat{\mathbf{M}}^{k-1}_t(j,1),\ldots,\hat{\mathbf{M}}^{k-1}_t(j,L)\}$ and its current measurement $x_j(t)$. Further, each sensor j can update the noise variance estimates $\hat{\sigma}^2_{j,t,k}$ using only locally available information as can be seen in (2.12). To facilitate a real-time implementation a small fixed number, say κ, of coordinate cycles is applied per time t. Note that the proposed scheme also involves constant updating of the single-hop covariance entries $\hat{\mathbf{\Sigma}}_{x,t}(j,i)$ needed in $\zeta^k_{t,\Sigma}(j,i,\ell)$ to account for the constantly changing statistical properties of the sensed field. Such online updating is not present in [101].

The task of forming the updates $\{\hat{\mathbf{M}}_t(j,\ell)\}_{\ell=1}^{L}$ at sensor j at time instant t boils down to determining the roots of the third-degree polynomials given in (9) and (10). The latter task involves: i) evaluating the quantities $\{\zeta^k_{t,\Sigma}(j,\mu,\ell)\}_{\mu\in\mathcal{N}_j\cup\{j\},\ell=1,\ldots,r}$, with a computational complexity of the order of $\mathcal{O}(|\mathcal{N}_j|r^2)$, i.e., $L|\mathcal{N}_j|$ coefficients each evaluated in (11) with a complexity of $\mathcal{O}(L)$; ii) evaluating the $4|\mathcal{N}_j|$ coefficients of the polynomials at (9) and (10) with a complexity of $\mathcal{O}(4|\mathcal{N}_j|L)$; and iii) determining the roots of the third-order polynomials in (9) and (10) that involves evaluation of the corresponding $2L$ companion matrices of size 3×3 at a computational complexity of $\mathcal{O}(L)$. Note that the cost at sensor j per coordinate cycle is linearly dependent on the number of single-hop neighbors $|\mathcal{N}_j|$, while the dependency is quadratic when it comes to the upper bound of the number of targets L. Nevertheless, the number of sources r (and thus L) in practical scenarios is much smaller than the number of sensors m.

Once the sparse factors $\{\hat{\mathbf{M}}_{t,:\ell}\}_{\ell=1}^{\hat{r}(t)}$ are estimated, where $\hat{r}(t) < L$ corresponds to the number of nonzero columns of $\hat{\mathbf{M}}_t := \hat{\mathbf{M}}^{\kappa}_t$ at t, their support (nonzero entries) can be used to identify the sensors that sense a specific target at time instant t. In that way sensor subsets $\mathcal{T}_{\ell_t,t}$ for $\ell_t = 1,\ldots,\hat{r}(t)$ can be identified and used to track $\hat{r}(t)$ different targets. One challenge that will be addressed in Sec. 2.4 is how to determine whether two

subsets $\mathcal{T}_{\ell_t,t}$ and $\mathcal{T}_{\ell'_t,t'}$ evaluated at different time instances correspond to the same target or not. This time-association step is necessary to make sure that estimated trajectories corresponding to different targets are updated using newly acquired sensor measurements that correspond to the correct dominant target. At a given time instant t the steps followed across sensors, which form a connected network, to perform decentralized data association is tabulated as Algorithm 1. During time instant t one coordinate cycle k involves updating the $m \times L$ entries $\hat{\mathbf{M}}_t^k$ and the m variance estimates $\sigma^2_{j,t,k}$ via (2.9), (2.10) and (2.12). In Apdx. B it is demonstrated that Alg. 1 converges at least to a stationary point of (4.8). The parameters $\{\lambda_\ell\}_{\ell=1}^L$ can be set using the strategy proposed in [101].

To end the iterative process involved in Alg. 1, each sensor j proceeds to evaluate the Euclidean norm of the difference between two consecutive estimates, namely $\|\hat{\mathbf{M}}^{k-1}_{t,j:} - \hat{\mathbf{M}}^k_{t,j:}\|_2$, found during iteration steps k and $k-1$. Using a max consensus scheme, e.g., [54], the maximum of these m norm quantities can be found across sensors which then they compare this maximum with a desired threshold of accuracy. Once the maximum norm $\|\hat{\mathbf{M}}^{k-1}_{t,j:} - \hat{\mathbf{M}}^k_{t,j:}\|_2$ (sensor with largest updating difference) is less than a threshold ϵ which could be set as a adjustable small positive value (in our tests is set as $5 \cdot 10^{-3}$), then the updating process involved in Alg. 1 will stop across sensors.

2.3 Tracking via Particle Filtering

Next we will take into our advantage the target-informative sensor subsets $\mathcal{T}_{\rho,t}$ which have been retrieved using the decentralized framework in Sec. 2.2 in order to perform multi-target tracking. We focus on executing the tracking process via particle filtering (PF) [5,41] due to its flexibility to handle nonlinear and/or nonGaussian scenarios as in our observation model in (4.3). For each subset $\mathcal{T}_{\rho,t}$ a different PF will be constructed to track the corresponding target ρ.

Algorithm 1 Distributed Target-Sensor Association

1: At time instant t:

2: Sensor j updates $\hat{\mathbf{\Sigma}}_{x,t}(j,j')$ for $j' \in \mathcal{N}_j \cup \{j\}$ using (4.6) after receiving the most recent data $\{x_{j'}(t)\}_{j' \in \mathcal{N}_j}$ from its neighbors.

3: Sensor j initializes the jth row of $\mathbf{M}_t$ as $\hat{\mathbf{M}}^0_{t,j:} = \mathbf{0}_{1\times L}$, while it sets $\hat{\sigma}^2_{j,t,0} = 0$.

4: **for** $k = 1, 2, \ldots, \kappa$ **do**

5: Each sensor j for $j = 1, \ldots, m$:

6: Transmits $\{\hat{\mathbf{M}}^{k-1}_t(j,\ell')\}^L_{\ell'=1}$ to its neighbors in $\mathcal{N}_j$, and receives $\{\hat{\mathbf{M}}^{k-1}_t(j',\ell')\}^L_{\ell'=1}$ from $j' \in \mathcal{N}_j$.

7: Evaluates $\varsigma^k_{t,\Sigma}(j,i,\ell)$ for $i \in \mathcal{N}_j \cup \{j\}$ via (2.11).

8: Determine the updates $\{\hat{\mathbf{M}}^k_t(j,\ell)\}^L_{\ell=1}$ after determining the positive roots of (2.9) and the negative roots of (2.10).

9: If $\max_{j=1,\ldots,m}(\|\hat{\mathbf{M}}^{k-1}_{t,j:} - \hat{\mathbf{M}}^k_{t,j:}\|_2) \leq \epsilon$ then stop.

10: **end for**

Tracking objects consists of computing a conditional expectation

$$\mathbb{E}(f(\mathbf{s}_{\rho,t})|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) = \int f(\mathbf{s}_{\rho,t}) p(\mathbf{s}_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) d\mathbf{s}_{\rho,t}$$

of a function of the state, $\mathbf{s}_{\rho,t}$, of a target ρ, using the measurements of the sensors within the informative subset $\mathcal{T}_{\rho,t}$. Equivalently, the conditional density $p(\mathbf{s}_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})$ given the measurements needs to be computed instead. These measurements are denoted herein by the $|\mathcal{T}_{\rho,t}| \times 1$ vector $\mathbf{x}_{\mathcal{T}_{\rho,t}} := \{x_j(t)\}_{j\in\mathcal{T}_{\rho,t}}$, where $|\mathcal{T}_{\rho,t}|$ denotes the cardinality of sensor subset $\mathcal{T}_{\rho,t}$. The measurements are affiliated with a pertinent likelihood function which depends on the underlying observation model. We generally denote this likelihood function by $p(\mathbf{x}_{\mathcal{T}_{\rho,t}}|\mathbf{s}_{\rho,t})$ given the state $\mathbf{s}_{\rho,t}$ of the ρth target at time t. The reader may refer to Sec. 2.6 for specific details on the likelihood and the associated observation model. All the available data from time 0 up to the current time instant t will be used. Let $\mathbf{x}_{\mathcal{T}_{\rho,0:t}}$ denote

the sensor measurements associated with the informative sensor subsets $\mathcal{T}_{\rho,0}, \mathcal{T}_{\rho,1}, \ldots, \mathcal{T}_{\rho,t}$ within the time horizon $[0, t]$.

However, in many instances, it is a rather formidable task to compute or approximate the conditional density, $p(\mathbf{s}_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})$. Therefore, employing importance sampling techniques, one may consider a different distribution, say $q(\mathbf{s}_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})$ and the aforementioned conditional expectation is derived

$$\mathbb{E}(f(\mathbf{s}_{\rho,t})|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) = \int f(\mathbf{s}_{\rho,t}) \frac{p(\mathbf{s}_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}{q(\mathbf{s}_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})} q(\mathbf{s}_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) d\mathbf{s}_{\rho,t}.$$

Consequently, if one draws Q samples, $\mathbf{s}^i_{\rho,t},\ i = 1, \cdots, Q$, from the proposal distribution $q(\mathbf{s}_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})$, the conditional expectation is in turn approximated by

$$\mathbb{E}(f(\mathbf{s}_{\rho,t})|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) \approx \frac{1}{Q} \sum_{i=1}^{Q} f(\mathbf{s}^i_{\rho,t}) \frac{p(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}{q(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}, \tag{2.13}$$

and by further approximating $Q \approx \sum_{i=1}^{Q} \frac{p(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}{q(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}$ we have that

$$\mathbb{E}(f(\mathbf{s}_{\rho,t})|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) \approx \frac{\sum_{i=1}^{Q} f(\mathbf{s}^i_{\rho,t}) \frac{p(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}{q(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}}{\sum_{i=1}^{Q} \frac{p(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}{q(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}}. \tag{2.14}$$

Defining

$$w^i_{\rho,t} \propto \frac{p(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})}{q(\mathbf{s}^i_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})} \tag{2.15}$$

to be the ith weight which corresponds to the ith particle $\mathbf{s}^i_{\rho,t}$, the conditional expectation is approximated by

$$\mathbb{E}(f(\mathbf{s}_{\rho,t})|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) \approx \sum_{i=1}^{Q} w^i_{\rho,t} f(\mathbf{s}^i_{\rho,t}).$$

The particle filter is an importance sampling with a special importance density $q(\cdot)$. To identify this density $q(\cdot)$, one takes into account that the conditional distribution, $p(\mathbf{s}_{\rho,0:t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})$, can be written

$$p(\mathbf{s}_{\rho,0:t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) \propto p(\mathbf{x}_{\mathcal{T}_{\rho,t}}|\mathbf{s}_{\rho,t}) p(\mathbf{s}_{\rho,t}|\mathbf{s}_{\rho,t-1}) p(\mathbf{s}_{\rho,0:t-1}|\mathbf{x}_{\mathcal{T}_{\rho,0:t-1}}), \tag{2.16}$$

where $\propto$ denotes that the two probability density functions (pdfs) on the right and left hand sides will be equal after appropriate scaling with a constant not dependent on the unknown state. Furthermore, one may consider that the importance density is factorized [5] such that

$$q(\mathbf{s}_{\rho,0:t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) = q(\mathbf{s}_{\rho,t}|\mathbf{s}_{\rho,0:t-1}, \mathbf{x}_{\mathcal{T}_{\rho,0:t}}) q(\mathbf{s}_{\rho,0:t-1}|\mathbf{x}_{\mathcal{T}_{\rho,0:t-1}}). \tag{2.17}$$

However, only a filtering estimate is propagated at each time step. Therefore, the importance density $q(\mathbf{s}_{\rho,t}|\mathbf{s}_{\rho,0:t-1}, \mathbf{x}_{\mathcal{T}_{\rho,t}})$ depends only on $\mathbf{s}_{\rho,t-1}$ and $\mathbf{x}_{\mathcal{T}_{\rho,t}}$ which yields that $q(\mathbf{s}_{\rho,t}|\mathbf{s}_{\rho,0:t-1}, \mathbf{x}_{\mathcal{T}_{\rho,t}}) = q(\mathbf{s}_{\rho,t}|\mathbf{s}_{\rho,t-1}, \mathbf{x}_{\mathcal{T}_t})$. Employing the framework of eqs. (2.16) and (2.17) into eq. (2.15) at each time step, we have that the weight corresponding to the ith particle can be updated by

$$w^i_{\rho,t} \propto w^i_{\rho,t-1} \frac{p(\mathbf{x}_{\mathcal{T}_{\rho,t}}|\mathbf{s}^i_{\rho,t}) p(\mathbf{s}^i_{\rho,t}|\mathbf{s}^i_{\rho,t-1})}{q(\mathbf{s}_{\rho,t}|\mathbf{s}_{\rho,t-1}, \mathbf{x}_{\mathcal{T}_{\rho,t}})}, \tag{2.18}$$

where $w^i_{\rho,t-1} = p(\mathbf{s}^i_{\rho,0:t-1}|\mathbf{x}_{\mathcal{T}_{\rho,0:t-1}})/q(\mathbf{s}^i_{\rho,0:t-1}|\mathbf{x}_{\mathcal{T}_{\rho,0:t-1}})$. A popular choice for the density $q(\mathbf{s}^i_{\rho,t}|\mathbf{s}^i_{\rho,t-1}, \mathbf{x}_{\mathcal{T}_{\rho,t}}) = p(\mathbf{s}^i_{\rho,t}|\mathbf{s}^i_{\rho,t-1})$ such that the weights are given by

$$w^i_{\rho,t} \propto w^i_{\rho,t-1} p(\mathbf{x}_{\mathcal{T}_{\rho,t}}|\mathbf{s}^i_{\rho,t}) \; ; i = 1, \ldots, Q \tag{2.19}$$

where $\mathbf{s}^i_{\rho,t}$ is the ith sample from the Markov transition density $p(\mathbf{s}^i_{\rho,t}|\mathbf{s}^i_{\rho,t-1})$ and $p(\mathbf{x}_{\mathcal{T}_{\rho,t}}|\mathbf{s}^i_{\rho,t})$ the corresponding likelihood function associated with the measurements $\mathbf{x}_{\mathcal{T}_{\rho,t}}$. Consequently, the posterior filtering density is approximated by

$$p(\mathbf{s}_{\rho,t}|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}) \approx \sum_{i=1}^{Q} w^i_{\rho,t} \delta(\mathbf{s}_\rho(t) - \mathbf{s}^i_{\rho,t}),$$

where the weights are defined in (2.19) and δ is the Dirac delta function.

One may easily conclude that the particle filter's implementation is straightforward and can be adapted for different problems as long as the algorithm is tuned according to the specific dynamics. Precisely this has led to the particle filter algorithm's increased popularity [30]. However, it has been shown, in [41] for example, that particle filter suffers from

degeneracy and that it needs a lot of samples in order to describe accurately the conditional density $p(\mathbf{s}_\rho(t)|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})$. Therefore several resampling schemes have been incorporated in order to alleviate this problem, e.g. cf [41, 61, 78, 104] by replicating samples with significant weight and disregarding the rest. In our paper, we used the popular multinomial resampling scheme, e.g. see [41]. After the resampling stage, the particles $\mathbf{s}^i_{\rho,t}$, are used to estimate the state for target ρ at time instant t. Of course there are multiple targets present in the field, thus in the next section it is described how the sensor-target distributed association algorithm in Sec. 2.2 can be combined with PF to track effectively multiple-targets.

2.4 Joint Sensor-Target Association and Particle Filtering

Here it is described in detail how the PF unit in Sec. 2.3 and the sensor-target association scheme in Sec. 2.2 interact to enable tracking using only target-informative sensors in the SN.

Specifically, during a start-up stage each sensor acquires T_s measurements, namely $\{x_j(\tau)\}_{\tau=-(T_s-1)}^{0}$. It is assumed that the sampling rate is fast enough such that the present targets, say $r(0)$ in number, are essentially stationary/immobile. The T_s acquired data are then used by the distributed sensor-target association framework in Sec. 2.2 to determine the sets of informative sensors $\{\mathcal{T}_{\rho^0_\ell,0}\}_{\ell=1}^{\hat{r}(0)}$ where each $\rho^0_\ell \in \{1,\ldots,R\}$ for $\ell = 1,\ldots,\hat{r}(0)$, and $\hat{r}(0)$ is the estimated number of $r(0)$ sensed targets at time $t = 0$. One sensor in each set $\mathcal{T}_{\rho^0_\ell,0}$ is designated as a leading sensor $C_{\rho^0_\ell,0}$ which collects from all sensors $j \in \mathcal{T}_{\rho^0_\ell,0}$ their corresponding measurements $x_j(0)$ and their position $\mathbf{p}_j$ for $j \in \mathcal{T}_{\rho^0_\ell,0}$ and $\ell = 1,\ldots,\hat{r}(0)$. During initialization the leading sensor $C_{\rho^0_\ell,0}$ can be selected randomly among the sensors in $\mathcal{T}_{\rho^0_\ell,0}$. Then, for time $t > 0$ it will be described later on how the leading sensors are

selected. Each leading sensor $C_{\rho_\ell^0,0}$, for $\ell = 1,\ldots,\hat{r}(0)$, then calculates the 'average' informative sensors' position as

$$\hat{\mathbf{p}}_{\rho_\ell^0}(0) = \sum_{j\in\mathcal{T}_{\rho_\ell^0,0}} \mathbf{p}_j, \quad \ell = 1,\ldots,\hat{r}(0). \tag{2.20}$$

Then, each leading sensor $C_{\rho_\ell^0,0}$ uses the corresponding average location in (2.20) to initialize the PF recursions in Sec. 2.3, and find a state estimate $\hat{\mathbf{s}}_{\rho_\ell^0}(0)$ for target ρ_ℓ^0 using the informative measurements $x_j(0)$, for $j \in \mathcal{T}_{\rho_\ell^0,0}$ and $\ell = 1,\ldots,\hat{r}(0)$.

Suppose that at time t each leading sensor $\{C_{\rho_\ell,t}\}$ has available state estimates $\hat{\mathbf{s}}_{\rho_\ell}(t)$ for $\ell = 1,\ldots,\hat{r}(t)$. From $\hat{\mathbf{s}}_{\rho_\ell}(t)$ the estimated target position $\hat{\mathbf{p}}_{\rho_\ell}(t)$ can be extracted and it is utilized to select a set of 'candidate' target-informative sensors, namely $\mathcal{J}_{\rho_\ell,t+1}$, for target ρ_ℓ. Specifically, the leading sensor $C_{\rho_\ell,t}$ transmits $\hat{\mathbf{s}}_{\rho_\ell}(t)$ to its single-hop neighbors, which will subsequently transmit to their own neighbors and the estimate propagates in time. A sensor j that receives $\hat{\mathbf{s}}_{\rho_\ell}(t)$ will forward this estimate only to those neighbors in $j' \in \mathcal{N}_j$ that are located within a radius R_s from the estimated target location, i.e., $\|\mathbf{p}_{j'} - \hat{\mathbf{s}}_{\rho_\ell}(t)\|_2 \leq R_s$. Note that through the aforementioned process the set of sensors $\mathcal{J}_{\rho_\ell,t+1}$ selected at time $t+1$ has the following two properties: i) each sensor $j \in \mathcal{J}_{\rho_\ell,t}$ is located within a radius R_s of the estimated position $\hat{\mathbf{p}}_{\rho_\ell}(t)$, i.e., $\|\mathbf{p}_j - \hat{\mathbf{p}}_{\rho_\ell}(t)\| \leq R_s$; and ii) sensors in $j \in \mathcal{J}_{\rho_\ell,t+1}$ form a connected communication subgraph characterized by the adjacency matrix $\mathbf{E}_{\rho_\ell,t+1}$ which is going to be a submatrix of $\mathbf{E}$ after keeping the rows and columns with indices in $\mathcal{J}_{\rho_\ell,t+1}$.

In each of the subsets $\mathcal{J}_{\rho_\ell,t+1}$ the distributed targets-to-sensors data association algorithm (Alg. 1) is employed to determine the target-informative sensor subsets $\mathcal{T}_{\rho_\ell,t+1} \subseteq \mathcal{J}_{\rho_\ell,t+1}$ for each of the targets ρ_ℓ at time instant $t+1$. The radius R_s through which $\mathcal{J}_{\rho_\ell,t+1}$ are constructed is up to our control, and the faster the target moves the larger R_s should be set to guarantee that all target-informative sensors are included in $\mathcal{J}_{\rho_\ell,t+1}$. Performing the sensor-target association algorithm in different sensor subsets $\mathcal{J}_{\rho_\ell,t+1}$ of the SN facilitates

tracking the present targets, while it requires less computational and communication complexity than when applied in the whole SN. In fact, there will also be instances where the sensors-targets association algorithm (Alg. 1) will be implemented across the whole SN whenever it is detected that the present targets may have changed in number and we have to redetermine the target-informative subsets. Indicators used to determine when to apply the scheme in Sec. 2.2 across the whole SN are the following:

C.1 If any of the estimated target-informative sets $\mathcal{T}_{\rho_\ell,t+1}$, returned by Algorithm 1, are empty. This implies that most likely some of targets being tracked at previous time instances are not present in the sensed field anymore.

C.2 If the energy of the measurements of a given sensor, not currently in any set $\mathcal{T}_{\rho_\ell,t}$ exceeds a certain threshold. This implies that most likely a new target has entered the sensed field and this is indicated by an elevated energy level in the measurements of a currently non-informative sensor.

The procedure described earlier for the start-up stage is applied every time it is determined that the distributed sensors-targets association scheme has to be applied in the whole network. This process is necessary due to the fact that the target population has changed since old targets may not be sensed anymore, while new ones may have entered the sensed field.

The leading sensor $C_{\rho_\ell,t+1}$ is chosen as that sensor in $\mathcal{T}_{\rho_\ell,t+1}$, which is closest to the estimated position of the ρ_ℓth target, i.e.,

$$C_{\rho_\ell,t+1} = \arg \min_{j \in \mathcal{T}_{\rho_\ell,t+1}} \|\mathbf{p}_j - \hat{\mathbf{p}}_{\rho_\ell}(t)\|_2. \tag{2.21}$$

The process of electing a new leading sensor can take place among the sensors in $\mathcal{T}_{\rho_\ell,t+1}$ that can determine their distance from $\hat{\mathbf{p}}_{\rho_\ell}(t)$ and find which sensor has the minimum in a distributed fashion, e.g., see [54]. The leading sensor $C_{\rho_\ell,t+1}$ then collects i) the corresponding state particles and weights $\{\mathbf{s}^i_{\rho_\ell,t}, w^i_{\rho_\ell,t}\}_{i=1}^{Q}$ from $C_{\rho_\ell,t}$; and ii) the sensors measurements $x_j(t+1)$ for $j \in \mathcal{T}_{\rho_\ell,t+1}$, namely the updated informative sensor subset for target ρ_ℓth at

time instant $t+1$. Assuming that the sampling rate is fast enough, the target locations $\mathbf{s}_{\rho_\ell}(t)$ and $\mathbf{s}_{\rho_\ell}(t+1)$ will be close, thus a few communication hops (1 or 2 in simulations) suffice to have the previous leading sensor $C_{\rho_\ell,t}$, as well as the current target-informative sensors in $\mathcal{T}_{\rho_\ell,t+1}$ transmit their information to $C_{\rho_\ell,t+1}$. For instance, in Fig. 2.2 sensor 4 corresponds to one leading sensor at time instant t, while the informative sensors are $\{1,2,3,4\}$. Then, at time instant $t+1$ and since the target has moved the informative sensor set changes to $\{5,6,8\}$ while sensor 6 is elected as the new leading sensor at $t+1$ being closer to the estimated target position.

The leading sensor $C_{\rho_\ell,t+1}$ proceeds to draw new state particles from the importance sampling pdf $q(\mathbf{s}_\rho(t)|\mathbf{x}_{\mathcal{T}_{\rho,0:t}})$ and update their corresponding weights as in (2.19). Then, $C_{\rho_\ell,t+1}$ forms the new state estimate $\hat{\mathbf{s}}_{\rho_\ell}(t+1) \approx E[\mathbf{s}_{\rho_\ell}(t+1)|\mathbf{x}_{\mathcal{T}_{\rho,0:t}}]$ using (2.13), and extract from $\hat{\mathbf{s}}_{\rho_\ell}(t+1)$ the estimated location for target ρ_ℓ at time instant t, namely $\hat{\mathbf{p}}_{\rho_\ell}(t+1)$. The leading sensor $C_{\rho_\ell,t+1}$ transmits $\hat{\mathbf{s}}_{\rho_\ell}(t+1)$ to its single-hop neighbors and the process described earlier is repeated to update the subsets of candidate informative sensors $\mathcal{J}_{\rho_\ell,t+2}$. The joint algorithmic framework for multi-target tracking and distributed sensor-target association is tabulated as Algorithm 2.

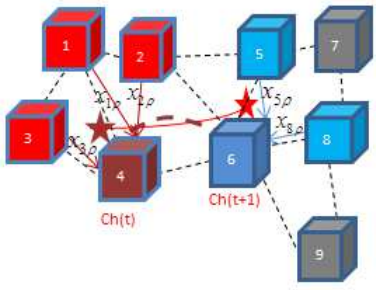

Figure 2.2. Update of the target-informative sets and the leading sensors as a target moves in the sensed field..

Algorithm 2 Joint Target-Sensor Association and Multi-Target Tracking

1: **Start-up stage ($t = 0$)/Reconfiguration ($t \neq 0$):** Each sensor j collects T_s measurements $x_j(t)$ and Algorithm 1 is applied in the whole network to determine the target-informative groups $\mathcal{T}_{\rho_\ell,t}$ and $\ell = 1, \ldots, \hat{r}(t)$, where $\hat{r}(t)$ is the estimated number of sensed targets.

2: **for** $\tau = t, \ldots,$ **do**

3: Determine the leading sensor $C_{\rho_\ell,\tau}$ in each $\mathcal{T}_{\rho_\ell,\tau}$ for $\ell = 1, \ldots, \hat{r}(t)$ as specified in (2.21).

4: Each leading sensor $C_{\rho_\ell,\tau}$ receives particles and weights from $C_{\rho_\ell,\tau-1}$, and $x_j(\tau)$ from $j \in \mathcal{T}_{\rho_\ell,\tau}$ to perform tracking for $\rho_\ell = 1, \ldots, \hat{r}(t)$ target via the PF recursions and obtain $\hat{\mathbf{s}}_{\rho_\ell}(\tau)$ via (2.13).

5: The state estimates $\hat{\mathbf{s}}_{\rho_\ell}(\tau)$ are propagated from $C_{\rho_\ell,\tau}$ via single-hop transmissions to every sensor j that can be reached from $C_{\rho_\ell,\tau}$ by a multi-hop path and satisfies $\|\mathbf{p}_j - \hat{\mathbf{p}}_{\rho_\ell}(\tau)\|_2 < R_s$. Then, the candidate informative sets $\{\mathcal{J}_{\rho_\ell,\tau+1}\}_{\ell=1}^{\hat{r}(t)}$ are formed.

6: Algorithm 1 is applied in each connected set of sensors $\mathcal{J}_{\rho_\ell,\tau+1}$ to obtain the target-informative sets $\mathcal{T}_{\rho_\ell,\tau+1}$.

7: If either condition C.1, or C.2 is true then go to step 1, otherwise go to step 2.

8: **end for**

2.5 Inter-Sensor Communication Costs

The information exchanges occurring during different steps of Algorithm 2 and the associated communication costs are outlined next. Specifically, inter-sensor communications take place when Algorithm 1 is applied to perform sensors-to-targets association and every time a leading sensor has to be updated.

In detail at time instant $t+1$ sensor j has to receive $|\mathcal{N}_j|$ scalar measurements from its neighbors, namely $\{x_{j'}(t+1)\}_{j' \in \mathcal{N}_j}$, to update $\hat{\mathbf{\Sigma}}_{x,t+1}(j, j')$ (step 1 in Alg. 1). Further, sensor j receives the updates $\{\hat{\mathbf{M}}_{t+1}^{k-1}(j', \ell)\}_{\ell=1}^{L}$, $L|\mathcal{N}_j|$ scalars in total, to form its local updates $\{\hat{\mathbf{M}}_{t+1}^{k}(j, \ell)\}_{\ell=1}^{L}$ (step 8 in Alg. 1). Thus, sensor j receives $(L+1)|\mathcal{N}_j|$ scalars. In the same way sensor j has to transmit $x_j(t+1)$ and $\{\hat{\mathbf{M}}_{t+1}^{k-1}(j, \ell)\}_{\ell=1}^{L}$, a total of $L+1$ scalars,

to its neighbors per iteration k in Alg. 1. It is worth emphasizing that the communication complexity for each sensor is linear with respect to its neighborhood size and the number L used to bound the number of targets at a given time instant. Such a complexity can be handled easily by networks of sensors. Note that the previous information exchanges occur during step 1 and step 6 in Alg. 2.

Every time the target-informative sets $\{\mathcal{T}_{\rho,t+1}\}$ are updated, the old leading sensor, say $C_{\rho,t}$, has to send to the new leader $C_{\rho,t+1}$ Q particles, each entailing $2K$ scalars, and Q corresponding scalar weights. Further, $C_{\rho,t}$ has to send out the $2K$ scalars corresponding to the estimate $\hat{\mathbf{s}}_\rho(t)$ (step 5 in Alg. 2). Thus, $C_{\rho,t}$ has to transmit $(2K+1)Q+2K$ scalars for implementing steps $3-5$ in Alg. 2. Every sensor in $\mathcal{T}_{\rho,t+1}$ has to also transmit its scalar measurement $x_j(t+1)$ that will reach $C_{\rho,t+1}$, and will be used to update the target state estimates. Thus, $C_{\rho,t+1}$ receives in total $(2K+1)Q+2|\mathcal{T}_{\rho,t}|$ scalars. Clearly, the communication cost is proportional to the number of particles Q and the dimensionality of the target state vectors $2K$.

In [70] a Monte Carlo method is employed to select the active sensors when tracking a *single* target. Specifically, at time t there is a cluster head sensor, say sensor j, that selects among its neighbors, say $\mathcal{N}_j$, a number of μ sensors whose measurements will be used to track, via EKF, the target at the next time instance. A probability of detection is calculated for each sensor in the neighborhood $\mathcal{N}_j$ by the cluster head, and the first active sensor is selected as the one in $\mathcal{N}_j$ having the highest probability of detection. This task is carried out at a computational complexity of $\mathcal{O}(|\mathcal{N}_j|)$. In the same way the second active sensor is selected, as the one among the remaining $|\mathcal{N}_j|-1$, achieving the highest value of a pertinent joint probability of detection metric; the corresponding complexity is $\mathcal{O}(|\mathcal{N}_j|-1)$. Similarly the ith active sensor is determined at a complexity of $\mathcal{O}(|\mathcal{N}_j|-i)$ for $i=1,\ldots,\mu$. Thus, the total complexity at the cluster head for selecting μ sensors is $\mathcal{O}(\mu(|\mathcal{N}_j|-\frac{\mu-1}{2}))$. Depending on the number of active sensors chosen ($1\leq\mu\leq|\mathcal{N}_j|$), if $\mu=1$ then complex-

ity is in the order of $\mathcal{O}(|\mathcal{N}_j|)$ per cluster head which is the same as in our scheme for $r = 1$ target. If $\mu = |\mathcal{N}_j|$ (all sensors selected), then the computational complexity of [70] is in the order of $\mathcal{O}(|\mathcal{N}_j|^2/2)$ which leads to a higher complexity compared to the one achieved in our case when $|\mathcal{N}|_j > 2$. Further, in [70], all neighboring sensors send their data to the current leading sensor, leading to a communication cost which is proportional to the neighborhood size $|\mathcal{N}_j|$.

The approach for choosing active sensors in [70] relies on the prediction state estimate and MSE covariance obtained through extended Kalman filtering (EKF), see, e.g., [63]. The latter two quantities are used in a Gaussian pdf to evaluate the aforementioned probabilities of detection whose values will determine the active sensors. Different from [70], the proposed sensors-to-targets association scheme does not depend on the state and observation model parameters in (4.3) and (5.1). On the contrary the novel tracking scheme here relies on the sensor measurements to update the target-informative portion of the SN, and it is not affected by the tracking algorithm [cf. (9) and (10)]. Linearization in EKF may result errors in the tracking process which can propagate to the sensor selection process in [70] and deteriorate performance. In the same way, selecting the closest sensors to the estimated target position is prone to error propagation and cannot perform better than [70]. Numerical tests will corroborate the previous claims.

2.6 Numerical Tests

2.6.1 Target Dynamics and Particle Sampling

As in the majority of methods developed for target tracking, e.g., [70] in the numerical tests we consider a scenario where the targets move according to a near constant

velocity model [7]. Specifically, the ρth target's state vector evolves according to the following model

$$\mathbf{s}_\rho(t+1) = \mathbf{A}\mathbf{s}_\rho(t) + \mathbf{u}_\rho(t), \tag{2.22}$$

where $\mathbf{A}$ is a $2K \times 2K$ transition matrix, while $\mathbf{u}_\rho(t)$ denotes zero-mean Gaussian noise with covariance $\mathbf{\Sigma}_u$. The matrices $\mathbf{A}$ and $\mathbf{\Sigma}_u$ have the following structure (e.g., see [7])

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \dots & \Delta T & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 & \dots & \Delta T \\ 0 & 0 & 1 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}, \qquad \mathbf{\Sigma}_u = \sigma_u^2 \begin{bmatrix} (\Delta T)^3/3 \cdot \mathbf{I}_K & (\Delta T)^2/2 \cdot \mathbf{I}_K \\ (\Delta T)^2/2 \cdot \mathbf{I}_K & \Delta T \cdot \mathbf{I}_K \end{bmatrix}$$

where ΔT is the sampling period, and σ_u^2 is a nonnegative constant controlling the variance of the noise entries in $\mathbf{u}_\rho(t)$ while $\mathbf{I}_K$ denotes the $K \times K$ identity matrix. The state noise is assumed to be temporally white and uncorrelated with the observation noise across sensors, namely $\mathbf{w}_t := [w_1(t), \dots, w_m(t)]^T$.

Using the state transition model in (5.1) it follows readily that state transition pdf $p(\mathbf{s}_\rho(t)|\mathbf{s}^i_{\rho,t-1})$ is Gaussian with expectation $\mathbf{A}\mathbf{s}^i_{\rho,t-1}$ and covariance $\mathbf{\Sigma}_u$. Thus, the new Q state particles at time instant t can be generated from the ones obtained at time instant $t-1$ as follows:

$$\mathbf{s}^i_{\rho,t} = \mathbf{A} \times \mathbf{s}^i_{\rho,t-1} + \mathbf{v}_t, \quad i = 1, \dots, Q \tag{2.23}$$

where $\mathbf{v}_t \in \mathbb{R}$ is a $2K \times 1$ zero-mean Gaussian vector with covariance $\mathbf{\Sigma}_u$.

From the observation model in (4.3) it turns out that the likelihood pdf of the informative observations corresponding to the sensors in $\mathcal{T}_{\rho,t}$, given the ith particle for the state of target ρ, i.e., $p(\mathbf{x}_{\mathcal{T}_{\rho,t}}|\mathbf{s}^i_{\rho,t})$ in (2.19) is Gaussian, i.e.,

$$p(\mathbf{x}_{\mathcal{T}_{\rho,t}}|\mathbf{s}^i_{\rho,t}) = \frac{1}{(2\pi\sigma_w^2)^{|\mathcal{T}_{\rho,t}|/2}} \exp\left(\frac{-\left\|\mathbf{x}_{\mathcal{T}_{\rho,t}} - \mathbf{d}_\rho(\mathbf{s}^i_{\rho,t})\right\|^2}{2\sigma_w^2} \right) \tag{2.24}$$

where σ_w^2 denotes the observation noise variance, while

$$\mathbf{d}_\rho(\mathbf{s}_{\rho,t}^i) = \left[\|\mathbf{p}_{j_1^t} - \mathbf{p}_{\rho,t}^i\|^{-2}, \|\mathbf{p}_{j_2^t} - \mathbf{p}_{\rho,t}^i\|^{-2}, \ldots, \|\mathbf{p}_{j_{|\mathcal{I}_{\rho,t}|}^t} - \mathbf{p}_{\rho,t}^i\|^{-2}\right]^T \tag{2.25}$$

where $\mathbf{p}_{j_1^t}$ is the known position of sensor j_1 at time t, while $j_1^t, j_2^t, \ldots, j_{|\mathcal{I}_{\rho,t}|}^t$ are the indices of the sensors in the informative set $\mathcal{T}_{\rho,t}$. Further, $\mathbf{p}_{\rho,t}^i$ is the estimated position for target ρ extracted from the state particle $\mathbf{s}_{\rho,t}^i$ for $i = 1, \ldots, Q$. Intuitively, the vector $\mathbf{d}_\rho(\mathbf{s}_{\rho,t}^i)$ in (2.25) can be viewed as an estimate (prediction) of the informative sensor measurements $\mathbf{x}_{\mathcal{T}_{\rho,t}}$ using the most recent particles. In this way, only the measurements from the informative sensors in $\mathcal{T}_{\rho,t}$ will be used to find the weights for the sampled particles, and thus track the corresponding target ρ.

2.6.2 Tracking of a Single-Target

We start by testing the performance of the novel tracking Alg. 2 and compare to existing alternatives in a wireless network setting with $m = 150$ sensors which are randomly placed in a region $[0, 100] \times [0, 100] m^2$. A scenario with a single target is considered first. The target starts at location $[27.00, 72.00]$ and moves with a speed of $1.8m/s$ at the x-axis and the y-axis. The state and observation models introduced in (5.1) and (4.3) are utilized here for $K = 2$.

In the following numerical tests we compare the tracking performance, via the localization root mean-square error (RMSE), among i) the novel Alg. 2; ii) the EKF approach with sensor selection in [70]; iii) EKF combined with the sensor-targets association Alg. 1 (EKF+Alg. 1); iv) EKF with all sensor measurements used and there is no sensors-targets association (EKF-All sensors); v) PF with all sensor measurements used and there is no sensors-targets association (PF-All sensors); vi) PF combined with a scheme that selects as target-informative sensors the J-nearest sensors to the current target position es-

timate (Nearest sensor) where here $J = 3$; and vii) unscented Kalman filtering (UKF) (see e.g., [55]) combined with the sensors-to-targets association Alg. 1 (UKF+Alg. 1).

In all these tracking methods, the target position is initialized by applying Alg. 1 and finding (2.20), ensuring that the initial error is the same for all different tracking approaches. As for the parameters in Alg. 2, the forgetting factor for updating (4.6) is set to $\gamma = 0.1$, while the radius R_s, used in forming the candidate sets $\mathcal{J}_{\rho,t}$, is set equal to 15. The threshold to decide which entries in $\hat{\mathbf{M}}_t$ are zero and which are nonzero is set to be 10^{-5} in the single-target case. The parameters λ_ℓ are set using the method in [101, Sec. V-A], and $\phi = 1.1$. Fig. 2.3 displays the root mean-square tracking error (RMSE) achieved by the tracking schemes described earlier versus time t. Four different test cases are considered in Fig. 2.3 in which the state noise and the measurement noise variances change. On the top left diagram a setting with relatively small state and observations noise variances, namely $\sigma_u^2 = 0.005$ and $\sigma_w^2 = 0.0001$ (corresponding to a sensing signal-to-noise ratio of 30dB), is considered. The bottom left diagram corresponds to a setting with relatively small state variance, and relatively large observation noise variance, namely $\sigma_u^2 = 0.005$ and $\sigma_w^2 = 0.1$ (corresponding to a sensing signal-to-noise ratio of 2dB). The top right diagram deals with a setting with relatively large state variance, and relatively small observation noise variance, namely $\sigma_u^2 = 0.05$ and $\sigma_w^2 = 0.0001$ (corresponding to a sensing signal-to-noise ratio of 30dB). Finally, the bottom right diagram corresponds to a setting where both the state and observation noise variances are relatively large, i.e., $\sigma_u^2 = 0.005$ and $\sigma_w^2 = 0.1$ (corresponding to a sensing signal-to-noise ratio of 2dB which is low).

Fig. 2.3 corroborates that the novel tracking Alg. 2 outperforms in terms of tracking RMSE alternative tracking approaches when the observation and/or state noise variance are relatively high. The EKF-based approaches rely on linearization and therefore the presence of high-variance noise in the state and/or observation models will result such approaches to deviate as time progresses. The method in [70] utilizes the EKF target position estimates

to determine the target informative sensors and as it can be seen in Fig. 2.3 the RMSE will diverge. The reason for the big performance gap between the EKF and Alg. 2 lies on the linearization errors involved in EKF. Such errors are accentuated in the presence of state and measurement noise of high variance. Then, bad trajectory estimates further deteriorate the sensor selection process causing an error propagation behavior. The same behavior is also exhibited when the nearest to the estimated target position sensor are selected to perform tracking. This method also diverges in the presence of high variance noises. Further, it can be seen that UKF combined with Alg. 1 performs better than EKF+Alg. 1 as expected, however its performance is still worse than Alg. 2. This is expected since UKF aims to approximate the mean and covariance of the state, whereas the particle filters in fact track the posterior pdf which in our setting is not Gaussian despite the presence of Gaussian noise.

Alg. 2 performs reliable tracking even under high-variance noise environments. The reason is that the sensors-to-targets association scheme (Alg. 1) does not rely on the tracking algorithm and employs the data directly to determine target-informative sensors. Another important property of Alg. 2 is that its tracking performance is very close to that of a particle filtering approach that uses all the sensors in the network. This demonstrates the efficiency of Alg. 2 in selecting a few target-informative sensors without compromising the tracking performance. Clearly, Alg. 2 has the potential to prolong the lifetime of the network without losing much tracking accuracy when compared to a setting where all sensors measurements are utilized during tracking. When the state and observation noise variances are small, then all six different methods perform well and reach a small tracking RMSE. In such a setting the PF-based methods appear to reach steady-state at a slower rate than the EKF-based and UKF-based approaches. The reason is the small number of particles used here $Q = 100$ per time instant t that results slower convergence. Nonetheless, what

is important is that the novel tracking Alg. 2 is able to accurately track the target even in hostile settings suffering from high variance noises.

Next, we consider the same setting as in the upper diagram of Fig. (2.3) where the state and observation noise variances, $\sigma_u^2 = 0.005$ and $\sigma_w^2 = 0.0001$, are relatively small. Fig. 2.4 depicts the number of sensors selected to utilize their measurements for tracking per time instant t. Alg. 2, the scheme in [70] and the nearest sensor selection rule for $J = 3$ are applied. Notice that the average number of selected sensors per method is equal to 3 ensuring a fair comparison in terms of RMSE in Fig. (2.3) (top). The bottomline is that for the same number of active sensors our approach outperforms existing alternatives for selecting informative sensors, while it performs closely to the benchmark (but demanding) scheme where all sensors are used to perform tracking.

2.6.3 Changing the number of particles

Fig. 2.5 depicts the tracking RMSE achieved by Alg. 2 versus time for different values of the number of particles Q used. Two different settings are considered: Fig. 2.5 (left) corresponds to a setting with relatively high state and observation noise; and Fig. 2.5 (right) corresponds to a setting with relatively low-variance state and observation noise. As expected the left diagram in Fig. 2.5 indicates worse tracking performance than the right diagram. Nonetheless, in all cases an increasing number of particles always leads to considerable improvements of the tracking performance. The reason for observing inflexions has to do with the number of sensors around the target at a give time instant. At certain time instances there may be only one sensor close to the target whereas the number of sensors may increase or decrease with time. The more sensors in the proximity of a target, the more active sensors will be selected via Alg. 1, resulting a reduction in the tracking RMSE. However, at times if there are not many sensors close to the target the RMSE may increase in value as observed in Fig. 2.5.

2.6.4 Tracking of multiple targets

A setting where three targets are moving in the sensed field is considered next. Note that the scheme in [70] and nearest sensor method are not capable to track multiple targets. The same applies when using all sensors measurements and there is no sensor-data association embedded in the tracking scheme (EKF or PF). Here we will test the tracking RMSE achieved by Alg. 2, as well as the scheme where EKF is combined with Alg. 1. A setting with $m = 120$ sensors is considered that are randomly deployed in the area of $[0, 100] \times [0, 100]m^2$. The three targets are set to move from initial positions $[30, 80]$, $[35, 25]$ and $[40, 45]$, respectively. The speed of the targets per x-axis and y-axis is set equal to $1.8m/s$. These trajectories are well separated in space and satisfy the assumption of having one dominant term in (4.3), as introduced in Sec. II. Step 1 in Alg. 2 is applied to associate sensors with targets and initialize Alg. 2. Fig. 2.6 depicts the average tracking RMSE (averaged across the three different targets) versus time for i) Alg. 2; and ii) the EKF combined with the sensors-to-targets association Alg. 1. The left diagram corresponds to a relatively low-variance state and observation noise setting, i.e., $\sigma_u^2 = \sigma_w^2 = 8 \cdot 10^{-3}$ (the corresponding sensing SNR is 13dB). The right diagram corresponds to a relatively high-variance state and observation noise setting, i.e., $\sigma_u^2 = \sigma_w^2 = 8 \cdot 10^{-2}$ (the corresponding sensing SNR is 3dB). As in the single-target case it follows that both Alg. 2 and the EKF-based schemes perform accurate tracking for the low-variance noise setting. Again Alg. 2 converges slower to steady state because of the limited number of particles, $Q = 100$, used per time instant t. In the high-variance noise setting clearly Alg. 2 is still able to track the three targets whereas the EKF based method fails since the linearization process breaks down and results misleading target position estimates that eventually diverge.

Fig. 2.7 demonstrates how the steady-state tracking RMSE behaves as a function the state and observation noise variance. Two curves are depicted, for the blue curve the x-axis corresponds to measurement noise variance while the state noise variance is set to

$\sigma_u^2 = 0.1$, whereas for the red curve the x-axis corresponds to state noise variance while the measurement noise variance is set to $\sigma_w^2 = 0.1$. Note that although the noise variance increases almost by an order of magnitude on the x-axis, the RMSE increases approximately by 1.6 meters which in the $[0, 100] \times [0, 100]$ area corresponds to a relatively small tracking performance degradation. This advocates the robustness and 'graceful' degradation of Alg. 2 in the presence of state/measurement noise.

2.6.5 Time-varying number of targets

Next, we test the tracking performance of our proposed method in a setting where the number of targets can change in time. Again a number of $m = 120$ sensors are placed randomly in the region of $[0, 100] \times [0, 100]m^2$. The total number of targets appearing and disappearing across time is $R = 12$. In that region, five groups of different targets appear and disappear orderly. The target configuration in the test is set as follows: Targets $\rho = 1, 2, 3$ start moving at positions $[35, 25]$, $[40, 45]$, $[20, 55]$ and follow the dynamics in (5.1), with a speed of $2m/s$ across the x-axis. Targets $\rho = 1, 2, 3$ move in the field for the time interval $[1, 15]$s and then are not sensed anymore. In the interval $[15, 17]$s no targets are present in the field. Then, targets $\rho = 4, 5$ start at positions $[12, 25]$, $[30, 80]$ and move according to same state model followed by the first three for the time interval $[17, 30]$s but with speed $1.5m/s$ across the x-axis. Again no targets are present during $[30, 32]$s. Then, targets $\rho = 6, 7$ show up at initial positions $[75, 35]$, $[10, 30]$ and start moving, according to (5.1), for the time interval $[32, 45]$s and speed $1.5m/s$ per axis. Two new targets, namely $\rho = 8, 9$, appear in initial positions $[40, 10]$, $[40, 70]$ and move in the field for the time interval $[47, 60]$s with speed $1.7m/s$ on the y-axis, and speed $1.5m/s$ across the x-axis. Finally, the last three targets $\rho = 10, 11, 12$ start at positions $[60, 20]$, $[60, 70]$ and $[70, 50]$ and move within the field for the time interval $[62, 72]$s. Target $\rho = 10$ follows the same state model as targets $\rho = 6, 7$, while targets $\rho = 11, 12$ follow the same state model as

targets $\rho = 1, 2, 3$. Again, the targets are placed in the field such that at every time instant t every sensor senses one dominant target in (4.3).

Here we have to emphasize that when testing the novel Alg. 2 we do not know the number of targets present in the field at a given time instant, and we do not know when the target configuration changes, with old targets vanishing and new targets showing up. Alg. 2 entail steps 1 and 7 that detect when a change in the targets' configuration may have happened and estimate the number of targets present as well as the sensors acquiring informative observations for the different targets present in the field. Thus, Alg. 2 does not really have available the time period for which each target is active; this is something that it estimates. This is to be contrasted with the sensor selection framework in [101], where the number of sources/targets is known and fixed. The parameter L in Alg. 1 is set to 4, which indicates that at every time instant the number of targets present in the field will not exceed 4. The radius R_s for determining the candidate informative sensors subsets $\mathcal{J}_{\rho,t}$ is set equal to $R_s = 10$. The forgetting factor is set $\gamma = 0.1$. The state noise variance is set as $\sigma_u^2 = 0.1$, while the measurement noise variance is also set to $\sigma_w^2 = 0.1$ (which amounts to an observation SNR of roughly 10dB).

Fig. 2.8 depicts the true target trajectories (blue dashed curves), along with the estimated trajectories from Alg. 2 (light green curves). The blue stars correspond to the starting position of the targets and the red stars denote the ending position. Clearly, Alg. 2 is able to carry out accurate tracking of all $R = 12$ targets. Another interesting property shown in Fig. 2.8 is the small number of sensors selected by Alg. 1 to be utilized in the tracking process. The red circles in Fig. 2.8 depict the target informative sensors at time instances 15s, 30s, 45s, 60s and 72s, thus all the red circles correspond to the cumulative number of informative sensors throughout the simulation. As it can be seen the informative sensors are selected such that they are closely located to a corresponding target. This further implies that Alg. 2 performs efficient tracking by utilizing only a small portion of

the sensors available in the network. It is worth mentioning that only a small portion of the network is used to gather data. In Fig. 2.9 it is depicted where the target is (red star) and what sensors are active during time-instances 45s and 72s. Clearly the active sensors are in the vicinity of the targets' location corroborating the capability of Alg. 1 to utilize the sparse covariance sensing data structure and select those sensors acquiring informative measurements about the present targets. Another tracking scenario where there are many trajectory crossings, the targets are moving at different directions and different speeds is depicted in Fig. 2.10. The noise remains the same as before and there are $R = 12$ targets in total. Specifically, targets $\rho = 1, 2, 3$ start moving at positions $[35, 25]$, $[40, 45]$, $[20, 55]$ and follow the dynamics in (5.1). While targets $\rho = 1, 3$ move at a speed of $2m/s$ across the x-axis, target $\rho = 2$ moves with a speed of $2m/s$ across the y-axis. Targets $\rho = 1, 2, 3$ move in the field for the time interval $[1, 15]$s and then are not sensed anymore. In the interval $[15, 17]$s no targets are present in the field. Then, targets $\rho = 4, 5$ start at positions $[23, 40]$, $[50, 75]$ and move according to same state model followed by the first three for the time interval $[17, 30]$s but with speeds $-1.3m/s$ and $-1.7m$ respectively across the x-axis. Again no targets are present during $[30, 32]$s. Then, targets $\rho = 6, 7$ show up at initial positions $[75, 35]$, $[10, 30]$ and start moving, according to (5.1), for the time interval $[32, 45]$s and speed $1.5m/s$ on x-axis and $1.7m/s$ across y-axis. Three new targets, namely $\rho = 8, 9, 10$, appear in initial positions $[40, 70]$, $[40, 10]$, $[60, 70]$ and move in the field for the time interval $[47, 60]$s with different speed $1.4, 1.2, 1.6m/s$ on both the y and x-axis. Finally, the last two targets $\rho = 11, 12$ start at positions $[85, 25]$ and $[48, 48]$ and move within the field for the time interval $[62, 72]$s. Target $\rho = 11$ moves with $-1.0m/s$ and $2.6m/s$ across x and y-axis, while target $\rho = 12$ with corresponding x-axis and y-axis speeds of $-0.7m/s$ and $-2.5m/s$, respectively. Clearly, it can be seen that the configuration of the targets does not really affect the tracking performance of Alg. 2 corroborating its flexibility to track under different geometric configurations of the targets.

In fact. Fig. 2.11 depicts the number of informative sensors versus time throughout the simulation. Clearly, the number of informative sensors does not exceed 20 (16% of the network), while the average number of informative sensors is equal to approximately 8. Fig. 2.11 depicts the capability of Alg. 1 to select only the target-informative sensors to perform tracking. Despite the fact that only a small percentage of the sensors is utilized, still it can track the trajectories quite accurately as seen by Fig. 2.8.

Fig. 2.12 depicts the average tracking RMSE corresponding to the tracking of the different targets present in the field at every time instant. Alg. 2, the EKF-based scheme combined with Alg. 1 and the UKF scheme combined with Alg. 1 are compared. Note that at the time intervals $15, 30, 45, 60$s the average tracking RMSE is zero. It is initialized there because during these time intervals no targets are detected in the field and thus there is nothing to track and no corresponding tracking RMSE. However, when targets are present, the superiority of Alg. 2 over the EKF based approach combined with Alg. 1 is apparent in Fig. 2.12. Clearly, the EKF-based approach cannot perform efficient tracking which is further challenged by the varying number of targets and high-variance state and observation noise. The linearization error is the reason for the big performance gap between the two aforementioned schemes. Further, it can be seen that UKF combined with Alg. 1 performs better than EKF combined with Alg. 1 as expected, however its performance is still worse than Alg. 2. Again EKF suffers from linearization errors, such errors are resolved by UKF which still is worse than Alg. 2 since it just estimates the posterior mean and covariance instead of tracking the posterior pdf (see also details for Fig. 2.3).

2.7 Conclusion

A novel method performing distributed sensor-target association and multi-target tracking was designed and tested in multi-sensor networks. Our approach is based on a

novel blending of particle filtering and sparsity-aware matrix decomposition techniques. Target-informative sensors are selected online and their measurements are used for tracking. The proposed approach is capable to detect changes in the configuration and population of the targets present in the sensed field. Extensive numerical tests show that the proposed tracking framework outperforms related approaches in tracking multiple targets. The novel tracking methodology is robust even in high-variance state and observations noises, and provides accurate estimates of the targets' position by utilizing only a small number of the available sensors.

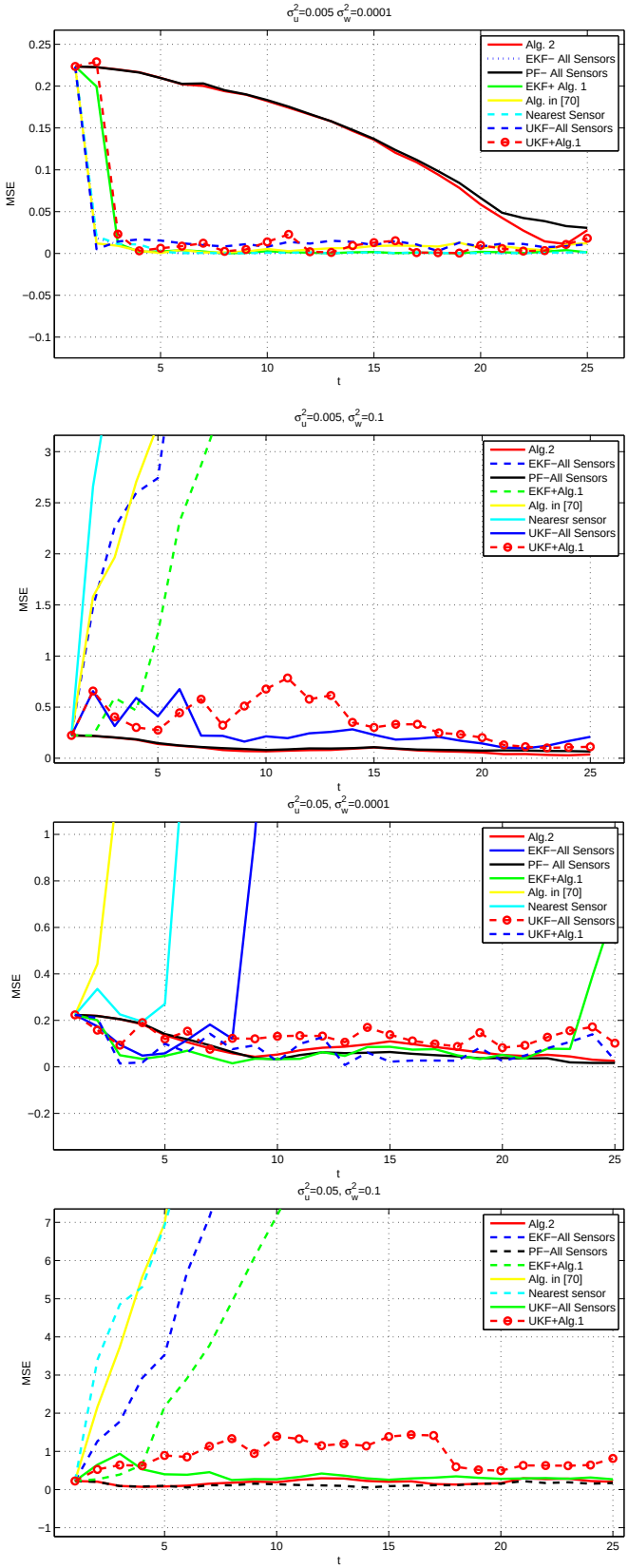

Figure 2.3. Tracking root mean-square error (RMSE) vs. time (t) for different tracking schemes and different tracking conditions..

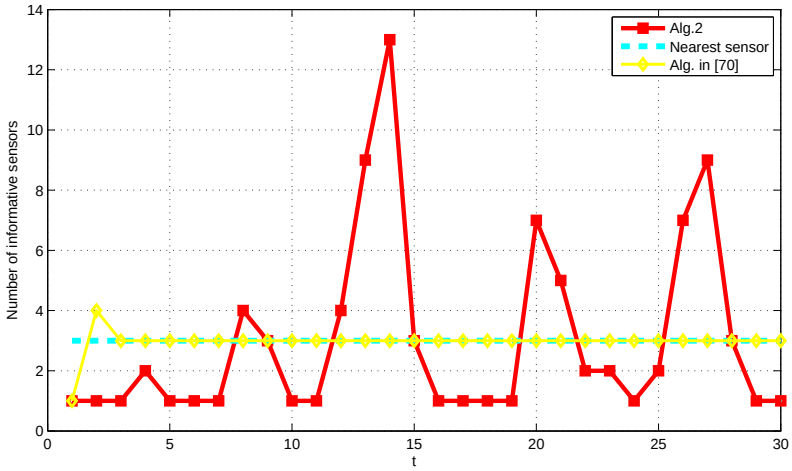

Figure 2.4. Number of selected sensors vs. time t in a single-target setting..

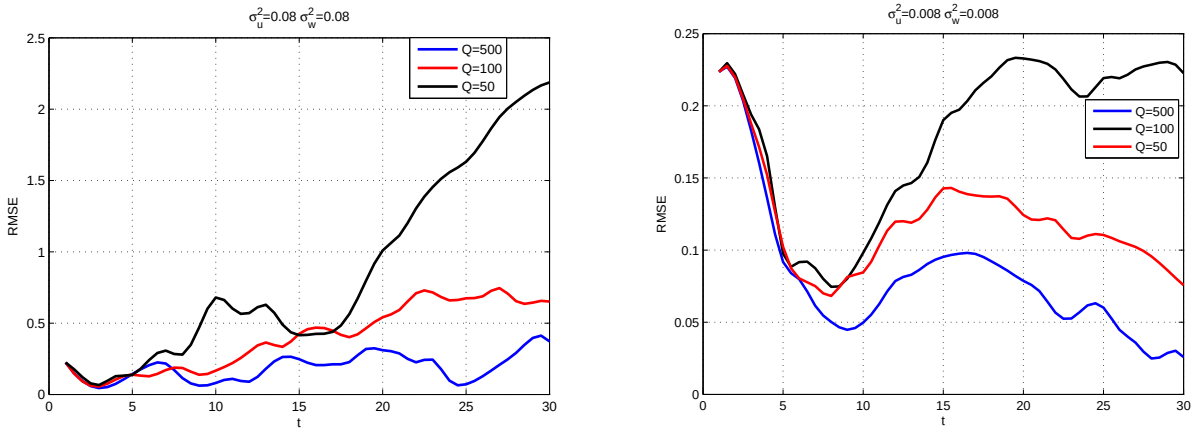

Figure 2.5. Tracking RMSE vs. time t for different number of particles Q using Alg. 2..

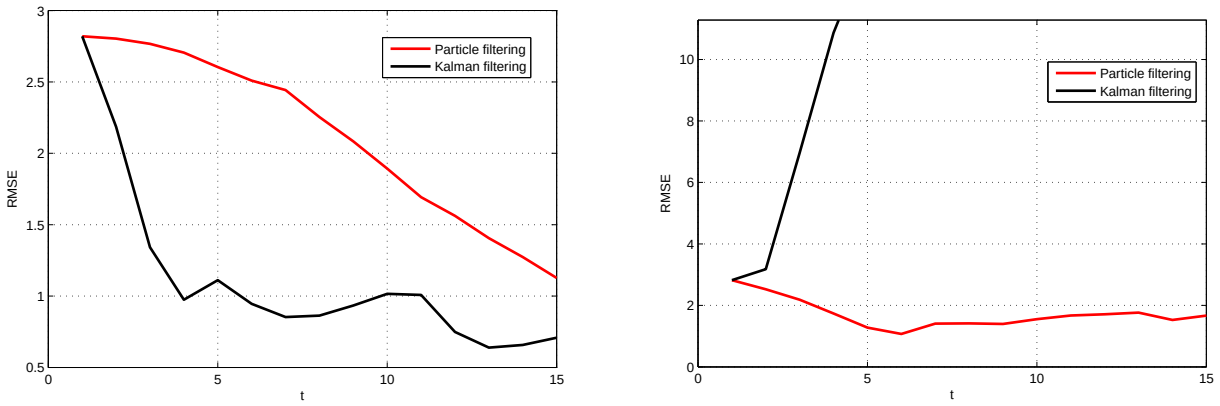

Figure 2.6. Average tracking RMSE versus time for a low-variance noise setting (left) and a high-variance noise setting (right)..

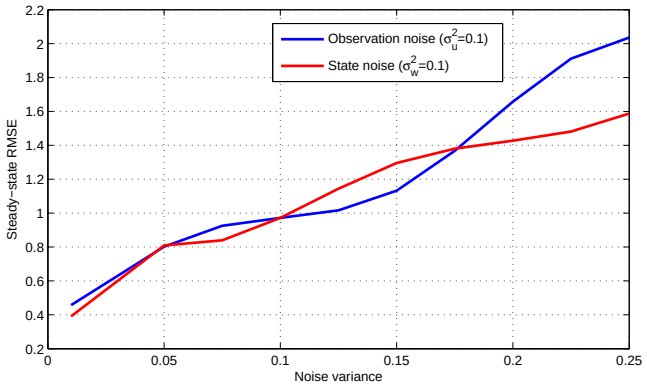

Figure 2.7. Steady-state RMSE versus state/ measurement noise variance..

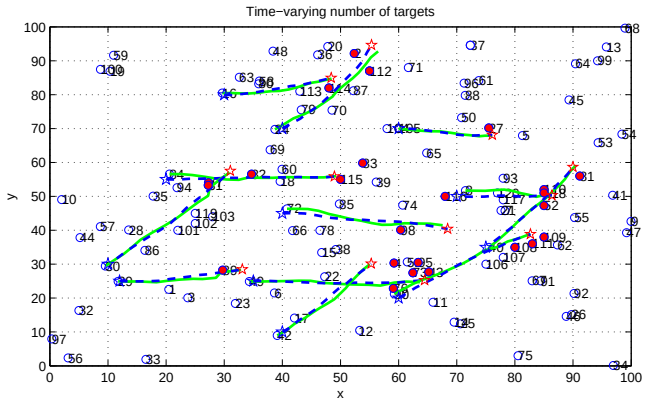

Figure 2.8. Tracking of multiple targets in a setting with time-varying number of targets..

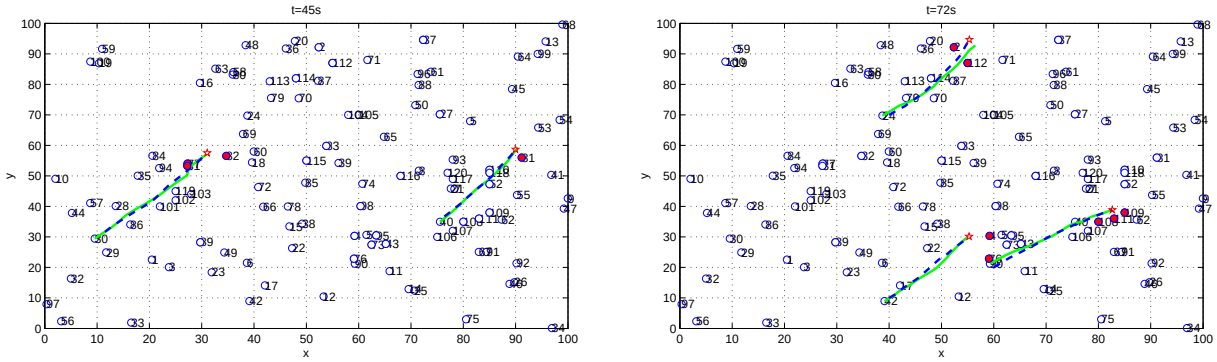

Figure 2.9. Active sensors and position of targets at time instances (left) $t = 45$s and (right) $t = 72$s. .

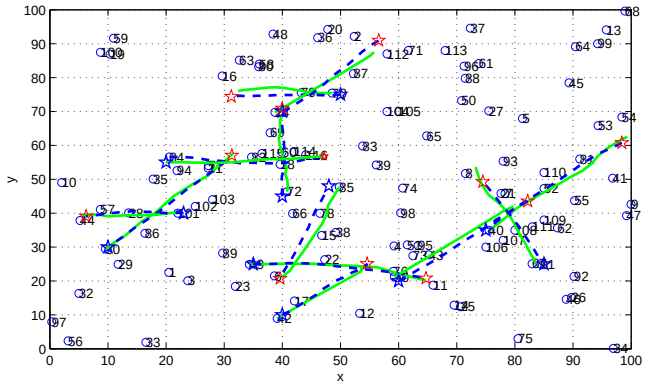

Figure 2.10. Tracking of multiple targets in a setting with time-varying number of targets.
.

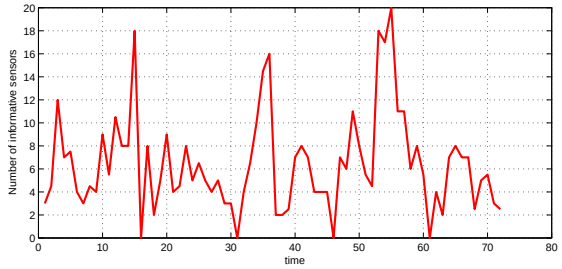

Figure 2.11. Number of target-informative sensors versus time t for the setting in Fig. 8..

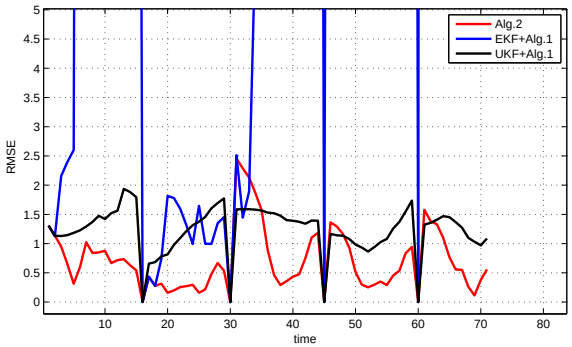

Figure 2.12. Average tracking RMSE versus time in a setting with $R = 12$ targets for the setting in Fig. 8..

CHAPTER 3

Distributed Spatio Time Association and Tracking of Sources in Sensor Network

3.1 Problem Statement

Consider a field sensed by a total of p sensors. Each sensor, say j, acquires scalar measurements $x_j(t)$ at time instant $t = 0, 1, 2, ...$ Sensor observations contain information about r underlying sources $s_\rho(t)$ which are represented by the scalar random variables $s_\rho(t)$, for $\rho = 1, \ldots, r$. Source signals stacked in the state vector $\mathbf{s}_t := [s_1(t) \ldots s_r(t)]^T$ evolve according to the model:

$$\mathbf{s}_t = \mathbf{F}\mathbf{s}_{t-1} + \mathbf{u}_t, \tag{3.1}$$

where $\mathbf{F}$ is the transition matrix, while $\mathbf{u}_t$ is the state noise assumed to be zero-mean Gaussian with covariance $\mathbf{Q} = \text{diag}(\sigma^2_{u,1}, \ldots, \sigma^2_{u,r})$. To ensure the stability of the state vector, the spectral radius of $\mathbf{F}$ is assumed to be less than unity. Further, it is assumed here that the sources are uncorrelated with each other, i.e., $\mathbf{F}$ is a diagonal matrix whose entries are less than one. Matrix $\mathbf{F}$ and the noise variances $\sigma^2_{u,\rho}$ are assumed available here, and can be found from e.g., the physics of the problem.

Each field source is observed by an unknown group of sensors. For instance, in Fig. 3.1 there are $r = 2$ sources in the field. At time instant t_0, sensors S_1, S_2, S_4 (green region) acquire measurements about source $s_1(t_0)$, while source $s_2(t_0)$ is observed by sensors S_8, S_{10} (red region). The rest of the sensors are just observing noise (blue sensors). Each sensor S_j at time instance t acquires a scalar measurement following the model

$$x_j(t) = \sum_{\rho=1}^{r} h_{j,\rho}(t)s_\rho(t) + w_j(t), \tag{3.2}$$

where the coefficients $h_{j,\rho}(t)$ denote the *unknown* channel attenuation coefficient from source $s_\rho(t)$ to sensor S_j at time t. Further, $w_j(t)$ is the corresponding zero-mean Gaussian sensing noise with variance σ_w^2. Such a setting could rise when tracking ground vibrating sources, see e.g., [116].

Specifically, let the ρth source represent a vibrating source while $s_\rho(t)$ corresponds to the vibration signal that the source generates on the ground with $\rho = 1, \ldots, r$. These r vibrating sources could correspond for instance to a moving vehicle, a person, animal or machinery, e.g., see [116]. The vibrating signals $s_\rho(t)$ in fact can be modeled in several scenaria, as autoregressive (AR) processes [see model in (3.1)] as advocated in [138]. Note that AR processes of higher order can be considered after employing vector states for each of the sources. For simplicity in exposition and clarity here AR-1 processes are considered.

Let $\mathbf{p}_j$ denote the fixed position of sensor S_j, while $\mathbf{l}_\rho(t)$ the location of the ρth source for $\rho = 1, \ldots, r$. According to [116] the intensity of the vibrations measured at distance d from the vibrating source are attenuated by a factor $e^{-\alpha d}$, where α a positive constant depending on the soil, e.g., for loess soil is $0.2m^{-1}$ [116]. Applying the superposition principle, and assuming there is one dominant path of signal propagation from the source to the sensor it turns out that sensor S_j acquires the measurement

$$x_j(t) = \sum_{\rho=1}^{r} e^{-\alpha \|\mathbf{p}_j - \mathbf{l}_\rho(t)\|_2} s_\rho(t) + w_j(t). \tag{3.3}$$

It is clear from (3.3) that the channel attenuation coefficients are $h_{j,\rho}(t) = e^{-\alpha \|\mathbf{p}_j - \mathbf{l}_\rho(t)\|_2}$ and depend on the distance of the sensor from the source. Notice that among the summands $e^{-\alpha \|\mathbf{p}_j - \mathbf{l}_\rho(t)\|_2} s_\rho(t)$ some have very small amplitude when sensor S_j is far from source ρ, while the amplitude increases as sensor S_j is located closer to source ρ.

Thus, some of the attenuation coefficients may be close to zero depending on the distance between sources and sensors, e.g., in Fig. 3.1(left), sensor S_2 observes source $s_1(t)$, thus $h_{2,1}(t_0)$ is expected to be nonzero whereas $h_{2,2}(t_0)$ is negligible while sensor S_{10} ob-

serves source $s_2(t)$, thus $h_{10,2}(t_0)$ is expected to be nonzero whereas $h_{10,1}(t_0)$ is negligible. Stacking the sensor measurements in $\mathbf{x}_t := [x_1(t) \dots x_p(t)]^T$ and the corresponding sensing noises in vector $\mathbf{w}_t := [w_1(t), \dots w_p(t)]^T$, the observation model can be summarized as

$$\mathbf{x}_t = \mathbf{H}_t \mathbf{s}_t + \mathbf{w}_t, \tag{3.4}$$

where the observation matrix $\mathbf{H}_t$ contains the unknown attenuation coefficients $\mathbf{H}_t(j, \rho)$ $:= h_{j,\rho}(t)$. Note that the ρth column of the observation matrix $\mathbf{H}_t$ contains the attenuation coefficients between all sensors and source $s_\rho(t)$ at time t. For sensors close to the ρth vibrating source, the corresponding distances, $\|\mathbf{p}_j - \mathbf{l}_\rho(t)\|_2$, will be relatively small, thus leading to relatively large entries $\mathbf{H}_t(j, \rho) = e^{-\alpha\|\mathbf{p}_j - \mathbf{l}_\rho(t)\|_2}$, compared to sensors that are farther away. Since sources at a given instant t are very localized and affect the measurements of a small percentage of sensors located a few meters away, many entries of $\mathbf{H}_t$ will be close to zero giving rise to an approximately sparse $\mathbf{H}_t$.

The field sources may be moving in space which results the time dependency of $\mathbf{H}_t$. As the sources move in the field, different group of sensors will observe them which results time-varying attenuation coefficients, resulting a time-varying $\mathbf{H}_t$. For example, in Fig. 3.1, at time instant t_0 source $s_1(t)$ is sensed by S_1, S_2 and S_4, while at time instant t_1 it is sensed by S_3, S_4. This further implies that the corresponding attenuation coefficients in the corresponding column of $\mathbf{H}_t$ will also change, e.g. in the first column of $\mathbf{H}_{t_0}$, entries $\mathbf{H}_{t_0}(1, 1)$, $\mathbf{H}_{t_0}(2, 1)$ and $\mathbf{H}_{t_0}(4, 1)$ have strong amplitude at t_0, whereas at t_1 they've switched to $\mathbf{H}_{t_1}(3, 1)$ and $\mathbf{H}_{t_1}(4, 1)$. Note that the support (indices of nonzero entries) of each column in $\mathbf{H}_t$, indicate which sensors acquire measurements about the source multiplying that column in (3.4).

Our goal here is to track the field source signals $\{s_\rho(t)\}_{\rho=1}^r$ using only sensor observations and the state model, but without having available the sensing matrix $\mathbf{H}_t$. For

instance in the presence of vibrating sources, tracking the different signals can facilitate applications such as target classification, see e.g. [113, 117]. In the following, novel algorithms will be developed to jointly track the sparse matrix $\mathbf{H}_t$, while performing source-sensor association to track each of the sources using only those observations acquired at sensors observing a source. The proposed scheme will enable multi-source tracking by using only the source-informative sensors. The latter benefits can be seen in Fig. 3.1 in which it is obvious that the blue sensors do not carry any information and therefore there is no need for the FC to communicate with them. The aforementioned problem will be considered in two different sensor network topologies.

First, a fusion-center based topology, as the one depicted in Fig. 3.1, is considered in which every sensor only acquires measurements and if necessary transmits its observation to a fusion center responsible for carrying out multi-source tracking and sources-to-sensors association. The proposed tracking framework will reduce sensor-to-FC communications, respect any power constraints across the network while using measurements from source-informative sensors to carry out the tracking.

The second topology considered, will be an ad hoc sensor network comprising of p sensors. In the ad-hoc topology, sensors not only will acquire measurements but also will be responsible for carrying out information processing in an in-network fashion. Specifically, sensors will only communicate with their single-hop neighboring sensors who are within communication range. For instance, in Fig. 3.2, sensor S_6 is able to communicate with sensors S_3, S_4, S_7 and S_8 which form the single-hop neighborhood for S_6. The neighboring sensors of sensor j are contained in the set $\mathcal{N}_j$, while the sensor network is modeled as an undirected graph and the inter-sensor links are assumed to be symmetric.

The same state and measurement models introduced in (3.1) and (3.4) will be considered here. The goal will be to identify the time-varying groups of sensors observing different sources by recovering the support of the unknown $\mathbf{H}_t$, while tracking $\mathbf{s}_t$. Since there is no

fusion-center, in the ad-hoc setting, sensors will have to collaborate and exchange information with their single-hop neighbors to locally track the sensors and identify the different groups of informative sensors.

Remark: Note that the two different topologies considered here are complimentary, thus it is of interest to consider the problem of multi-source tracking in both the FC-based topology which enables fast processing/tracking since sensors send their data directly to a FC. The ad hoc topology involves iterative algorithms that may take longer time to converge, though this architecture offers robustness to failures and lack of a single point of failure, such as a FC.

3.2 FC-Based Tracking and Sensor Selection

We start with developing a proper joint tracking and source-to-sensor association algorithm for a FC-based setting. A pertinent minimization framework based on a novel blending of Kalman filtering/smoothing and sparsity-aware matrix recovery via norm-one regularization is utilized to jointly recover the sensing matrix $\mathbf{H}$, and track the states $\{\mathbf{s}_\rho(t)\}_{\rho=1}^r$ at the FC.

3.2.1 Kalman Filtering Optimization

Kalman filtering (KF) is an iterative filtering technique which updates the prediction and correction state estimates, namely $\hat{\mathbf{s}}(t|t-1) := E[\mathbf{s}(t)|\mathbf{x}_0,\ldots,\mathbf{x}_{t-1}]$, $\hat{\mathbf{s}}(t|t) := E[\mathbf{s}(t)|\mathbf{x}_0,\ldots,\mathbf{x}_t]$, as well as their corresponding error covariance matrices

$$\begin{aligned}\mathbf{M}_{t|t-1} &:= E[(\mathbf{s}_t-\hat{\mathbf{s}}(t|t-1))(\mathbf{s}_t-\hat{\mathbf{s}}(t|t-1))^T]\\ \mathbf{M}_{t|t} &:= E[(\mathbf{s}_t-\hat{\mathbf{s}}(t|t))(\mathbf{s}_t-\hat{\mathbf{s}}(t|t))^T]\end{aligned} \tag{3.5}$$

by utilizing all data acquired up to time t. Detailed KF recursions can be found in [63](Ch. 12). If the model parameters in (3.1) and (3.4) are available, the Kalman estimates for $K+1$

(where $K \geq 1$) consecutive states, say $\{\mathbf{s}_{t-K}, \ldots, \mathbf{s}_t\}$, can be formulated as the minimizer, see e.g., [4], of

$$\{\check{\mathbf{s}}_\tau\}_{\tau=t-K}^{t} = \arg \min_{\mathbf{s}_{t-K},\ldots,\mathbf{s}_t} J_{t,K}(\{\mathbf{s}_\tau, \mathbf{H}_\tau\}_{\tau=t-K}^{t}) \tag{3.6}$$

where

$$\begin{aligned} J_{t,K}(\{\mathbf{s}_\tau, \mathbf{H}_\tau\}_{\tau=t-K}^{t}) :=& \sum_{\tau=t-K+1}^{t} (\mathbf{x}_\tau - \mathbf{H}_\tau \mathbf{s}_\tau)^T \mathbf{\Sigma}_w^{-1} (\mathbf{x}_\tau - \mathbf{H}_\tau \mathbf{s}_\tau) \\ &+ \sum_{\tau=t-K+1}^{t} (\mathbf{s}_\tau - \mathbf{F}\mathbf{s}_{\tau-1})^T \mathbf{Q}^{-1} (\mathbf{s}_\tau - \mathbf{F}\mathbf{s}_{\tau-1}) \\ &+ (\mathbf{s}_{t-K} - \hat{\mathbf{s}}_{t-K})^T \mathbf{M}_{t-K|t-K}^{-1} (\mathbf{s}_{t-K} - \hat{\mathbf{s}}_{t-K}), \end{aligned} \tag{3.7}$$

where $\mathbf{\Sigma}_w := \sigma_w^2 \cdot \mathbf{I}$ contains the sensing noise variances and $\hat{\mathbf{s}}_{t-K}$ corresponds to an estimate of the state $\mathbf{s}_{t-K}$ and $\mathbf{M}_{t-K|t-K}$ the corresponding covariance at time instant $t-K$ using sensor data $\mathbf{x}_0, \ldots, \mathbf{x}_{t-K}$.

Traditionally, see e.g., [63], the cost in (3.7) is minimized w.r.t the states assuming that $\mathbf{H}_\tau$ is given. If $K = 1$, the standard KF recursions are obtained by minimizing (3.7), namely $\check{\mathbf{s}}_\tau = \hat{\mathbf{s}}[\tau|\tau]$. Notice that when $K > 1$, the estimates $\check{\mathbf{s}}_t$ correspond to the state estimates obtained by a fixed-lag (K lags here) Kalman smoother, see e.g., [4, pg. 177], i.e., $\check{\mathbf{s}}_\tau = \hat{\mathbf{s}}(\tau|t)$ for $\tau = t-K, \ldots, t$.

Assuming that the sensing matrix $\mathbf{H}_t$ is available, the minimization problem is performed with respect to the unknown sources' states $\{\mathbf{s}_\tau\}_{\tau=t-K}^{t}$. However, the sensing matrix itself is unknown in the considered setting and need to be found as well. The formulation in (3.7) does not account for the sparse unknown structure of the sensing matrix $\mathbf{H}_t$.

To this end, and inspired by existing work in ℓ_1-regularization, see e.g., [114], we introduce a norm-one regularization term in (3.7) and obtain the minimization formulation

$$\begin{aligned}\min_{\{\mathbf{s}_\tau\}_{\tau=t-K}^{t},\mathbf{H}_\tau} &\sum_{\tau=t-K+1}^{t}(\mathbf{x}_\tau-\mathbf{H}_\tau\mathbf{s}_\tau)^T\mathbf{\Sigma}_w^{-1}(\mathbf{x}_\tau-\mathbf{H}_\tau\mathbf{s}_\tau)+\textstyle\sum_{\rho=1,\tau=t-K+1}^{r,t}\lambda_\rho\|\mathbf{h}_{\tau,\rho}\|_1\\ &+\sum_{\tau=t-K+1}^{t}(\mathbf{s}_\tau-\mathbf{F}\mathbf{s}_{\tau-1})^T\mathbf{Q}^{-1}(\mathbf{s}_\tau-\mathbf{F}\mathbf{s}_{\tau-1})\\ &+(\mathbf{s}_{t-K}-\hat{\mathbf{s}}_{t-K})^T\mathbf{M}_{t-K|t-K}^{-1}(\mathbf{s}_{t-K}-\hat{\mathbf{s}}_{t-K}),\end{aligned}\tag{3.8}$$

where $\|\cdot\|_1$ denotes the one-norm, while $\mathbf{h}_{\tau,\rho}$ corresponds to the ρth column of matrix $\mathbf{H}_\tau$. The weights λ_ρ are nonnegative sparsity-controlling coefficients that control the number of nonzeros in the column of $\mathbf{h}_{\tau,\rho}$. Tackling the optimization problem in (3.8) will allow i) recover the missing observation matrix $\mathbf{H}_\tau$ and perform sources-to-sensors association; ii) track the unknown r sources present in the monitored field.

3.2.2 Multi-source State Tracking

Aiming at minimizing the cost in (3.8), we resort to block coordinate descent technique where we minimize (3.8) wrt the states $\{\mathbf{s}_\tau\}_{\tau=t-K}^{t}$ by fixing the matrices $\mathbf{H}_\tau$ to their most recent update. Further, to simplify (3.8), the time index of $\mathbf{H}_t$ is removed by assuming that $\mathbf{H}$ is varying very slowly during K consecutive time instances. Then, the cost is minimized wrt $\mathbf{H}_t$ while fixing the states to their most up-to-date values. This section focuses on updating the states while fixing $\mathbf{H}$.

Denote $\hat{\mathbf{H}}_{t-1}$ as the estimate for $\mathbf{H}$ obtained during the time interval $[t-K-1, t-1]$. By fixing $\mathbf{H}$ to $\hat{\mathbf{H}}_{t-1}$, the cost in (3.8) is optimized w.r.t the states $\{\mathbf{s}_\tau\}_{\tau=t-K}^{t}$. After differentiating the cost in (3.8) w.r.t the augmented state vector $\mathbf{s}_{\alpha,t} := [\mathbf{s}_{t-K}^T \dots \mathbf{s}_t^T]^T$ and setting the derivative equal zero, we will get the following state estimates (See Apdx. C):

$$\check{\mathbf{s}}_{\alpha,t} := [\check{\mathbf{s}}_{t-K,t}^T,\dots,\check{\mathbf{s}}_{t,t}^T]^T = \left(\mathbf{F}_\alpha^T\mathbf{Q}_\alpha^{-1}\mathbf{F}_\alpha\right)^{-1}\mathbf{F}_\alpha^T\mathbf{Q}_\alpha^{-1}\mathbf{x}_{\alpha,t}\tag{3.9}$$

where

$$\mathbf{Q}_{\alpha,t} := \text{bdiag}(\mathbf{M}_{t-K|t-K}, \overbrace{\mathbf{Q}, \ldots, \mathbf{Q}}^{K\text{times}}, \overbrace{\mathbf{\Sigma}_w, \ldots, \mathbf{\Sigma}_w}^{K\text{times}}), \tag{3.10}$$

$$\mathbf{x}_{\alpha,t} := [-\hat{\mathbf{s}}_{t-K}^T, \mathbf{0}^T, \ldots, \mathbf{0}^T, \mathbf{x}_{t-K+1}^T, \ldots, \mathbf{x}_t^T]^T \tag{3.11}$$

The state estimate $\hat{\mathbf{s}}_{t-K}$ is provided by a local Kalman filter run at the FC with the estimated $\hat{\mathbf{H}}_t$ and the available sensor measurements in $\mathbf{x}_{t-K}$. Matrices $\mathbf{Q}_{\alpha,t}$ and $\mathbf{x}_{\alpha,t}$ have sizes $[Kp + (K+1)r] \times [Kp + (K+1)r]$ and $[(K+1)r + Kp] \times 1$ respectively, while matrix $\mathbf{F}_\alpha := [\mathbf{F}_1^T \;\; \mathbf{F}_{2,t}^T]^T$ has size $(K+1)r + Kp \times (K+1)r$ and formed as:

$$\mathbf{F}_1 = \begin{bmatrix} -\mathbf{I}_{r\times r} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{F} & -\mathbf{I}_{r\times r} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{F} & -\mathbf{I}_{r\times r} \end{bmatrix}, \; \mathbf{F}_{2,t} = \begin{bmatrix} \mathbf{0} & \hat{\mathbf{H}}_{t-1} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{H}}_{t-1} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \hat{\mathbf{H}}_{t-1} \end{bmatrix}. \tag{3.12}$$

It is worth mentioning that matrix $(\mathbf{F}_\alpha^T \mathbf{Q}_\alpha^{-1} \mathbf{F}_\alpha)^{-1}$ is block tri-diagonal. In order to solve the system of linear equations in (3.9), there exist efficient approaches, see e.g. [44, pg. 174] whose computational complexity is linearly increasing with K.

3.2.3 Sparse Sensing Matrix Recovery

As stated earlier, the task of performing real-time sources-to-sensors association turns out to boil down to the problem of recovering the support of the sensing matrix $\mathbf{H}_t$. To this end, the cost (3.8) will be minimized wrt to $\mathbf{H}$ by fixing the states to their latest updated estimates $\{\check{\mathbf{s}}_{\tau,t}\}_{\tau=t-K}^t$ obtained in (3.9). A coordinate descent way is utilized to minimize (3.8) wrt one entry of $\mathbf{H}$ while fixing the other terms to their most recent values. At time t, let $\hat{\mathbf{H}}_t^\varkappa(i,\rho)$ denote the coordinate update for entry $\mathbf{H}(i,\rho)$ obtained during coordinate cycle $\varkappa$ for all $i = 1, \ldots, p$ and $\rho = 1, \ldots, r$. When updating $\mathbf{H}(i,j)$, all remaining

entries of $\mathbf{H}$ are fixed to the previous coordinate cycle updates, namely $\hat{\mathbf{H}}_t^{\varkappa-1}(i,\ell)$, and as shown in Apdx. D $\hat{\mathbf{H}}_t^{\varkappa}(i,\rho)$ is obtained as:

$$\hat{\mathbf{H}}_t^{\varkappa}(i,\rho) := \arg\min_h \sigma_w^{-2}\|\mathbf{y}_{i,t}^{\varkappa-1} - \check{\mathbf{s}}_{\rho,t}\cdot h)\|_2^2 + \lambda_\rho|h|, \tag{3.13}$$

where $\mathbf{y}_{i,t}^{\varkappa-1}$ is a $K\times 1$ vector with entries

$$\mathbf{y}_{i,t}^{\varkappa-1}(m) = x_i(t-K+m) - \sum_{\ell=1,\ell\neq\rho}^{r} \hat{\mathbf{H}}_t^{\varkappa-1}(i,\ell)\check{\mathbf{s}}_{t-K+m,t}(\ell) \tag{3.14}$$

for $m=1,\ldots,K$ and $\check{\mathbf{s}}_{\rho,t} := [\check{\mathbf{s}}_{t-K+1,t}(\rho),\ldots,\check{\mathbf{s}}_{t,t}(\rho)]^T$. The minimization in (3.13) boils down to a sparse regression (Lasso) problem involving a scalar. It turns out (see Apdx. D) that the minimizer in (3.13) is:

$$\hat{\mathbf{H}}_t^{\varkappa}(i,\rho) = \text{sgn}(\check{\mathbf{s}}_{\rho,t}^T\mathbf{y}_{i,t}^{\varkappa-1})\cdot\left(\frac{\check{\mathbf{s}}_{\rho,t}^T\mathbf{y}_{i,t}^{\varkappa-1}}{\|\check{\mathbf{s}}_{\rho,t}\|_2^2} - \frac{\lambda_\rho}{2\sigma_w^{-2}\|\check{\mathbf{s}}_{\rho,t}\|_2^2}\right)_+ \tag{3.15}$$

where $(z)_+ = \max(z,0)$. Note that warm starts are employed at every time instant t, where $\hat{\mathbf{H}}_t^0(i,\ell) = \hat{\mathbf{H}}_{t-1}^C(i,\ell)$ for $i=1,\ldots,p$ and $\ell=1,\ldots,r$. C is the number of coordinate cycles applied at every time instant t. A stopping criterion applied for the coordinate method is the following: For two consecutive updating cycles, a cost difference, using the cost in (3.8), is evaluated. If the cost difference becomes lower than a predefined threshold, say ϵ, then the updating process is terminated automatically.

3.2.4 Transmission Power Budget Constraints

It is often the case that sensors have to operate under a stringent power budget, imposed here in the form of a constraint in the total transmission power that all sensors can consume. Thus, it is of interest to introduce in the formulation in (3.8) a mechanism respecting such a power constraint.

Starting from the formulation at (3.8), we introduce in the cost in (3.8) selection variables to decide which sensors' data participate, or not, while ensuring the total transmission power does not exceed a desired level P_{tot}. In detail:

$$\begin{aligned} &\arg \min_{\mathbf{s}_{t-K},\ldots,\mathbf{s}_t} J_{t,K}(\mathbf{D}, \{\mathbf{s}_\tau\}_{\tau=t-K}^{t}, \mathbf{H}) \\ &\text{s. to } \sum_{j=1}^{p} d_j \sigma_{x,j}^2 < P_{tot}, \text{ and } d_j \in [0,1], \quad j = 1, \ldots, p \end{aligned} \tag{3.16}$$

where

$$\begin{aligned} J_{t,K}(\mathbf{D}, \{\mathbf{s}_\tau\}_{\tau=t-K}^{t}, \mathbf{H}) :=& \sum_{\tau=t-K+1}^{t} (\mathbf{x}_\tau - \mathbf{H}\mathbf{s}_\tau)^T \mathbf{D} \boldsymbol{\Sigma}_w^{-1} (\mathbf{x}_\tau - \mathbf{H}\mathbf{s}_\tau) \\ &+ \sum_{\tau=t-K+1}^{t} (\mathbf{s}_\tau - \mathbf{F}\mathbf{s}_{\tau-1})^T \mathbf{Q}^{-1} (\mathbf{s}_\tau - \mathbf{F}\mathbf{s}_{\tau-1}) \\ &+ (\mathbf{s}_{t-K} - \hat{\mathbf{s}}_{t-K})^T \mathbf{M}_{t-K|t-K}^{-1} (\mathbf{s}_{t-K} - \hat{\mathbf{s}}_{t-K}) + \textstyle\sum_{\rho=1}^{r} \lambda_\rho \|\mathbf{h}_\rho\|_1, \end{aligned} \tag{3.17}$$

where $\boldsymbol{\Sigma}_w := \sigma_w^2 \cdot \mathbf{I}_{p\times p}$, while $\hat{\mathbf{s}}_{t-K}$ corresponds to an estimate of the state $\mathbf{s}_{t-K}$ and $\mathbf{M}_{t-K|t-K}$ denotes the corresponding covariance acquired as described in Sec. 3.2.2. $\mathbf{D}$ is the diagonal matrix whose diagonal entries d_j lie in the interval $[0, 1]$, and control which sensor measurements will participate in tracking the source states in $\mathbf{s}_t$ during the interval $[t-K+1, t]$. $\mathbf{D}$ will be selected such that the prediction error is minimized while ensuring that the transmission power will not exceed P_{tot}. Note that signal variance in a way quantifies the power of the signal and therefore the amount of power needed to transmit it, thus the summation $\sum_{j=1}^{P} d_j \sigma_{x,j}^2$ quantifies the average power needed to transmit data from the selected sensors to the FC. However, it should be emphasized that this constraint does not account for communication parameters such as the channel.

We resort to block coordinate descent techniques, see e.g. [10], where we minimize (3.17) w.r.t. the entries of $\mathbf{D}$, or $\mathbf{H}$ or $\{\mathbf{s}_\tau\}_{\tau=t-K+1}^{t}$ while fixing the others to their latest

update. Fixing the state vectors and $\mathbf{H}$ to their latest updates, the minimization of (3.17) w.r.t. the diagonal coefficients d_j in $\mathbf{D}$ corresponds to

$$\{\hat{d}_{1,t}, \hat{d}_{2,t}, \ldots, \hat{d}_{p,t}\} := \arg\min \sum_{\tau=t-K+1}^{t} \sum_{j=1}^{p} d_j \sigma_w^{-2} [x_j(\tau) - [\hat{\mathbf{H}}_{t-1}]_{j:} \check{\mathbf{s}}_{\tau,t-1}]^2$$

$$\text{s. to} \sum_{j=1}^{p} d_j \sigma_{x,j}^2 < P_{tot}, \text{ and } d_j \in [0,1], \quad j = 1, \ldots, p \tag{3.18}$$

where the sensor observation variances can be estimated as

$$(\hat{\sigma}_{x,1,t-1}^2, \ldots, \hat{\sigma}_{x,\rho,t-1}^2) = \text{diag}(\hat{\mathbf{H}}_{t-1} \hat{\mathbf{\Sigma}}_s \hat{\mathbf{H}}_{t-1}^T), \tag{3.19}$$

while the source covariance estimate $\hat{\mathbf{\Sigma}}_s$ can be estimated as $\hat{\mathbf{\Sigma}}_s = \hat{\mathbf{s}}_{\tau,t-1} \cdot \hat{\mathbf{s}}_{\tau,t-1}^T$. Further, $[\hat{\mathbf{H}}_{t-1}]_{j:}$ denotes the jth row of $\hat{\mathbf{H}}_{t-1}$, while $\check{\mathbf{s}}_{\tau,t-1}$ corresponds to the state estimates obtained from minimizing the cost $J_{t-1,K}$, and $\hat{\mathbf{H}}_{t-1}$ is the sensing matrix estimate obtained during time interval $[t-K, t-1]$. The minimization problem in (3.18) corresponds to a linear program with linear constraints and can be solved, e.g., using the interior point method, see e.g. [16]. After the coefficients $\hat{d}_{j,t}$ are determined in (3.18) for the time interval $[t-K+1, t]$, they are compared with a threshold. If $\hat{d}_{j,t}$ is larger than the threshold, then it is set to $\check{d}_{j,t} = 1$ and the FC pings the corresponding sensor to transmit its measurements, otherwise $\hat{d}_{j,t}$ is set to $\check{d}_{j,t} = 0$ and the corresponding sensor remains silent. The threshold is selected as the smallest value in the interval $(0.5, 1)$ such that the total power consumed by the selected sensors, namely $\sum_{j=1}^{p} (\check{d}_{j,t} \sigma_{x_j}^2)$, does not exceed the power budget P_{tot}. The left end of the interval 0.5 is set such that any sensor j with $\hat{d}_{j,t} > 0.5$ is considered a candidate sensor for being selected. Further, the smallest value in the interval $(0.5, 1)$ allows the largest number of sensors to be selected while not exceeding the power budget P_{tot}.

Once $\hat{d}_{j,t}$ are determined and set to 1 or 0 as described earlier, estimates of the sources' state vectors $\{\mathbf{s}_\tau\}_{\tau=t-K+1}^{t}$, and recovery of the sparse $\mathbf{H}_t$ can be done in the same way as described in Section 3.2. Note that since some $\check{d}_{j,t}$ are zero, corresponding sum-

mands in the first term of (3.17) will not be participating in the updating of the state vectors and $\mathbf{H}$ via (3.9) and (3.15), respectively.

3.2.5 Algorithmic Summary

To initialize the matrix $\mathbf{H}_t$, at time instant $t = 0$, and obtain an estimate of the number of sources in the monitored field, the sparse decomposition scheme in [101], summarized in Sec. 3.2.6, can be applied. To collect a sufficiently large number of data while ensuring that source states do not change significantly, as required to apply [101], a sufficiently fast sampling rate can be applied at a start-up phase to ensure the aforementioned requirement. The processing takes place at the FC which is responsible for tracking $\{d_{j,t}\}_{j=1}^{p}$, $\mathbf{H}_t$ and $\mathbf{s}_t$. Once the sensor selection variables $\hat{d}_{j,t}$ are found via solving (3.18), they are compared with a threshold. Then, depending on whether $\hat{d}_{j,t}$ is larger or smaller than the threshold, is set to $\check{d}_{j,t} = 1$ or $\check{d}_{j,t} = 0$, respectively. Then, the state vectors $\{\mathbf{s}_\tau\}_{\tau=t-K+1}^{t}$ and $\mathbf{H}_t$ are estimated, by solving the costs in (3.17) after keeping only these summands within the first term of (3.17) with index j for which $\check{d}_{j,t} = 1$.

At the start-up stage all sensors transmit their data to the FC such that the state vectors, $\mathbf{H}_0$ and $d_{j,0}$'s are initialized. Then, throughout the execution stage of the algorithm, only the sensors with $\check{d}_{j,t} = 1$ will transmit their data into the FC. This further implies that only the corresponding rows of $\mathbf{H}_t$, namely $[\hat{\mathbf{H}}_t]_{j:}$ for which $\{j : \check{d}_{j,t} = 1\}$ will be updated.

Since the sources may be moving, there is a need to update $\hat{d}_{j,t}$ on a continuous basis. The approach followed here is to re-update the weights using the start-up phase steps every, say Ω, time instances. For slowly varying sources, the time interval separating two updates of the d_j weights can be increased and the reconfiguration step does not have to be applied very frequently. Thus, a small percentage of sensors will be sending data to the FC most of the time. In the simulation we quantify the percentage of active sensors and the associated tracking error. In the FC-based architecture, sensors are just sensing measurements and

get pinged (if $\check{d}_{j,t} = 1$) by the FC to transmit their data. Note that if $P_{tot} > \sum_{j=1}^{p} \sigma_{x,j}^2$, then all d_j's can be 1 and the sensors selection step is not really needed. The approach is summarized in the Algorithm 1 Table.

3.2.6 Sparsity-aware Matrix Decomposition via [101]

The scheme in [101] focuses on estimating the sparse observation matrix $\mathbf{H}_0$ in (3.4) at $t = 0$, by decomposing into sparse factors the data covariance matrix

$$\mathbf{\Sigma}_x = \mathbf{H}_0 \mathbf{\Sigma}_s \mathbf{H}_0^T + \sigma_w^2 \mathbf{I}_p = \bar{\mathbf{H}}_0 \bar{\mathbf{H}}_0^T + \sigma_w^2 \mathbf{I}_p, \tag{3.20}$$

where $\mathbf{\Sigma}_s$ corresponds to the diagonal covariance of the entries in the state vector $\mathbf{s}_0$, while $\bar{\mathbf{H}}_0 := \mathbf{H}_0 \mathbf{\Sigma}_s^{1/2}$. Further, let

$$\mathbf{M}_x := \mathbf{\Sigma}_x - \sigma_w^2 \mathbf{I}_p = \bar{\mathbf{H}}_0 \bar{\mathbf{H}}_0^T \tag{3.21}$$

which corresponds to the signal covariance matrix after removing the impact of the noise covariance from the data covariance matrix. Note that the support of $\mathbf{H}$ and $\bar{\mathbf{H}}$ is the same since $\mathbf{\Sigma}_s$ is diagonal in structure. The covariance $\mathbf{M}_x$ at $t = 0$ can be estimated by employing fast sampling during the initialization stage to ensure that the source state is essentially time-invariant, and then employing sample-averaging of the acquired measurements and removal of the noise variance. Let $\hat{\mathbf{M}}_x = \hat{\mathbf{\Sigma}}_x - \sigma_w^2 \mathbf{I}$ be the estimator of $\mathbf{M}_x$, where $\hat{\mathbf{\Sigma}}_x$ corresponds to the sample-average covariance estimate.

An estimate of the sparse matrix $\mathbf{H}_0$ is found in [101] by resorting to the sparsity-inducing minimization formulation

$$\hat{\mathbf{H}}_0 = \arg\min_{\mathbf{H}} \|\hat{\mathbf{M}}_x - \mathbf{H}\mathbf{H}^T\|_F^2 + \sum_{\rho=1}^{r} \lambda_\rho \|\mathbf{h}_\rho\|_1 \tag{3.22}$$

where $\|\cdot\|_1$ refers to norm-one and λ_ρ is the sparsity-controlling coefficients for the ρth column of matrix $\mathbf{H}$, namely $\mathbf{h}_\rho$. The minimization problem in (3.22) is tackled via a

coordinate decent approach where the cost in (3.22) is cyclically minimized with respect to an entry of $\mathbf{H}$ while keeping the rest entries fixed. Details on the recursive updating formulas involved in minimizing (3.22) can be found in [101].

3.3 Distributed Tracking and Sensor Selection

Our goal is to develop a joint multi-source association and tracking algorithm that is distributed, and can be implemented in an ad hoc network of spatially distributed sensors. Sensors placed in an ad hoc configuration can both acquire and process data.

3.3.1 Separable Minimization

Since there is not a central FC in the ad hoc setting, there is a need to rewrite the centralized cost in (3.7) in a way that takes into account the network topology. The reason for this step is to derive subsequently an algorithm that involves information exchanges only between single-hop neighbors that are sufficiently close to communicate. Toward this end, the cost in (3.7) will be splitted into local subcosts which can be tackled in parallel and locally across sensors, while proper equality constraints will be used to account for the network topology (communication graph).

Local vectors $\mathbf{s}_{\tau,j}$ for $j = 1, \ldots, p$ are introduced to indicate local optimization variables stored and updated at sensor j. The vectors $\mathbf{s}_{\tau,j}$ corresponds to a local version of state vector $\mathbf{s}_\tau$, at sensor j. The local variables $\mathbf{s}_{\tau,j}$ are introduced to allow every sensor j to keep tracking $\mathbf{s}_\tau$. Since all the vectors $\{\mathbf{s}_{\tau,j}\}_{j=1}^p$ correspond to local copies of $\mathbf{s}_\tau$, they should be equal. Thus, the equality constraint $\mathbf{s}_{\tau,1} = \mathbf{s}_{\tau,2} = \ldots = \mathbf{s}_{\tau,p}$ needs to accompany a minimization formulation that will rely on the local vectors $\mathbf{s}_{\tau,j}$.

However, such a constraint is challenging to satisfy since it requires data communication between any possible pair of sensors in the network (full connectivity). In order to comply with the network topology, each local state $\mathbf{s}_{\tau,j}$ is required to be equal to

the local state vectors update in its neighborhood $\mathcal{N}_j$, i.e., $\mathbf{s}_{\tau,j} = \mathbf{s}_{\tau,j'}$ for every sensor $j' \in \mathcal{N}_j$. Connectivity of the network communication graph ensures that the previous single-neighborhood constraints guarantee that $\mathbf{s}_{\tau,1} = \mathbf{s}_{\tau,2} = \ldots = \mathbf{s}_{\tau,p}\ \forall \tau = 0, 1, 2, \ldots$ Then, the cost in (3.8) can be equivalently rewritten as:

$$\begin{aligned} J_{t,K}(\{\mathbf{s}_{\tau,j}\}_{\tau=t-K}^{t}, \mathbf{H}) := &\sum_{j=1}^{p} \sum_{\tau=t-K+1}^{t} (x_j(\tau) - \mathbf{H}_{j:}\mathbf{s}_{\tau,j})^2 \sigma_w^{-2} \\ &+ \frac{1}{p} \sum_{j=1}^{p} \sum_{\tau=t-K+1}^{t} (\mathbf{s}_{\tau,j} - \mathbf{F}\mathbf{s}_{\tau-1,j})^T \mathbf{Q}^{-1} (\mathbf{s}_{\tau,j} - \mathbf{F}\mathbf{s}_{\tau-1,j}) \\ &+ \frac{1}{p} \sum_{j=1}^{p} (\mathbf{s}_{t-K,j} - \hat{\mathbf{s}}_{t-K,j})^T \mathbf{M}_{t-K|t-K,j}^{-1} (\mathbf{s}_{t-K,j} - \hat{\mathbf{s}}_{t-K,j}) + \textstyle\sum_{\rho=1}^{r} \lambda_\rho \|\mathbf{h}_\rho\|_1, \end{aligned} \tag{3.23}$$

The cost in (3.23) can be further rewritten as (details in Apdx. E):

$$J_{t,K}(\{\mathbf{s}_{\tau,j}\}_{\tau=t-K}^{t}, \mathbf{H}) := \sum_{j=1}^{p} \sum_{\tau=t-K}^{t} \|\boldsymbol{\xi}_{\tau,j} - \mathbf{G}_{\tau,j}\mathbf{s}_{\tau,j}\|_2^2 + \textstyle\sum_{\rho=1}^{r} \lambda_\rho \|\mathbf{h}_\rho\|_1 \tag{3.24}$$

where

$$\xi_1(\tau) = \begin{cases} 1 & \tau > t-K \\ & \\ 0 & \tau = t-K \end{cases}, \quad \text{and } \xi_2(\tau) = \begin{cases} 0 & \tau > t-K \\ & \\ 1 & \tau = t-K \end{cases},$$

while

$$\boldsymbol{\xi}_{\tau,j} = \begin{bmatrix} \xi_1(\tau)\sigma_w^{-1} x_j(\tau) \\ \xi_1(\tau)(p\mathbf{Q})^{-\frac{1}{2}} \mathbf{F}\mathbf{s}_{\tau-1,j} \\ \xi_2(\tau) p^{-1/2} \mathbf{M}_{t-K|t-K,j}^{-\frac{1}{2}} \hat{\mathbf{s}}_{t-K,j} \end{bmatrix} \text{ and } \mathbf{G}_{\tau,j} = \begin{bmatrix} \xi_1(\tau)\sigma_w^{-1}\mathbf{H}_{j:} \\ \xi_1(\tau)(p\mathbf{Q})^{-\frac{1}{2}} \\ \xi_2(\tau) p^{-1/2} \mathbf{M}_{t-K|t-K,j}^{-\frac{1}{2}} \end{bmatrix}.$$

The state estimate $\hat{\mathbf{s}}_{t-K,j}$ can be updated at each sensor by running a local Kalman filter utilizing its neighboring sensors' measurements along with its own measurement at $t-K$, while $\hat{\mathbf{M}}_{t-K|t-K,j}$ denotes the corresponding local covariance matrix. After incorporating

the local equality constraints introduced earlier, the minimization formulation in (3.8) is reformulated in a separate way as follows:

$$\arg \min_{\{\mathbf{s}_{\tau,j}\}_{\tau=t-K,j=1}^{t,p},\mathbf{H}} \sum_{j=1}^{p} \sum_{\tau=t-K}^{t} \|\boldsymbol{\xi}_{\tau,j} - \mathbf{G}_{\tau,j}\mathbf{s}_{\tau,j}\|_2^2 + \textstyle\sum_{\rho=1}^{r} \lambda_\rho \|\mathbf{h}_\rho\|_1,$$

$$\text{s. to } \mathbf{s}_{\tau,j} = \mathbf{s}_{\tau,j'}, \text{ for } j' \in \mathcal{N}_j,\ j = 1,\ldots,p \text{ and } \tau = t-K,\ldots,t \tag{3.25}$$

As in the FC-based setting, the time index of the sensing matrix $\mathbf{H}$ has been removed, since it is treated fixed during the time window $[t-K, t]$. Note that the minimization formulation in (3.8) and (3.25) are equivalent in the sense that they have the same minimizers for the state vectors and $\mathbf{H}$.

In order to perform joint multi-source tracking and source-to-sensor association, it is required to solve (3.25) in a distributed fashion. An alternating scheme will be employed, where we optimize (3.25) w.r.t the entries of $\mathbf{H}$ while fixing the state, and vice versa.

3.3.2 Algorithmic Construction

To minimize (3.25) in a fully distributed fashion w.r.t the state vectors $\mathbf{s}_{\tau,j}$, we employ the alternating direction method of multipliers (ADMM), see e.g., [17] [15]. To facilitate applicability of ADMM, we introduce a set of auxiliary variables $\mathbf{z}_{\tau,j}^{j'}$ for $j' \in \mathcal{N}_j$ and $\tau = t-K, ..., t$. These auxiliary variables will be used to derive the local updating recursions across sensors and will finally be eliminated. These variables are used to substitute the constraints in (3.25) with the equivalent ones:

$$\mathbf{s}_{\tau,j} = \mathbf{z}_{\tau,j}^{j'} \text{ and } \mathbf{s}_{\tau,j} = \mathbf{z}_{\tau,j'}^{j},\ \text{for } j' \in \mathcal{N}_j \text{ and } j \neq j' \tag{3.26}$$

Necessary in utilizing the ADMM to tackle (3.25) is the augmented Lagrangian function, see, e.g. [17, Chapt 3], which here takes the form

$$\begin{aligned}\mathcal{L}_a[\{\mathbf{s}_{\tau,j}\},\mathbf{v},\boldsymbol{\omega}] = &\sum_{j=1}^{p}\sum_{\tau=t-K}^{t}\|\boldsymbol{\xi}_{\tau,j}-\mathbf{G}_{\tau,j}\mathbf{s}_{\tau,j}\|_2^2+\textstyle\sum_{\rho=1}^{r}\lambda_\rho\|\mathbf{h}_\rho\|_1\\ &+\sum_{j=1}^{p}\sum_{j'\in\mathcal{N}_j}\sum_{\tau=t-K}^{t}[(\mathbf{v}_{\tau,j}^{j'})^T(\mathbf{s}_{\tau,j}-\mathbf{z}_{\tau,j}^{j'})+(\boldsymbol{\omega}_{\tau,j}^{j'})^T(\mathbf{s}_{\tau,j}-\mathbf{z}_{\tau,j'}^{j})]\\ &+0.5c\sum_{j=1}^{p}\sum_{j'\in\mathcal{N}_j}\sum_{\tau=t-K}^{t}[\|\mathbf{s}_{\tau,j}-\mathbf{z}_{\tau,j}^{j'}\|_2^2+\|\mathbf{s}_{\tau,j}-\mathbf{z}_{\tau,j'}^{j}\|_2^2] \qquad (3.27)\end{aligned}$$

The Lagrangian multipliers $\mathbf{v}_{\tau,j}^{j'}$'s and $\boldsymbol{\omega}_{\tau,j}^{j'}$ correspond to the constraints $\mathbf{s}_{\tau,j}=\mathbf{z}_{\tau,j}^{j'}$ and $\mathbf{s}_{\tau,j}=\mathbf{z}_{\tau,j'}^{j}$, respectively. Further, c is a positive coefficient, while the vectors $\mathbf{v}$ and $\boldsymbol{\omega}$ contain the multipliers $\mathbf{v}_{\tau,j}^{j'}$ and $\boldsymbol{\omega}_{\tau,j}^{j'}$ that will make sure that the equality constraints $\mathbf{s}_{\tau,j}=\mathbf{z}_{\tau,j}^{j'}$ and $\mathbf{s}_{\tau,j}=\mathbf{z}_{\tau,j'}^{j}$ are satisfied when minimizing (3.25).

Due to the nonconvex structure of the cost in (3.27), we employ block coordinate descent. First, (3.27) is minimized w.r.t $\mathbf{H}$ while treating the states $\mathbf{s}_{\tau,j}$ as fixed. Then, after fixing $\mathbf{H}$ the cost in (3.27) will be minimized w.r.t $\mathbf{s}_{\tau,j}$ subject to the constraints (3.26).

Minimization of (3.27) **w.r.t H:**

Let $k=0,1,2,\ldots,T$ denote the coordinate descent cycle index when tackling (3.24). Every sensor j is responsible for updating the jth row of $\mathbf{H}$, namely the entries $\{\mathbf{H}(j,\rho)\}_{\rho=1}^{r}$. To this end, the local states $\mathbf{s}_{\tau,j}$ will be set to their most recent updates $\hat{\mathbf{s}}_{\tau,j}^{k+1}$ which corresponds to the ADMM outcome at the end of coordinate cycle $k+1$ (details later). From the augmented Lagrangian function in (3.24) it turns out (using similar arguments as in Apdx. D).

$$\hat{\mathbf{H}}_t^{k+1}(j,\rho):=\arg\min_h \sigma_w^{-2}\|\mathbf{y}_{j,t}^{k+1}-\hat{\mathbf{s}}_{t,j}^{\rho,k}\cdot h)\|_2^2+\lambda_\rho|h|, \qquad (3.28)$$

where $\hat{\mathbf{s}}_{t,j}^{\rho,k} := [\hat{\mathbf{s}}_{t-K+1,j}^{k}(\rho), \hat{\mathbf{s}}_{t-K+2,j}^{k}(\rho), \ldots, \hat{\mathbf{s}}_{t,j}^{k}(\rho)]^T$ and $\mathbf{y}_{j,t}^{k+1}$ is a $K \times 1$ vector with entries

$$\mathbf{y}_{j,t}^{k+1}(m) = x_j(t-K+m) - \sum_{\ell=1,\ell\neq\rho}^{r} \hat{\mathbf{H}}_t^k(j,\ell)\hat{\mathbf{s}}_{t-K+m,j}^{k}(\ell),\ m = 1,\ldots,M, \tag{3.29}$$

where $\hat{\mathbf{s}}_{t-K+m,j}^{k}(\ell)$ correspond to the ℓth entry of the state estimates $\hat{\mathbf{s}}_{t-K+m,j}^{k}$ obtained via ADMM as described next at the end of coordinate cycle k. Using similar arguments as in Apdx. D, it turns out that the optimal solution in (3.28) is:

$$\hat{\mathbf{H}}_t^{k+1}(j,\rho) = \text{sgn}([\hat{\mathbf{s}}_{t,j}^{\rho,k}]^T\mathbf{y}_{j,t}^{k+1}) \times \left(\frac{[\hat{\mathbf{s}}_{t,j}^{\rho,k}]^T\mathbf{y}_{j,t}^{k+1}}{\|\hat{\mathbf{s}}_{t,j}^{\rho,k}\|_2^2} - \frac{\lambda_\rho}{2\sigma_{w,j}^{-2}\|\hat{\mathbf{s}}_{t,j}^{\rho,k}\|_2^2} \right)_+ . \tag{3.30}$$

Sensor j can update $\hat{\mathbf{H}}_t^{k+1}(j,\rho)$, for $\rho = 1,\ldots,r$, locally since both $\mathbf{y}_{j,t}^{k+1}$ and $\hat{\mathbf{s}}_{t,j}^{\rho,k}$ can be formed using its local measurements $\mathbf{x}_j(\tau)$ and the local estimates $\hat{\mathbf{s}}_{\tau,j}^{k}$.

ADMM-based minimization of (3.27) **w.r.t** $\mathbf{s}_{\tau,j}$ **and** $\mathbf{z}_{\tau,j}^{j'}$**'s:**

Here C ADMM iterations are applied per coordinate cycle $k+1$ to obtain across sensors estimates for the states $\mathbf{s}_{\tau,j}$ for the time window $\tau = t-K,\ldots,t$. ADMM involves the following three steps (detailed later):

i) Update the Lagrange multipliers using gradient ascent iterations. Let $\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)$ and $\boldsymbol{\omega}_{\tau,j}^{j',k+1}(\varkappa)$ denote the ADMM updates for the multipliers $\mathbf{v}_{\tau,j}^{j'}$ and $\boldsymbol{\omega}_{\tau,j}^{j'}$, where the positive integer $\varkappa$ denotes the ADMM iteration index performed during coordinate cycle $k+1$. Here C ADMM iterations are nested within coordinate cycle $k+1$.

ii) Minimize the augmented Lagrangian function in (3.27) w.r.t $\mathbf{s}_{\tau,j}$ for $j = 1,\ldots,p$, while fixing $\mathbf{z}_{\tau,j}^{j'}$, $\mathbf{v}_{\tau,j}^{j'}$ and $\boldsymbol{\omega}_{\tau,j}^{j'}$ to their most up-to-date values, namely $\mathbf{z}_{\tau,j}^{j',k+1}(\varkappa)$, $\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)$ and $\boldsymbol{\omega}_{\tau,j}^{j',k+1}(\varkappa)$ obtained at step i) to obtain update $\mathbf{s}_{\tau,j}^{k+1}(\varkappa+1)$.

iii) Minimize the augmented Lagrangian function in (3.27) w.r.t the variables $\mathbf{z}_{\tau,j}^{j'}$ while fixing $\mathbf{s}_{\tau,j}$, $\mathbf{v}_{\tau,j}^{j'}$ and $\boldsymbol{\omega}_{\tau,j}^{j'}$ to their most up-to-date values $\mathbf{s}_{\tau,j}^{k+1}(\varkappa)$ obtained at step ii), and $\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)$, $\boldsymbol{\omega}_{\tau,j}^{j',k+1}(\varkappa)$ obtained at step i).

Specifically, the first step of applying ADMM, during coordinate descent cycle $\varkappa+1$, is to update the multipliers using the gradient ascent iterations:

$$\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)=\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa-1)+c(\mathbf{s}_{\tau,j}^{k+1}(\varkappa)-\mathbf{z}_{\tau,j}^{j',k+1}(\varkappa)) \tag{3.31}$$

$$\boldsymbol{\omega}_{\tau,j}^{j',k+1}(\varkappa)=\boldsymbol{\omega}_{\tau,j}^{j',k+1}(\varkappa-1)+c(\mathbf{s}_{\tau,j}^{k+1}(\varkappa)-\mathbf{z}_{\tau,j'}^{j,k+1}(\varkappa)) \tag{3.32}$$

for $j'\in\mathcal{N}_j$, and $\tau=t-K,...,t$. Warm starts are employed to initialize the Lagrange multipliers before applying C ADMM iterations, namely $\mathbf{v}_{\tau,j}^{j',k+1}(0)=\mathbf{v}_{\tau,j}^{j',k}(C)$. The same procedure is followed to update and initialize the variables $\mathbf{s}_{\tau,j}$, $\mathbf{z}_{\tau,j}^{j'}$ and $\boldsymbol{\omega}_{\tau,j}^{j'}$.

The second step involves minimizing the augmented Lagrangian in (3.27) w.r.t. $\mathbf{s}_{\tau,j}$, while fixing the other variables to their most up-to-date values, namely $[\hat{\mathbf{H}}_t^{k+1}]_{j:}$ (j-th row of $\hat{\mathbf{H}}_t^{k+1}$), $\mathbf{z}_{\tau,j}^{j',k+1}(\varkappa)$, $\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)$ and $\boldsymbol{\omega}_{\tau,j}^{j',k+1}(\varkappa)$. Then, as established in Apdx. F, the update $\mathbf{s}_{\tau,j}^{k+1}(\varkappa+1)$ is formed as:

$$\begin{aligned}\mathbf{s}_{\tau,j}^{k+1}(\varkappa+1)=&(2\mathbf{G}_{\tau,j}^T\mathbf{G}_{\tau,j}+2c|\mathcal{N}_j|\mathbf{I}_{r\times r})^{-1} \\ &\times[2\mathbf{G}_{\tau,j}^T\boldsymbol{\xi}_{\tau,j}-\sum_{j'\in\mathcal{N}_j}(\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)+\boldsymbol{\omega}_{\tau,j}^{j',k+1}(\varkappa))+c\sum_{j'\in\mathcal{N}_j}(\mathbf{z}_{\tau,j}^{j',k+1}(\varkappa)+\mathbf{z}_{\tau,j'}^{j,k+1}(\varkappa))]\end{aligned} \tag{3.33}$$

The third ADMM step entails minimization of (3.27) w.r.t $\mathbf{z}_{\tau,j}^{j'}$, while fixing the remaining variables to their most recent updates, namely $\hat{\mathbf{H}}_t^{k+1}$, $\mathbf{s}_{\tau,j}^{k+1}(\varkappa+1)$ $\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)$ and $\boldsymbol{\omega}_{\tau,j}^{j',k+1}(\varkappa)$. Then, in Apdx. F it is shown that

$$\begin{aligned}\mathbf{z}_{\tau,j}^{j',k+1}(\varkappa+1)=&\frac{1}{2}[\mathbf{s}_{\tau,j}^{k+1}(\varkappa+1)+\mathbf{s}_{\tau,j'}^{k+1}(\varkappa+1)] \\ &+\frac{1}{2c}[\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)+\boldsymbol{\omega}_{\tau,j'}^{j,k+1}(\varkappa)],\ \text{for } j=1,...,p \text{ and } j'\in\mathcal{N}_j.\end{aligned} \tag{3.34}$$

In fact, the Lagrangian multipliers $\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)$ and $\boldsymbol{\omega}_{\tau,j}^{j',k+1}(\varkappa)$ can be initialized randomly without affecting the convergence of the three ADMM steps as $\varkappa\to\infty$, see e.g., [97]. Substituting (3.34) into (3.31), and initializing the multipliers such that $\mathbf{v}_{\tau,j}^{j',k}(0)=$

$-\boldsymbol{\omega}_{\tau,j'}^{j,k}(0)$, then for all τ and $\varkappa = 1, \ldots, C$, it holds $\mathbf{v}_{\tau,j}^{j',k}(\varkappa) = -\boldsymbol{\omega}_{\tau,j'}^{j,k}(\varkappa)$ from which the first updating formulate in (3.31) can rewritten as

$$\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa) = \mathbf{v}_{\tau,j}^{j',k+1}(\varkappa - 1) + 0.5c(\mathbf{s}_{\tau,j}^{k+1}(\varkappa) - \mathbf{s}_{\tau,j'}^{k+1}(\varkappa)) \tag{3.35}$$

Thus, there is no need to update and store $\boldsymbol{\omega}_{\tau,j'}^{j,k}(\varkappa)$, since $\mathbf{v}_{\tau,j}^{j',k}(\varkappa) = -\boldsymbol{\omega}_{\tau,j'}^{j,k}(\varkappa)$. Sensor j updates the Lagrange multipliers $\{\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)\}_{j' \in \mathcal{N}_j}$ for $\tau = t - K, \ldots, t$ and the local state vectors $\mathbf{s}_{\tau,j}$ whose updating formula in (3.33) can be written as

$$\begin{aligned} \mathbf{s}_{\tau,j}^{k+1}(\varkappa + 1) =& (2\mathbf{G}_{\tau,j}^T \mathbf{G}_{\tau,j} + 2c|\mathcal{N}_j|\mathbf{I}_{r\times r})^{-1} \\ & \times [2\mathbf{G}_{j,\tau}^T \boldsymbol{\xi}_{\tau,j} - \sum_{j' \in \mathcal{N}_j} (\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa) - \mathbf{v}_{\tau,j'}^{j,k+1}(\varkappa)) + c \sum_{j' \in \mathcal{N}_j} (\mathbf{s}_{\tau,j}^{j',k+1}(\varkappa) + \mathbf{s}_{\tau,j'}^{j,k+1}(\varkappa))] \end{aligned} \tag{3.36}$$

where it can be clearly seen that the auxiliary variables $\mathbf{z}_{\tau,j}^{j'}$'s have been eliminated. Recursions (3.35) and (3.36) constitute the novel distributed approach where each sensor j just have to keep track of the multipliers $\{\mathbf{v}_{\tau,j}^{j'}\}_{j' \in \mathcal{N}_j}$ and the state updates $\mathbf{s}_{\tau,j}$ for $\tau = 0, ..., t$. The initial values of $\mathbf{v}_{\tau,j}^{j'}$ can be set to 0, while the initial values of the $\mathbf{s}$'s will be obtained from sensor j's local Kalman filter prediction obtained using only its own measurements. In this way, all the sensors exchange information to their single-hop neighboring sensors. The convergence results in [97] ensure that if the number of ADMM iterations $C \to \infty$ during coordinate cycle $k+1$, then the local variables $\mathbf{s}_{\tau,j}$ become equal and $\lim_{\varkappa \to \infty} \mathbf{s}_{\tau,j}^{k+1}(\varkappa)$ converge to the state estimates in (3.9), when $\mathbf{H}$ is set to $\hat{\mathbf{H}}_t^{k+1}$ in the corresponding quantities used to form them in (3.12).

3.4 Computational and Communication Cost

During time instant t, the augmented Lagrangian function in (3.27) is minimized by applying T coordinate descent cycles. Once new measurements have been collected across sensors, namely $\mathbf{x}_j(t)$ for $j = 1, 2, \ldots, p$, the following process takes place. During coordinate cycle $k + 1$, sensor j first updates the jth row of $\mathbf{H}$, and evaluates $\{\hat{\mathbf{H}}_t^{k+1}(j, \rho)\}_{\rho=1}^r$

via (3.30) using its local state updates $\hat{\mathbf{s}}_{t-K+m,j}^{k}$ for $m = 1, \ldots, K$, the measurements $\mathbf{x}_j(t-k+1), \ldots, \mathbf{x}_j(t)$ and latest coordinate updates $\hat{\mathbf{H}}_t^k(j,\ell),\ \ell = 1$ and $\ell \neq \rho$. Then, sensor j runs C ADMM iterations by carrying out (3.35) and (3.36). In detail, during ADMM iteration $\varkappa+1$, inside coordinate cycle $k+1$, sensor j receives the $r \times 1$ state vectors $\mathbf{s}_{\tau,j}^{k+1}(\varkappa)$ from its neighbors $j' \in \mathcal{N}_j$, and updates its Lagrange multipliers $\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)$ via (3.35).

Sensor j receives the multiplier $\mathbf{v}_{\tau,j'}^{j,k+1}(\varkappa)$ from neighbors $j' \in \mathcal{N}_j$ which are used along with $\{\mathbf{s}_{\tau,j'}^{k+1}(\varkappa)\}_{j' \in \mathcal{N}_j}$ to update $\mathbf{s}_{\tau,j}^{k+1}(\varkappa+1)$ via (3.36). The steps are summarized in Alg. 2 Table.

During one ADMM iteration, sensor j receives $\{\mathbf{v}_{\tau,j'}^{j,k+1}(\varkappa), \mathbf{s}_{\tau,j'}^{k+1}(\varkappa)\}_{\tau=t-K}^{t}$ from its neighbors $j' \in \mathcal{N}_j$, which corresponds to $2r(K+1)|\mathcal{N}_j|$ scalars. Since per time instant t, there are T coordinate cycles with C nested ADMM iterations each, the total scalars received during time instant t are $2r(K+1)CT|\mathcal{N}_j|$. Further, sensor j has to transmit its local state vector estimate $\mathbf{s}_{\tau,j}^{k+1}(\varkappa)$ and multipliers $\{\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)\}_{j' \in \mathcal{N}_j}$ which corresponds to $r(|\mathcal{N}_j|+1)(K+1)$ scalars. Thus, during time instant t sensor j transmits $r(K+1)(|\mathcal{N}_j|+1)CT$ scalars. Thus, the communication cost is proportional to the number of sources r, the neighborhood size and the number of coordinate cycles, as well as ADMM iterations applied per time instant t. Although the proposed algorithm is not scalable with respect to the number of sources, it has to be emphasized that the number of sources in practice is smaller than the number of sensors. The cost is not affected by the total size of the whole network, and is easily manageable for small values of the number of sources r, K, $|\mathcal{N}_j|$, C and T.

3.5 Some Remarks

The two tracking algorithms proposed earlier have complimentary nature. Specifically, the one proposed in Sec. (3.2) is developed for a FC-based setting in which a central computing center receives and processes the data acquired at the sensors. A power constraint was introduced to ensure that the power of the signals to be transmitted does not exceed the total power budget imposed across all sensors. A different setting in which no FC is present, while only communication between neighboring sensors is allowed was considered and led to a totally distributed algorithm. The proposed joint source tracking and association scheme allows sensors at a close distance (and thus efficient to communicate with) to collaborate and exchange information to identify and track the sources. The FC-based setting allows faster processing, though it does not scale well with the size of the network and is prone to FC failures. Such issues are resolved by the distributed setting, though at the expense of slower processing due to the nested ADMM iterations during each coordinate cycle.

3.6 Simulations

Next, we test the tracking and association performance of the two proposed schemes and compare it with alternative approaches. In the numerical tests it is assumed that among the summands in (4.3) only one has strong amplitude whereas the rest are negligible. This pertains to a setting where only one source, say the ρth, is close to sensor S_j whereas the rest are sufficiently far, thus their contribution in $x_j(t)$ is very small. Given the exponential attenuation that vibration sources experience in (3.3), it suffices for the sources to be separated by a distance of the order of $20-30$ meters. Consider the 2D field $[0,60]\times[0,60]m^2$ in which $p=35$ sensors are deployed randomly. In the numerical tests, $r=2$ scalar sources are present. The transition matrix in (3.1) is set equal to $\mathbf{F}=\text{diag}(0.95,0.95)$.

The (j, ρ) entry of matrix $\mathbf{H}_t$ is created according to a Gaussian random variable with expectation $e^{-\alpha\|\mathbf{p}_j-\mathbf{l}_\rho(t)\|_2}$ and variance 10^{-4}, i.e., $\mathbf{H}_t(j,\rho) \sim \mathcal{N}(e^{-\alpha\|\mathbf{p}_j-\mathbf{l}_\rho(t)\|_2}, 10^{-4})$, where $\mathbf{p}_j$ corresponds to the location of sensor j, while $\mathbf{l}_\rho(t)$ the position of the ρth source at time instant t and $\alpha = 0.15$. The sources are located such that sensors $16, 27$ observe source $s_1(t)$ and sensors 18 and 23 observe source $s_2(t)$. This test holds for a total of 120 seconds. The sources move roughly for $0.13m$ per time instant t on a straight line which results the attenuation coefficients corresponding to sensors from which the sources move away to decrease to zero at a rate of approximately 0.98, while the coefficients corresponding to sensors to which the sources approach closer, increase at a rate of $0.98^{-1} = 1.02$. For instance, if source ρ is moving away from sensor j then $\mathbf{H}_t(j,\rho)$ decreases by the ratio $\frac{\mathbf{H}_t(j,\rho)}{\mathbf{H}_{t-1}(j,\rho)} = \frac{e^{-\alpha\|\mathbf{p}_j-\mathbf{l}_\rho(t)\|_2}}{e^{-\alpha\|\mathbf{p}_j-\mathbf{l}_\rho(t-1)\|_2}} = 0.98$ since the source moves $\|\mathbf{p}_j-\mathbf{l}_\rho(t)\|_2-\|\mathbf{p}_j-\mathbf{l}_\rho(t-1)\|_2 \approx 0.13m$ away from sensor j per time instant t. Similarly a rate of increase of 1.02 can be obtained when a source is approaching a sensor. The sparsity-controlling coefficients λ_ρ's are set as $\lambda_1 = 0.005$, $\lambda_2 = 0.005$. The state and measurement noise variances are both set to be $\sigma_u^2 = 1.25$ and $\sigma_w^2 = 1.25$.

The FC-based tracking scheme is tested first. The tracking root mean-square error(RMSE) versus time t is depicted in Fig. 3.3. The smallest tracking RMSE is achieved from the K-lag Kalman Smoother (KS) (using all the data until time t to provide an estimate $\check{\mathbf{s}}_{t-K}$ with delay K), followed by the traditional KF filter. In the first two cases, the (true) $\mathbf{H}_t$ is available. These two schemes treating $\mathbf{H}_t$ as known will be used as benchmarks to compare with the novel methods proposed here. In detail, note that the FC-based scheme evaluates K different state estimates for the sources' state vector $\{\mathbf{s}_\tau\}_{\tau=t-K}^{t}$ at time instant t, according to (3.9). Among these K estimates, a filtered estimate (no delay) $\check{\mathbf{s}}_{t,t}$ is also provided (depicted as the green dashed-dot curve in Fig. 3.3). The performance of these estimates is pretty close to the estimates provided from a standard KF using our estimated $\hat{\mathbf{H}}_t$ (red solid curve in Fig. 3.3). Though in the long run, among the three schemes that do

no have available the true $\mathbf{H}_t$, our FC-based scheme using $\check{\mathbf{s}}_{t-K}$, (see blue curve with cross markers) achieves a better performance while the performance gap with the traditional KF (purple curve with 'x' markers) is reduced with time.

Considering the same FC-based setting described earlier, here we consider a limited power budget. The same testing scenario with $p = 35$ sensors monitored field is considered here. The average tracking RMSE for tracking two sources is depicted in Fig. 3.4 with state and observation noise variances both set equal to $\sigma_u^2 = 1.25$ and $\sigma_w^2 = 1.25$. The total power budget is set as $P_{tot} = 40$, resulting on average (50 Monte Carlo runs) of 86% of all the sensors being selected to transmit their data to the fusion center. Our approach is compared with a FC-based tracking scheme where 86% of the sensors is selected randomly (blue solid curve). Moreover, our approach is compared with a KF scheme utilizing the true $\mathbf{H}_t$ and using all sensors' data which, not surprisingly, achieves the lowest tracking RMSE (black crossed curve). This will act as a benchmark. Note that our approach (red dashed curve in Fig. 3.4) is following relatively close the benchmark KF scheme using the true sensing matrix $\mathbf{H}$, while 86% of the sensors is utilized to respect the total power constraint. This implies that the sensor selection process proposed in Sec. 3.2.4 efficiently finds sensors that contain information about the underlying sources, while respecting the available power budget.

Fig. 3.5 shows the average steady-state RMSE versus the percentage of sensors being selected i) via the scheme in Sec. 3.2.4; and ii) randomly. Note that the steady-state RMSE achieved by our proposed sensor selection scheme outperforms the random selecting scheme different values of the portion of selected sensors. In fact the advantage of using the sensor selection scheme in Sec. 3.2.4 increases as fewer sensors are being selected, following closely the steady-state RMSE achieved using all sensors and having available the true $\mathbf{H}$. Thus, the proposed sensor selection scheme judiciously selects sensors to respect the total power budget without significantly compromising the steady-state tracking RMSE.

The tracking performance of the novel distributed tracking and association algorithm in Sec. 3.3 is tested next. Again $p = 35$ sensors are considered, while there are $r = 4$ sources. The state transition matrix is set as $\mathbf{F} = \text{diag}(0.95, 0.95, 0.95, 0.95)$. The observation matrix $\mathbf{H}_t$ is generated as described earlier. The sparsity-controlling coefficients λ_ρ's are set as $\lambda_1 = 0.008$, $\lambda_2 = 0.007$, $\lambda_3 = 0.006$ and $\lambda_4 = 0.005$, while the incremental step parameter c in the ADMM scheme is set equal to 1.0. Fig. 3.6 depicts the average tracking root mean-square error of 150 Monte Carlo trials for different tracking schemes with $K = 80$, $\sigma_u^2 = 3.25$ and $\sigma_w^2 = 4.25$. The following four tracking schemes are tested and compared: i) the distributed tracking scheme in Sec. 3.3 (blue with cross markers curve); ii) the FC-based approach described in Sec. 3.2 (black dashed curve); iii) the distributed tracking scheme in Sec. 3.3 where $\mathbf{H}$ is fixed to the true $\mathbf{H}_0$ and not recursively updated later on (red solid curve); and iv) the FC-based approach described in Sec. 3.2 where the true $\mathbf{H}_t$ is used when estimating the states (purple with 'x' markers curve). The number of ADMM iterations in the i) and iii) schemes is set to $C = 50$ per coordinate descent cycle. Clearly, the centralized FC-based methods (black dashed and purple with 'x' markers curves) achieve better tracking performance than the distributed algorithm. Nevertheless, our distributed algorithm (blue with cross markers curve) will eventually stay fairly close to the FC-based tracking scheme when $\mathbf{H}_t$ is not known. This is to be expected since in the distributed setting it takes more time for the information to propagate through the network. When the distributed scheme in 3.3 loses its capability to estimate $\mathbf{H}_t$ (red solid curve) then performance deteriorates compared to the fully blown distributed version (blue with cross markers curve) which includes estimation of $\mathbf{H}$. The lowest tracking RMSE is achieved by the centralized tracking scheme while using the true sensing matrix $\mathbf{H}_t$ throughout the tracking period.

In the same network setting, it is examined next in Fig. 3.7 how the number of ADMM iterations affects the steady-state RMSE of the distributed tracking scheme in Sec. 3.3.

Two different noise settings are considered: i) $\sigma_u^2 = 1.25, \sigma_w^2 = 1.25$ (low-noise setting); and ii) $\sigma_u^2 = 3.25, \sigma_w^2 = 4.25$ (high-noise setting). The more ADMM iterations are applied, the closer to the centralized performance the distributed scheme is getting at the expense of a higher computational and communication complexity. The steady-state RMSE versus the window length K is depicted in Fig. 3.8 for the two different noise settings described earlier. It can be seen that the tracking error decreases with an increasing window length K. The performance gap between the distributed and centralized schemes is smaller in the low-noise case compared to the high-noise setting.

The probability of successfully recovering the support of $\mathbf{H}$ at $t = 120s$ versus the window size K is plotted in Fig. 3.9. Note that if the window length is $K = 50$ or less, the sensing matrix $\mathbf{H}$ is not recovered well. As the window length becomes larger, the support of the unknown sensing matrix $\mathbf{H}$ becomes more likely to be recovered. The proposed distributed tracking scheme in Sec. 3.3 keeps close to the FC-based centralized scheme especially when a relatively large window length (say, $K = 80$) is selected, for a number of $C = 50$ ADMM iterations. The probability of recovering the support of $\mathbf{H}$ at $t = 120s$ versus the number of ADMM iterations C is also shown in Fig. 3.10. The larger the number of ADMM iterations, the more likely to recover correctly the support of $\mathbf{H}$.

3.7 Conclusions

The paper considers the problem of jointly associating sources with sensors and subsequently tracking them in i) a fusion center based sensor network; and ii) an ad hoc network architecture. The algorithm proposed for the FC-based is exploiting sparsity in the sensing matrix along with a power constraint, to jointly estimate the support of the sensing matrix along with the source states. Coordinate descent techniques along with solving a linear program are the building stones of the proposed algorithm. In the ad hoc setting the

alternating direction method of multipliers is utilized to fully distribute the joint tasks of association and source tracking across the sensors. Numerical tests demonstrate the potential of the proposed schemes.

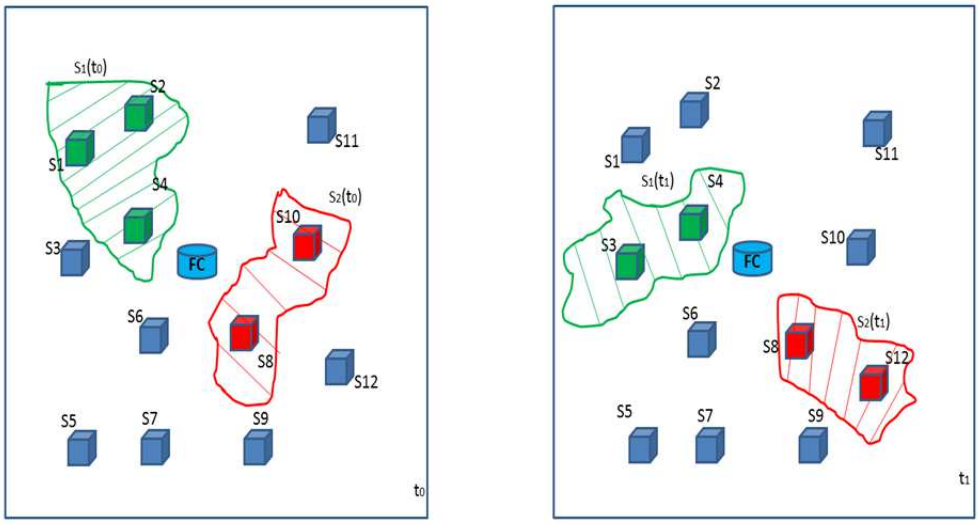

Figure 3.1. Time-varying fusion-center based multi-sensor multi-source configuration. .

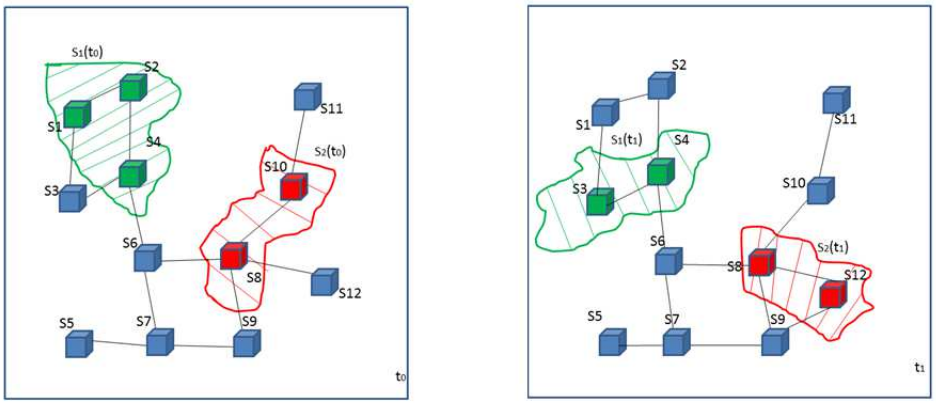

Figure 3.2. Ad hoc distributed time-varying multi-sensor multi-source configuration. .

Algorithm 3 FC-Based Joint Sensors-Sources Association and Tracking

Start-up stage: Acquire (at a sufficiently fast sampling rate) Q data from all sensors and apply the covariance sparse decomposition scheme in [101] to obtain an estimate $\hat{\mathbf{H}}_0$. Initialize the states as $\mathbf{s}(0) = [\mathbf{s}_1(0), \ldots, \mathbf{s}_r(0)]$. Initialize the weights d_j's such that $\hat{d}_{j,0} = 1,\ \forall j$ for which $[\hat{\mathbf{H}}_0]_{j,:} \neq 0$ (otherwise $\hat{d}_{j,0} = 0$), and $\sum_{j=1}^{p} \hat{d}_{j,0}\hat{\sigma}^2_{x,j,0} \leq P_{tot}$.

for $t = 1, 2, 3, \ldots$ **do**

for $\tau = t - K, \ldots, t$ **do**

Receive data from sensor $j \in \{1, \ldots, p\}$ for which $\hat{d}_{j,t-1} = 1$.

Using $\hat{\mathbf{H}}_{t-1}$ in (3.8) and only those summands for which $\hat{d}_{j,t-1} = 1$, update the source state estimates $\check{\mathbf{s}}_{t-K,t}, \ldots, \check{\mathbf{s}}_{t,t}$ via (3.9).

Using $\check{\mathbf{s}}_{t-K,t}, \ldots, \check{\mathbf{s}}_{t,t}$ and only those summands in (3.8) for which $\hat{d}_{j,t-1} = 1$, update the entries of $\hat{\mathbf{H}}_t$ employing the formulas in (3.15) for all j for which $\hat{d}_{j,t-1} = 1$ and $\rho = 1, \ldots, r$.

if $t \bmod \Omega = 0$ **then**

Reapply start-up stage steps to update $\mathbf{H}_t$, $\{\mathbf{s}_\tau\}_{\tau=t-K}^{t}$ and $\{d_j\}_{j=1}^{p}$.

end if

end for

end for

Algorithm 4 Distributed Joint Sensors-Sources Association and Tracking

1: **Start-up stage:** Acquire (at a sufficiently fast sampling rate) Q data and apply the covariance sparse decomposition scheme in [101] to obtain an estimate $\hat{\mathbf{H}}_0$. Initialize the multipliers $\mathbf{v}$ and states $\mathbf{s}$ as described in Sec 3.6.

2: **for** $t = 1, 2, 3, \ldots$ **do**

3: **for** $\tau = t - K, \ldots, t$ **do**

4: **for** $k = 1, \ldots, T$ **do**

5: Sensor $j = 1, \ldots, p$ updates $\{\hat{\mathbf{H}}_t^{k+1}(j, \rho)\}_{\rho=1}^{r}$ via updating recursion (3.30).

6: Sensor j applies C ADMM iterations to update locally $\mathbf{s}_{\tau,j}^{k+1}(\varkappa + 1)$ and $\mathbf{v}_{\tau,j}^{j',k+1}(\varkappa)$ for $j' \in \mathcal{N}_j$ and obtain updates $\hat{\mathbf{s}}_{\tau,j}^{k+1} := \mathbf{s}_{\tau,j}^{k+1}(C)$.

7: **end for**

8: **end for**

9: **end for**

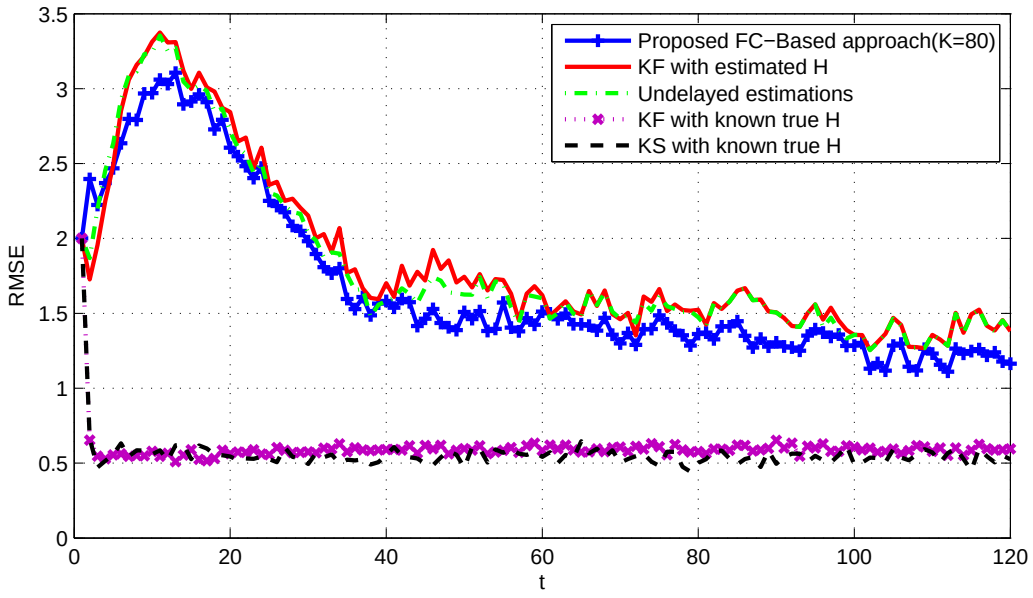

Figure 3.3. FC-based tracking root mean-square error (RMSE) vs. time t.

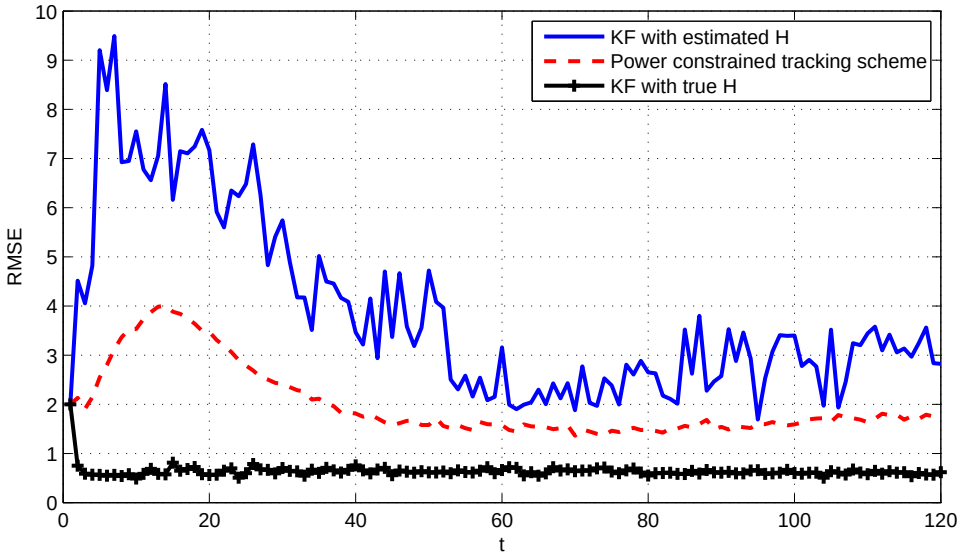

Figure 3.4. Tracking error vs. time t under power constraints..

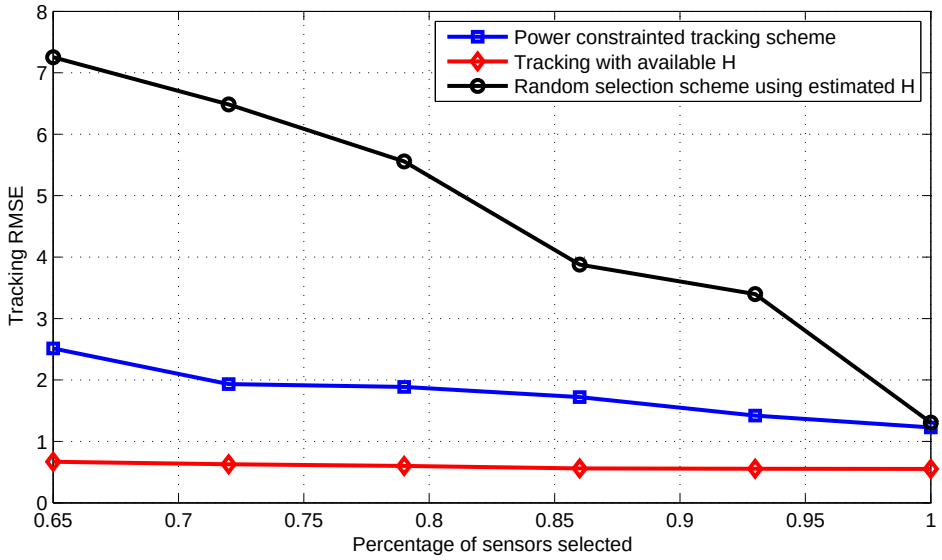

Figure 3.5. Steady-state root mean-square error (RMSE) vs. percentage of sensors selected.

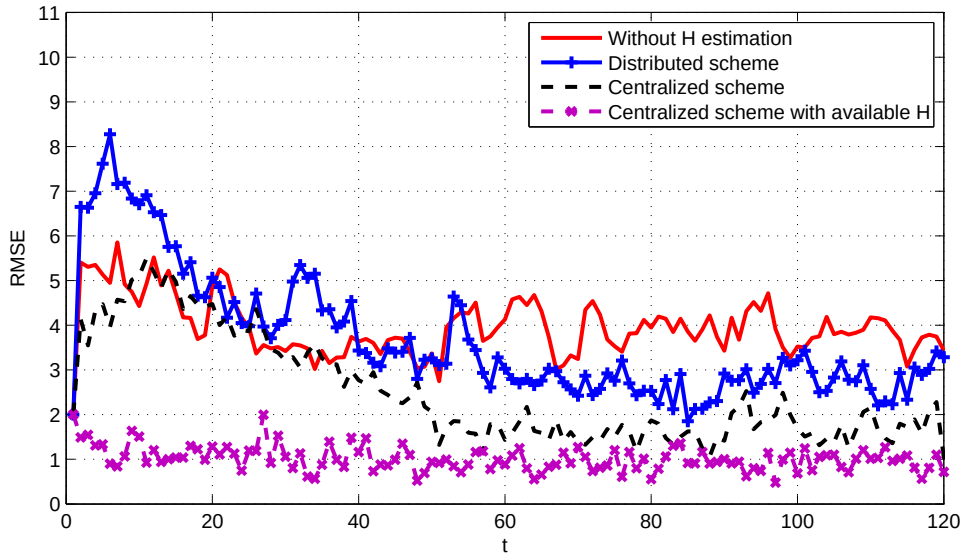

Figure 3.6. Tracking root mean-square error vs time t..

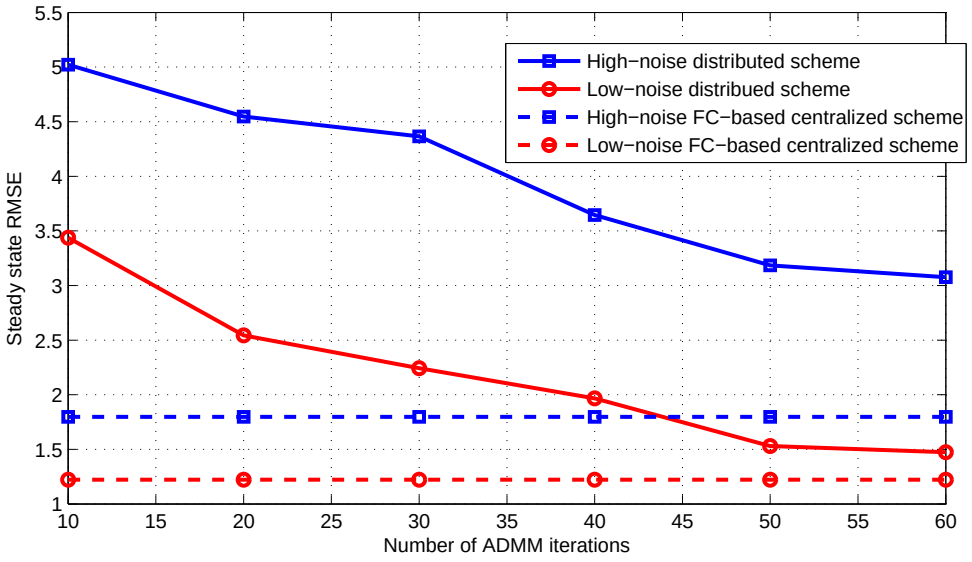

Figure 3.7. Steady-state root mean-square error (RMSE) vs. number of ADMM iterations.
.

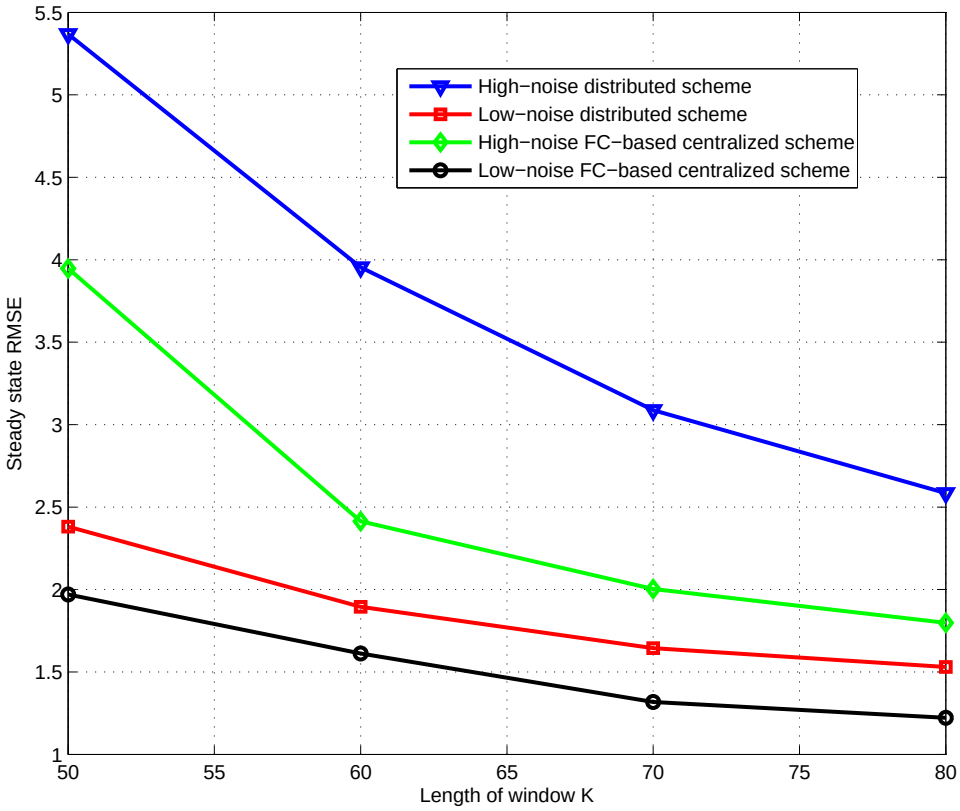

Figure 3.8. Steady-state root mean-square error (RMSE) vs. length of time window K. .

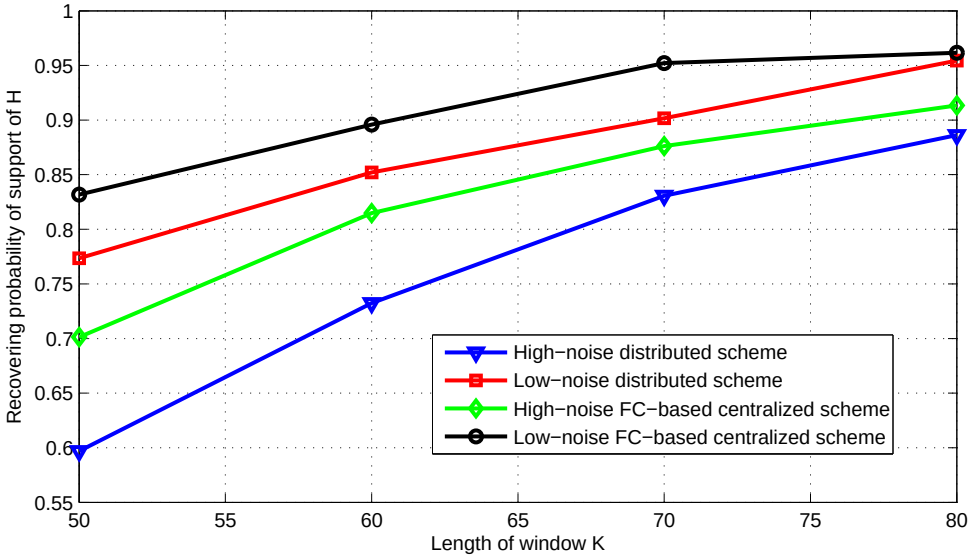

Figure 3.9. Probability of correctly recovering support of $\mathbf{H}$ vs. length of time window K.
.

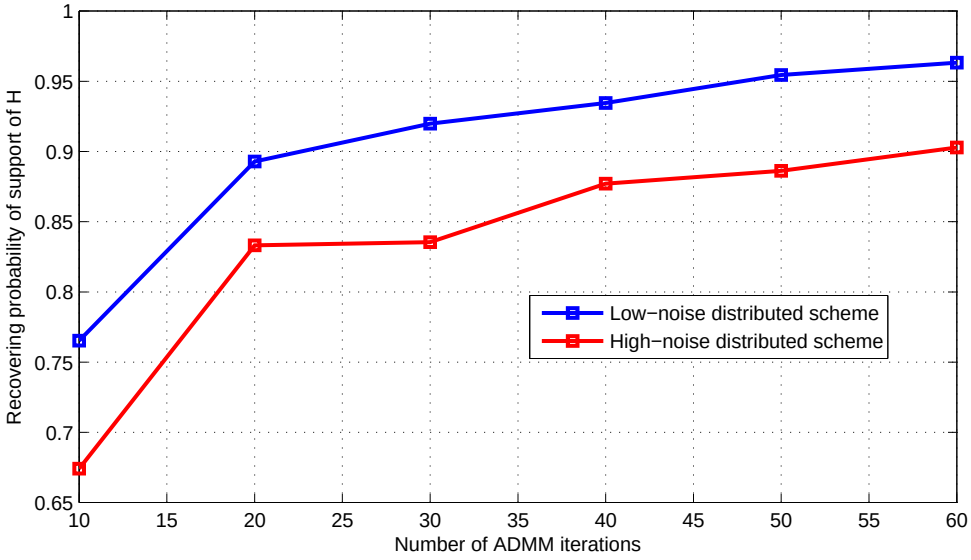

Figure 3.10. Probability of correctly recovering support of $\mathbf{H}$ vs. number of ADMM iterations..

CHAPTER 4

Exploiting Sensor Mobility and Covariance Sparsity For Distributed Tracking of Multiple Sparse Targets

4.1 Problem Setting

An ad hoc sensor network conformed by p mobile sensors is considered here. The sensors monitor a field where an unknown and possibly time-varying number of moving targets is present. Each sensor communicates only with its neighboring sensors which are within its communication range and are able to exchange information with a single-hop of communication. The single-hop neighborhood of sensor j is denoted as $\mathcal{N}_j(t)$, where t denotes the time index.

In general, all targets are assumed to be moving in a K-dimensional space. Then every target, say the ρth is characterized by a $2K \times 1$ state vector which contains both its position coordinates and velocity information for each coordinate. The position coordinates for the ρth target at time t are stacked in vector $\mathbf{p}_\rho(t) = [p_{\rho,x_1}(t), \ldots, p_{\rho,x_K}(t)]^T$, while the velocity per coordinate is in vector $\mathbf{v}_\rho(t) = [v_{\rho,x_1}(t), \ldots, v_{\rho,x_K(t)}]^T$. So at time t, the state vector can be written as $\mathbf{s}_\rho(t) = [\mathbf{p}_\rho^T(t), \mathbf{v}_\rho^T(t)]^T$, while it evolves according to a near constant velocity model [7]. Specifically, the ρth target's state vector evolves according to the following constant velocity model, see e.g., [7]:

$$\mathbf{s}_\rho(t+1) = \mathbf{F}\mathbf{s}_\rho(t) + \mathbf{u}_\rho(t), \tag{4.1}$$

where $\mathbf{F}$ is the $2K \times 2K$ state transition matrix, and $\mathbf{u}_\rho(t)$ the zero-mean Gaussian state noise with variance $\mathbf{\Sigma}_u$. Matrices $\mathbf{F}$ and $\mathbf{\Sigma}_u$ are given as follows:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \dots & \Delta T & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 & \dots & \Delta T \\ 0 & 0 & 1 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}, \quad \mathbf{\Sigma}_u = \sigma_u^2 \begin{bmatrix} (\Delta T)^3/3 \cdot \mathbf{I}_K & (\Delta T)^2/2 \cdot \mathbf{I}_K \\ (\Delta T)^2/2 \cdot \mathbf{I}_K & \Delta T \cdot \mathbf{I}_K \end{bmatrix}, \quad (4.2)$$

where σ_u^2 is the noise variance and $\mathbf{I}_K$ denotes the identity matrix of size $K \times K$, while ΔT denotes the sampling period.

Sensor j, senses the moving targets, by acquiring at time t a scalar measurement depending on the target location according to the following nonlinear model:

$$x_j(t) = \sum\nolimits_{\rho=1}^{R} a_\rho(t) d_{j,\rho}^{-2}(t) + w_j(t), \; j = 1, \dots, p, \quad (4.3)$$

where $a_\rho(t)$ denotes the intensity of a signal emitted from the ρth target and $d_{j,\rho}(t) = \|\mathbf{p}_j(t) - \mathbf{p}_\rho(t)\|$ is the distance between sensor j and the ρth target at time t. The total number of targets which move in the field through the lifespan of the SN is indicated as R, while $w_j(t)$ denotes the white sensing noise with variance σ_w^2 and zero-mean. The following assumptions are introduced in the considered setting:

- **A**1: In the measurement model in Eq:(4.3), it is assumed that the targets act as transmitters and each sensor will receive one reflection of the signal emitted from the targets. Signals emitted from the targets propagate via free space, explaining the $d_{j,\rho}^{-2}(t)$ attenuation coefficients, and are superimposed as shown in Eq:(4.3), see e.g., [40].
- **A**2: The signal amplitudes $a_\rho(t)$ are considered to be uncorrelated across the different targets.

- $\mathbf{A}3$: Among the summands $a_\rho(t)d_{j,\rho}^{-2}(t)$ in Eq:(4.3), only one has a large amplitude when sensor j is close to the ρth target, whereas others are negligible due to the square-law attenuation $d_{j,\rho}^{-2}(t)$ caused by the free space propagation.

Note that assumption A3 corresponds to a setting where at most one target is present within the sensing range of a sensor. Note that this is a more relaxed version of the common assumption that one sensor just contains the measurement of a specific target, [52, 85, 118]. The signal amplitudes $a_\rho(t)$ will be nonzero for the interval in which the corresponding target is active and moving while is kept at zero when the target is inactive and disappears.

The emitted, from the targets, signals $a_\rho(t)$ could correspond to communication radio signals that possibly the targets are transmitting, e.g., targets could correspond to cell phone users moving in an area, or military vehicles or moving radio emitters that move within the monitored area and need to be tracked, see e.g., [106]. The deployed sensors are listening for these signals to track the moving entities. The targets could correspond to independent entities, thus it is expected that the information bits they transmit, are uncorrelated, giving rise to uncorrelated transmission signals [87]. Thus, the communication radio signals that the targets may be emitting are utilized to perform tracking and move the sensors appropriately. Applications include localization and tracking of mobile users in wireless networks, as well as tracking of radio emitters in tactical environments [106].

Stacking all the sensor measurements in Eq:(4.3) on an $p \times 1$ vector it follows:

$$\mathbf{x}_t = \mathbf{D}_t\mathbf{a}_t + \mathbf{w}_t, \text{ where } \mathbf{a}_t := [a_1(t)\ a_2(t) \dots a_R(t)]^T, \tag{4.4}$$

where $\mathbf{D}_t$ is a $p \times R$ matrix with entries $\mathbf{D}_t(j,\rho) = d_{j,\rho}^{-2}(t)$ with $j = 1, \dots, p$ and $\rho = 1, \dots, R$. The noise $\mathbf{w}_t$ has covariance $\mathbf{\Sigma}_w = \sigma_w^2\mathbf{I}_p$. Given that the entries of $\mathbf{a}_t$ are uncorrelated, it follows readily that the data covariance matrix is

$$\mathbf{\Sigma}_{x,t} = \mathbf{D}_t\mathbf{\Sigma}_a\mathbf{D}_t^T + \sigma_w^2\mathbf{I}_p = \bar{\mathbf{D}}_t\bar{\mathbf{D}}_t^T + \sigma_w^2\mathbf{I}_p, \tag{4.5}$$

where $\mathbf{\Sigma}_a$ is the diagonal covariance matrix of $\mathbf{a}_t$, while $\bar{\mathbf{D}}_t := \mathbf{D}_t\mathbf{\Sigma}_a^{1/2}$. Among the R entries in $\mathbf{a}_t$, there will be $r(t)$ nonzero entries corresponding to the active targets moving at the sensed field at t. In the setting here, once a target becomes inactive (i.e. $a_\rho(t) = 0$) it remains inactive for the rest of time.

The goal is to enable the mobile sensors to track an unknown number of targets present in the monitored field. Novel target association and sensor mobility strategies will be combined with tracking techniques to enable sensors to accurately track the different target trajectories. Proper kinematic strategies will be developed to allow only a small percentage of target-informative sensors to move, different from existing approaches [67, 136] where *all* sensors are moving at every time instant that may be more resource demanding. Judiciously selecting and moving sensors will enable target tracking even when the targets move outside the area originally monitored by the sensors.

4.2 Distributed Association, Tracking, and Sensor Kinematic Strategies

4.2.1 Target-Informative Sensor Selection

Due to the presence of multiple target in the monitored field, the first goal is to determine sets of sensors, namely $\mathcal{S}_{\rho,t}$, that acquire information bearing measurements about the ρth target. From the observation model in Eq:(4.4), note that the strong-amplitude entries of the ρ column in $\mathbf{D}_t$, namely $\{\mathbf{D}_{t,:\rho}\}_{\rho=1}^R$, can reveal the sensors within subset $\mathcal{S}_{\rho,t}$. Specifically, recall that $\mathbf{D}_t(j,\rho) = d_{j,\rho}^{-2}(t)$, thus when sensor j and target ρ are close in distance then the corresponding entry is expected to have large amplitude, while the further away they get from each other the closer to zero the corresponding entry gets. The matrix $\mathbf{D}_t$ can be assumed approximately sparse. Thus, the strong-amplitude entries (away from zero) in $\mathbf{D}_{t,:\rho}$ can be used to determine the informative sensor members of $\mathcal{S}_{\rho,t}$ at time

instant t. Thus, determining $\mathcal{S}_{\rho,t}$ boils down to the problem of recovering the support of the columns of $\mathbf{D}_t$.

To recover the sparse matrix $\bar{\mathbf{D}}_t$ in Eq:(4.4) the data covariance matrix will be decomposed into sparse factors. Due to the fact that the targets and sensors may be moving while the number of targets is changing, the sparse sensor data covariance $\mathbf{\Sigma}_{x,t}$ is also time-varying. In practice, the ensemble covariance is not available and needs to be estimated. To this end, exponential weighing is employed to estimate the time-varying covariance entries. The notion of exponential weighing in recursive least-squares used in processing non-stationary signals, see e.g., [100, Ch. 9], is estimating the time-varying covariance matrix here as follows:

$$\hat{\mathbf{\Sigma}}_{x,t} = \frac{1-\omega}{1-\omega^{t+1}} \sum_{\tau=0}^{t} \omega^{t-\tau} (\mathbf{x}_\tau - \bar{\mathbf{x}}_t)(\mathbf{x}_\tau - \bar{\mathbf{x}}_t)^T, \tag{4.6}$$

where $\omega \in (0,1)$ denotes a forgetting factor and

$$\bar{\mathbf{x}}_t = \frac{1-\omega}{1-\omega^{t+1}} \sum_{\tau=0}^{t} \omega^{t-\tau} \mathbf{x}_\tau, \tag{4.7}$$

corresponds to a real-time estimate for the ensemble mean at time instant t. Note that ω in Eqs. (4.6) and Eq:(4.7) is used in a way that puts more emphasis to the recent data while it gradually forget the past data, which is exactly what an up-to-date estimator needs to do for the time-varying setting considered here. The scaling $(1-\omega)(1-\omega^{t+1})^{-1}$ in (4.6) and (4.7) is to ensure that the two estimates $\hat{\mathbf{\Sigma}}_{x,t}$ and $\bar{\mathbf{x}}_t$ for the ensemble quantities $\mathbf{\Sigma}_{x,t}$ and $\mathbb{E}[\mathbf{x}_t]$ will be unbiased.

To account for the nearly sparse structure of $\bar{\mathbf{D}}_t$, the unknown number of targets and single-hop connectivity of the sensor nodes the following formulation relying on norm-one/norm-two regularization is utilized:

$$\begin{aligned}\left(\hat{\mathbf{M}}_t, \{\hat{\sigma}_j\}_{j=1}^m\right) := \arg \min_{\mathbf{M}_t, \{\sigma_j\}_{j=1}^m} \|\mathbf{E} \odot \left(\hat{\mathbf{\Sigma}}_{x,t} - \mathbf{M}_t\mathbf{M}_t^T - \text{diag}(\sigma_{1,t}^2, \ldots, \sigma_{p,t}^2)\right)\|_F^2 \\ + \sum_{\ell=1}^{L} \left(\lambda_\ell \|\mathbf{M}_{t,:\ell}\|_1 + \phi \|\mathbf{M}_{t,:\ell}\|_2\right),\end{aligned} \tag{4.8}$$

where $\odot$ denotes the Hadamard operator (entry-wise matrix product), while σ_j^2 is the noise variance estimate at sensor j, and L is an upper bound for the number of active sensed targets $r(t)$ ($L \geq r(t)$) and $\mathbf{M}_t \in \mathbb{R}^{p\times L}$ contains L columns that estimate the sparse columns of $\bar{\mathbf{D}}_t$. $\mathbf{M}_{t,:\ell}$ denotes the ℓth column of $\mathbf{M}_t$. The formulation was first proposed in [93, 101] to perform target-sensor association in a network of stationary sensors that do not have moving capabilities. Here this formulation will be utilized to determine the different sets of informative sensors observing different targets before being integrated with kinematic control rules.

The Hadamand operator $\odot$ along with the adjacency matrix $\mathbf{E}$ in Eq:(4.8) allow only the single-hop covariance entries to be used since they can be calculated by direct communication of the corresponding neighboring sensors. The first term in Eq:(4.8) accounts for the structure in Eq:(4.5). Sparsity is induced in the columns of $\mathbf{M}_t$ using the norm-one term in Eq:(4.8), (see e.g., [137]), while λ_ℓ denotes the nonnegative sparsity-controlling coefficient used to adjust the number of zeros in $\hat{\mathbf{M}}_{t,:\ell}$. The coefficient $\phi \geq 0$ in the last term of Eq:(4.8) promotes group sparsity among rows, [129], thus is introduced to adjust the number of nonzero columns of $\hat{\mathbf{M}}_t$ needed to approximate $\hat{\mathbf{\Sigma}}_{x,t}$. This is done to zero-out unnecessary columns in $\hat{\mathbf{M}}$ when the number of active targets in the field is smaller than R. The number of nonzero columns in $\mathbf{M}_t$ indirectly estimates the number of targets at time instant t, namely $\hat{r}(t)$.

The cost in Eq:(4.8) is minimized by an iterative minimization scheme based on coordinate descent [10, 112], where sensor j is responsible for updating the jth row of $\mathbf{M}_t$, namely $\mathbf{M}_{t,j:}$. Specifically, the cost is minimized wrt one entry of $\mathbf{M}_t$ or $\text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$, while keeping the rest fixed to their most up-to-date values. Sensor j updates the entries $\{\mathbf{M}_t(j,\ell)\}_{\ell=1}^L$ and variance $\sigma_{j,t}^2$. During one coordinate cycle all the entries of $\mathbf{M}_t$ and $\text{diag}(\sigma_{1,t}^2, \ldots, \sigma_{p,t}^2)$ will be updated.

The updates for entries $\hat{\mathbf{M}}_t^k(j,\ell)$ will be formed by differentiating Eq:(4.8) wrt $\mathbf{M}_t(j,\ell)$ and setting the derivative equal to zero, while fixing the rest of the entries of $\mathbf{M}_t$, and $\sigma_{j,t}$ to their most recent updates in $\hat{\mathbf{M}}_t^{k-1}$ and $\{\sigma^2_{j,t,k-1}\}$ evaluated at cycle $k-1$. It turns out (details in [93, 101]) that during coordinate cycle k, the update $\hat{\mathbf{M}}_t^k(j,\ell)$ can be obtained as the value that gets the minimum possible cost in Eq:(4.8) (while fixing the rest of the variables) among the candidate values: i) $z=0$; ii) the real positive roots of the third-degree polynomial

$$z^3+\left[\sum_{i\in\mathcal{N}_j}[\hat{\mathbf{M}}_t^{k-1}(i,\ell)]^2-\psi^k_{t,\Sigma}(j,j,\ell)+\frac{\phi}{2}\right]z-\left[\sum_{i\in\mathcal{N}_j}\psi^k_{t,\Sigma}(j,\mu,\ell)\hat{\mathbf{M}}_t^{k-1}(i,\ell)\right]+\frac{\lambda_\ell}{4}=0 \tag{4.9}$$

and iii) the real negative roots of the third-degree polynomial

$$z^3+\left[\sum_{i\in\mathcal{N}_j}[\hat{\mathbf{M}}_t^{k-1}(\mu,\ell)]^2-\psi^k_{t,\Sigma}(j,j,\ell)+\frac{\phi}{2}\right]z-\left[\sum_{i\in\mathcal{N}_j}\psi^k_{t,\Sigma}(j,i,\ell)\hat{\mathbf{M}}_t^{k-1}(i,\ell)\right]-\frac{\lambda_\ell}{4}=0 \tag{4.10}$$

where

$$\psi^k_{t,\Sigma}(j,i,\ell):=\hat{\mathbf{\Sigma}}_{x,t}(j,i)-\delta_{j,i}\hat{\sigma}^2_{j,t,k-1}-\sum_{\ell'=1,\ell'\neq\ell}^{L}\hat{\mathbf{M}}_t^{k-1}(j,\ell')\hat{\mathbf{M}}_t^{k-1}(i,\ell'), \tag{4.11}$$

while $\delta_{j,i}$ denotes the Kronecker delta function, i.e., $\delta_{j,i}=1$ if $j=i$, and $\delta_{j,i}=0$ if $j\neq i$. The roots of the two aforementioned polynomials can be calculated using companion matrices, see e.g., [50].

Furthermore, during cycle k at time instant t, the noise variance estimates across sensors can be updated as

$$\hat{\sigma}^2_{j,t,k}=\hat{\mathbf{\Sigma}}_{x,t}(j,j)-\hat{\mathbf{M}}^k_{t,j:}(\hat{\mathbf{M}}^k_{t,j:})^T,\quad j=1,\ldots,p. \tag{4.12}$$

Sensor j needs to communicate only with its single-hop neighbors in $\mathcal{N}_j$, in order to evaluate the coefficients of the polynomials in Eq:(4.9) and Eq:(4.10), and to update the noise

variance estimates in Eq:(4.12). It can be shown that as $k \to \infty$, the updates $\hat{\mathbf{M}}_t^{k-1}$ converge at least to a stationary point of Eq:(4.8). Further, the sparsity controlling coefficients $\{\lambda_\ell\}_{\ell=1}^L$ can be set using the strategy proposed in [101, Sec.]. Once the sparse columns $\{\hat{\mathbf{M}}_{t,:\ell}\}$ are estimated, their support (the indices of relatively strong-amplitude entries) is used to determine which sensors sense a specific target at time t.

4.2.2 Tracking via Extended Kalman Filtering

The target informative sensor subsets $\mathcal{S}_{\rho_\ell,t}$ for $\ell = 1, \ldots, \hat{r}(t)$, where $\hat{r}(t)$ corresponds to an estimate of the number of targets at time instant t obtained from the number of nonzero columns of $\hat{\mathbf{M}}_t := \hat{\mathbf{M}}_t^{\bar{K}}$, after applying $\bar{K}$ coordinate cycles. Extended Kalman filtering is employed to process the nonlinear observations and track each target's location using the observations of the corresponding set $\mathcal{S}_{\rho_\ell,t}$. For simplicity in exposition, the specifics of EKF will be delineated here for $K = 2$ dimensions, but it can be readily generalized to more dimensions. The target state estimator and corresponding error covariance matrix, obtained by the extended Kalman filter using the observations in $\mathcal{S}_{\rho_\ell,t}$ for target ρ_ℓ are denoted as $\hat{\mathbf{s}}_{\rho_\ell}(t|t)$ and $\mathbf{M}_{\rho_\ell}(t|t)$, respectively. The prediction step, see e.g. [63], involves the following updating recursions for the state estimator and covariance at time instant t

$$\hat{\mathbf{s}}_{\rho_\ell}(t+1|t) = \mathbf{F}\hat{\mathbf{s}}_{\rho_\ell}(t|t), \quad \hat{\mathbf{M}}_{\rho_\ell}(t+1|t) = \mathbf{F}\hat{\mathbf{M}}_{\rho_\ell}(t|t)\mathbf{F}^T + \mathbf{\Sigma}_u. \tag{4.13}$$

The measurements of the sensors within set $\mathcal{S}_{\rho_\ell,t}$ will then be used to carry out the correction step of the extended Kalman filter which involves the following update recursions:

$$\hat{\mathbf{s}}_{\rho_\ell}(t+1|t+1) = \hat{\mathbf{s}}_{\rho_\ell}(t+1|t) + \mathbf{K}(t+1) \cdot [\mathbf{x}_{t+1} - a_\rho(t)\hat{\mathbf{D}}_{\mathcal{S}_{\rho_\ell,t}}] \tag{4.14}$$

$$\mathbf{M}_{\rho_\ell}(t+1|t+1) = \mathbf{M}_{\rho_\ell}(t+1|t) + \mathbf{D}_{\nabla,\rho_\ell}^T(t+1|t) \cdot \sigma_w^2 \mathbf{I}_{|\mathcal{S}_{\rho_\ell,t}|} \cdot \mathbf{D}_{\nabla,\rho_\ell}(t+1|t), \tag{4.15}$$

for $\ell = 1, \ldots, \hat{r}(t)$, while the matrix $\mathbf{K}_{\rho_\ell}(t+1)$ corresponds to the Kalman gain given as

$$\mathbf{K}_{\rho_\ell}(t+1) = \mathbf{M}_{\rho_\ell}(t+1|t+1) \cdot \mathbf{D}_{\nabla,\rho_\ell}^T(t+1|t) \cdot \sigma_w^2 \mathbf{I}_{|\mathcal{S}_{\rho_\ell,t}|}, \tag{4.16}$$

where $\hat{\mathbf{D}}_{\mathcal{S}_{\rho_\ell,t}}$ is a $|\mathcal{S}_{\rho_\ell,t}|\times 1$ vector whose entries are given by $\{||\mathbf{p}_j(t)-\hat{\mathbf{p}}_{\rho_\ell}(t+1|t)||^{-2}\}_{j\in\mathcal{S}_{\rho_\ell,t}}$, in which $\hat{\mathbf{p}}_{\rho_\ell}(t+1|t)$ is the ρ_ℓ-th target position extracted from the state prediction $\hat{\mathbf{s}}_{\rho_\ell}(t+1|t)$. Further, $\mathbf{D}_{\nabla,\rho_\ell}(t+1|t)$ is the $|\mathcal{S}_{\rho_\ell,t}| \times 4$ matrix whose rows constitute of gradient $\nabla\mathbf{D}_t(j,\rho_\ell)$ with respect to the state vector $\mathbf{s}_{\rho_\ell}$ and evaluated at $\hat{\mathbf{s}}_{\rho_\ell}(t+1|t)$ for $j \in \mathcal{S}_{\rho_\ell,t}$, i.e.,

$$\nabla_{\mathbf{s}_\rho}\mathbf{D}_t(j,\rho_\ell)\,|_{\mathbf{s}_{\rho_\ell}=\hat{\mathbf{s}}_{\rho_\ell}(t+1|t)} = \frac{2\cdot[\mathbf{p}_{j,x}(t)-\hat{\mathbf{p}}_{\rho_\ell,x}(t+1|t),\mathbf{p}_{j,y}(t)-\hat{\mathbf{p}}_{\rho_\ell,y}(t+1|t),0,0]^T}{[(\mathbf{p}_{j,x}(t)-\hat{\mathbf{p}}_{\rho_\ell,x}(t+1|t))^2+(\mathbf{p}_{j,y}(t)-\hat{\mathbf{p}}_{\rho_\ell,y}(t+1|t))^2]^2}. \tag{4.17}$$

Within each informative subset of sensors $\mathcal{S}_{\rho_\ell,t}$, the sensor closest in distance to the predicted position of the ρ_ℓ-th target, namely $\hat{\mathbf{s}}_{\rho_\ell}(t+1|t)$, is set as a the subset head sensor that will gather the measurements of all other sensors in $\mathcal{S}_{\rho_\ell,t}$ and perform the EKF tracking recursions.

4.2.3 Sensor Kinematics

The focus in this section is to derive kinematic rules for the target-informative sensors, which are selected according to the scheme in Sec. 4.2.1, such that they follow closely the moving targets and give accurate position estimates. The benefit from having a few sensors moving is that targets can be tracked even when they move away from the original field monitored by the sensors. Having sensors following closely the moving targets can provide more reliable measurements about the targets than just using static sensors. Note that only informative sensors close to the targets will be responsible for carrying out the tracking procedure leading to resource savings. Toward this end, the informative sensors in each subset $\mathcal{S}_{\rho_\ell}$ will be placed/move in locations that minimize the trace of the error covariance associated with the estimator $\hat{\mathbf{s}}_{\rho_\ell}(t|t)$. This will ensure that the informative sensors associated with each target move to a location that will provide measurements that result good tracking accuracy. The idea of minimizing a scalar function of the predicted error covariance was

also applied in moving all sensors in a network for tracking *a single* target, [80, 86, 136]. Here kinematic strategies are derived in the presence of multiple targets, while a judiciously selected small portion of target-informative sensors will be moving instead of all sensors moving.

Among the two terms in the covariance matrix in equation (15), only the second term is affected by the sensors' location. The latter term, after using Eq:(4.17), can be written as:

$$\sum_{j\in\mathcal{S}_{\rho_\ell,t}} \frac{4}{[(\mathbf{p}_{j,x}(t+1)-\hat{\mathbf{p}}_{\rho_\ell,x}(t+1|t))^2+(\mathbf{p}_{j,y}(t+1)-\hat{\mathbf{p}}_{\rho_\ell,y}(t+1|t))^2]^3}. \tag{4.18}$$

Clearly Eq:(4.18) depends on the position of the sensors associated with target ρ_ℓ, namely the sensors in subset $\mathcal{S}_{\rho_\ell,t}$, at time instant t. Letting $\mathbf{p}_j(t+1) := [\mathbf{p}_{j,x}(t+1) \;\; \mathbf{p}_{j,y}(t+1)]^T$ notice that the trace cost in Eq:(4.18) is separable with respect to the position of each sensor j within subset $\mathcal{S}_{\rho_\ell,t}$. Thus, the position of sensor $j \in \mathcal{S}_{\rho_\ell,t}$ at time instant $t+1$ is determined by minimizing the corresponding summand in Eq:(4.18), i.e., the updated location for sensors $j \in \mathcal{S}_{\rho_\ell,t}$ can be found as

$$\begin{aligned}\mathbf{p}_j(t+1) = \arg\min_{\mathbf{p}_{j,x},\mathbf{p}_{j,y}} & \frac{4}{([\mathbf{p}_{j,x}-\hat{\mathbf{p}}_{\rho_\ell,x}(t+1|t)]^2+[\mathbf{p}_{j,y}-\hat{\mathbf{p}}_{\rho_\ell,y}(t+1|t)]^2)^3} \\ \text{s. to } & [\mathbf{p}_{j,x}-\hat{\mathbf{p}}_{\rho_\ell,x}(t+1|t)]^2+[\mathbf{p}_{j,y}-\hat{\mathbf{p}}_{\rho_\ell,y}(t+1|t)]^2 < R^2 \end{aligned} \tag{4.19}$$

Note that the inequality constraint in Eq:(4.19) ensures that the new location of the moving sensors $j \in \mathcal{S}_{\rho_\ell}$ will be within distance R_j from the latest target location estimate $\hat{\mathbf{p}}_{\rho_\ell}(t+1|t)$. This inequality further ensures that all sensors in $\mathcal{S}_{\rho_\ell}$ will move to new locations which are 'close' to the target. After applying the triangle inequality for the new locations of two sensors j and j' within $\mathcal{S}_{\rho_\ell}$ and using the constraint in Eq:(4.19) it turns out that the new location should satisfy

$$\|\mathbf{p}_j(t+1)-\mathbf{p}_{j'}(t+1)\|_2 \leq \sqrt{2}R, \tag{4.20}$$

which ensures that each subset $\mathcal{S}_{\rho_\ell}$ of moving sensors will stay connected, as long as the communication range of the sensing units is at least $\sqrt{2}R$. Thus, R can be set such that the moving sensors stay connected. Connectivity is necessary to elect a head sensor for each moving subset of sensors that will acquire the measurements of all other sensors and perform clustering. Details of the algorithm are given in Section 4.3.1. Note that existing approaches do not entail mechanisms as the one introduced here to ensure that sensors will be connected.

Next, the modified barrier method (MBM) is utilized [10, pg. 423] to allow every sensor $j \in \mathcal{S}_{\rho_\ell}$ to solve Eq:(4.19) and determine its next location. To this end, let $f(\mathbf{p}_j)$ denote the cost in Eq:(4.19) and $g(\mathbf{p}_j)$ denote the left hand side function of the inequality constraint in Eq:(4.19). MBM involves an iterative application of the following unconstrained minimization problem (where κ denotes the iteration index within time instant $t+1$):

$$\mathbf{p}_j^\kappa(t+1) \in \arg \min_{\mathbf{p}_{j,x},\mathbf{p}_{j,y}} \{f(\mathbf{p}_j) + \frac{\mu^\kappa}{c^\kappa}\phi[c^\kappa \cdot g(\mathbf{p}_j)]\}, \tag{4.21}$$

where the Lagrange multiplier-like scalar μ^κ is updated as

$$\mu^{\kappa+1} = \mu^\kappa \cdot \nabla\phi[c^\kappa \cdot g(\mathbf{p}_j^\kappa(t+1))], \tag{4.22}$$

while the barrier function $\phi[\tau]$ is chosen as a logarithmic function having the form

$$\phi(\tau) = -ln(1-\tau) \tag{4.23}$$

and c^κ is a penalty parameter associated with the inequality constraint in Eq:(4.19) that is updated according to the recursion

$$c^\kappa = \frac{\gamma^\kappa}{\mu^\kappa} \tag{4.24}$$

where $\{\gamma^k\}$ is a positive monotonically increasing scalar sequence [10].

For the logarithmic barrier function in Eq:(4.23) the updating recursion of the multipliers in Eq:(4.22) takes the following form

$$\mu^{k+1} = \frac{\mu^{\kappa}}{1 - c^{\kappa} g(\mathbf{p}_j^{\kappa}(t+1))}. \tag{4.25}$$

Further, letting $F(\mathbf{p}_j) := f(\mathbf{p}_j) + \frac{\mu^{\kappa}}{c^{\kappa}} \phi[c^{\kappa} \cdot g(\mathbf{p}_j)]$ the coordinates of the new sensor location $\mathbf{p}_j(t+1)$ are updated during iteration κ according to the following gradient descent recursions

$$\mathbf{p}_{j,x}^{\kappa+1}(t+1) = \mathbf{p}_{j,x}^{\kappa}(t+1) - \Gamma \cdot \frac{dF(\mathbf{p}_j)}{d\mathbf{p}_{j,x}} \Big|_{\mathbf{p}_{j,x}=\mathbf{p}_{j,x}^{\kappa}(t+1)} \tag{4.26}$$

$$\mathbf{p}_{j,y}^{\kappa+1}(t+1) = \mathbf{p}_{j,y}^{\kappa}(t+1) - \Gamma \cdot \frac{dF(\mathbf{p}_j)}{d\mathbf{p}_{j,y}} \Big|_{\mathbf{p}_{j,y}=\mathbf{p}_{j,y}^{\kappa}(t+1)} \tag{4.27}$$

where Γ is the step size for the gradient descent method, while the derivatives in Eq:(4.26) are given as

$$\begin{aligned}
\frac{dF(\mathbf{p}_j)}{d\mathbf{p}_{j,x}} \Big|_{\mathbf{p}_{j,x}=\mathbf{p}_{j,x}^{\kappa}(t+1)} =& \frac{24 \cdot [-\mathbf{p}_{j,x}^{\kappa}(t+1) + \hat{\mathbf{p}}_{\rho,x}(t+1|t)]}{([\mathbf{p}_{j,x}^{\kappa}(t+1) - \hat{\mathbf{p}}_{\rho,x}(t+1|t)]^2 + [\mathbf{p}_{j,y}^{\kappa} - \hat{\mathbf{p}}_{\rho,y}(t+1|t)]^2)^4} \\
&+ \frac{\mu^{\kappa}}{1 - c^{\kappa} g(\mathbf{p}_j^{\kappa}(t+1))} \cdot 2 \cdot (\mathbf{p}_{j,x}^{\kappa}(t+1) - \hat{\mathbf{p}}_{\rho,x}(t+1|t)), \\
\frac{dF(\mathbf{p}_j)}{d\mathbf{p}_{j,y}} \Big|_{\mathbf{p}_{j,y}=\mathbf{p}_{j,y}^{\kappa}(t+1)} =& \frac{24 \cdot [-\mathbf{p}_{j,y}^{\kappa}(t+1) + \hat{\mathbf{p}}_{\rho,y}(t+1|t)]}{([\mathbf{p}_{j,x}^{\kappa}(t+1) - \hat{\mathbf{p}}_{\rho,x}(t+1|t)]^2 + [\mathbf{p}_{j,y}^{\kappa}(t+1) - \hat{\mathbf{p}}_{\rho,y}(t+1|t)]^2)^4} \\
&+ \frac{\mu^{\kappa}}{1 - c^{\kappa} g(\mathbf{p}_j^{\kappa}(t+1))} \cdot 2 \cdot (\mathbf{p}_{j,y}^{\kappa}(t+1) - \hat{\mathbf{p}}_{\rho,y}(t+1|t)).
\end{aligned} \tag{4.28}$$

During time instant $t+1$ each sensor j within the subset $\mathcal{S}_{\rho_\ell,t}$ will keep updating their location until the cost function in Eq:(4.19) is not reduced more than a predefined threshold ϵ within two consecutive updating steps $\kappa, \kappa+1$. The location $\mathbf{p}_j(t+1)$ will be set to the last update $\mathbf{p}_j^{K'}(t+1)$ obtained after K' MBM iterations during time instant $t+1$. The following steps are carried out during the determination of the sensor's new location:

S1) The head sensor in each subset $\mathcal{S}_{\rho_\ell,t}$ sends the predicted position estimate of target ρ_ℓ, namely $\hat{\mathbf{p}}_{\rho_\ell}(t+1|t)$, to all sensors in $\mathcal{S}_{\rho_\ell,t}$.

S2) Each sensor in $j \in \mathcal{S}_{\rho_\ell,t}$, determines its new location using the MBM scheme. Sensors

in $\mathcal{S}_{\rho_\ell,t}$ check their distances to other neighboring sensors and if their future location is too close they adjust their coordinates to avoid collision when moving. Similarly, each moving sensor checks the distance between its updated location and the position estimate of target ρ_ℓ, and if too close, adjustments will be made to the sensor's location such that a minimum distance will be kept from the target. Specifically, if sensor j has an updated location $\mathbf{p}_j^{K'}(t+1)$ which is too close to the already updated location of sensor j', namely $\mathbf{p}_{j'}(t+1)$, i.e., $\|\mathbf{p}_j^{K'}(t+1) - \mathbf{p}_{j'}(t+1)\|_2 \leq R_{\text{min},a}$, where $R_{\text{min},a}$ is the smallest distance allowed that two sensors can be separated from each other, then $\mathbf{p}_j^{K'}(t+1)$ is updated as follows:

$$\mathbf{p}_j(t+1) = \mathbf{p}_{j'}(t+1) + R_{\text{min},a}\frac{\mathbf{p}_j^{K'}(t+1) - \mathbf{p}_{j'}(t+1)}{\|\mathbf{p}_j^{K'}(t+1) - \mathbf{p}_{j'}(t+1)\|_2}. \tag{4.29}$$

Similarly, if sensor j has an updated location $\mathbf{p}_j^{K'}(t+1)$ which is too close to the ρ_ℓ target location estimate, i.e., $\|\mathbf{p}_j^{K'}(t+1) - \hat{\mathbf{p}}_{\rho_\ell}(t+1|t)|_2 \leq R_{\text{min},b}$ where $R_{\text{min},b}$ is the smallest distance that a sensor can be placed from a target, then location $\mathbf{p}_j^{K'}(t+1)$ is updated as follows:

$$\mathbf{p}_j(t+1) = \hat{\mathbf{p}}_{\rho_\ell}(t+1|t) + R_{\text{min},b}\frac{\mathbf{p}_j^{K'}(t+1) - \hat{\mathbf{p}}_{\rho_\ell}(t+1|t)}{\|\mathbf{p}_j^{K'}(t+1) - \hat{\mathbf{p}}_{\rho_\ell}(t+1|t)\|_2}. \tag{4.30}$$

The collision-avoidance position modifications in Eq:(4.30) was proposed in [33] to prevent collision of unmanned aerial vehicles with a stationary target. The position updates in Eq:(4.29) and Eq:(4.30) ensure that the updated locations are at distance $R_{\text{min},a}$ and $R_{\text{min},b}$ from another moving sensor, or moving target respectively, satisfying the minimum distance required to prevent collision.

The actual movement can be achieved using for example robotic sensors, see e.g., [18, 19]. Each sensor $j \in \mathcal{S}_{\rho_\ell,t}$ updates its location $\mathbf{p}_j(t+1)$, in a coordinate fashion while the remaining sensors in $\mathcal{S}_{\rho_\ell,t}$ are kept stationary waiting for their turn to update their location.

4.3 Algorithmic Summary

4.3.1 Implementation

At the start-up stage, fast sampling is used to acquire Q measurements fast enough that the initial number of targets $r(0)$ can be assumed stationary. By utilizing the Q acquired data, the subsets of target-informative sensors $\{\mathcal{S}_{\rho_\ell,0}\}$ are initialized, where $\ell = 1, \ldots, \hat{r}(0)$ and $\hat{r}(0)$ is the estimated number of $r(0)$ sensed targets at time $t = 0$ (number of nonzero columns in the sparse matrix $\hat{\mathbf{M}}_0$). One sensor within each $\mathcal{S}_{\rho_\ell,0}$ will be randomly selected as the head sensor, which will collect the measurements $x_j(0)$ and their positions $\mathbf{p}_j(0)$ from all the other sensors $j \in \mathcal{S}_{\rho_\ell,0}$. Each head sensor $C_{\rho_\ell,0}$ averages the positions of the informative sensors in subset $\mathcal{S}_{\rho_\ell,0}$ to be the initial estimate of the corresponding target ρ_ℓ. The latter target location estimate along with the informative measurements $x_j(0)$, for $j \in \mathcal{S}_{\rho_\ell,0}$ are utilized to initialize the recursions of the extended Kalman filtering carrying out the target tracking in Sec. 4.2.2.

At time instant t, every head sensor $C_{\rho_\ell,t}$ has available the state estimates for active target ρ_ℓ, namely $\hat{\mathbf{s}}_{\rho_\ell}(t|t)$, obtained from EKF in Sec. 4.2.2. The target's estimated position $\hat{\mathbf{p}}_{\rho_\ell}(t|t)$ is then used to select a group of 'candidate informative' sensors, which are denoted as $\mathcal{J}_{\rho_\ell,t}$ for target ρ_ℓ at time instant t. This set is formed by having the head sensor transmit the estimated state $\hat{\mathbf{s}}_{\rho_\ell}(t|t)$ to its single-hop neighboring sensors which then transmit the same information to their own neighbors. Every sensor j who receives $\hat{\mathbf{s}}_{\rho_\ell}(t|t)$, from a neighboring sensor, subsequently forwards this estimate only to those sensors $j' \in \mathcal{N}_j$ whose present position is within radius R_s from the estimated target location, i.e., $\|\mathbf{p}_{j'}(t) - \hat{\mathbf{p}}_{\rho_\ell}(t|t)\|_2 \leq R_s$. The parameter R_s can be set to be sufficiently large in order for all ρ_ℓ-target informative sensors to be incorporated in subset $\mathcal{J}_{\rho_\ell,t}$. The sensor subset $\mathcal{J}_{\rho_\ell,t}$ by construction is connected.

Since not all sensors within the candidate subsets $\mathcal{J}_{\rho_\ell,t}$ maybe informative, the scheme in Sec. 4.2.1 is employed among the sensors in $\mathcal{J}_{\rho_\ell,t}$ to find out the target-informative sensor subset $\mathcal{S}_{\rho_\ell,t+1} \subseteq \mathcal{J}_{\rho_\ell,t}$ for all the active targets. Rather than running the target-sensor association scheme in Sec. 4.2.1 in the whole sensor network, it is performed independently in the different sensor subsets $\mathcal{J}_{\rho_\ell,t}$ associated with each target.

Once subsets $\mathcal{S}_{\rho_\ell,t+1}$ are found, the head sensor in each of these subsets is chosen to be the sensor whose distance is the closest to the estimated position of the corresponding target ρ_ℓ, i.e.,

$$C_{\rho_\ell,t+1} = \arg \min_{j \in \mathcal{S}_{\rho_\ell,t+1}} \|\mathbf{p}_j(t) - \hat{\mathbf{p}}_{\rho_\ell}(t|t)\|_2.$$

The head sensor $C_{\rho_\ell,t+1}$ gathers the sensor measurements $x_j(t+1)$ from the informative sensors $j \in \mathcal{S}_{\rho_\ell,t+1}$ to carry out the extended Kalman filtering recursions at time instant $t+1$ as outlined in Sec. 4.2.2. Then, steps S1 and S2 in Sec. 4.2.3 are employed to allow all sensors in $\mathcal{S}_{\rho_\ell,t+1}$ to determine and move to their new positions $\mathbf{p}_j(t+1)$. Note that connectivity of the sensors in $\mathcal{S}_{\rho_\ell,t+1}$ is preserved as explained in Sec. 4.2.3. The head sensor $C_{\rho_\ell,t+1}$ broadcasts the latest state estimate $\hat{\mathbf{s}}_{\rho_\ell}(t+1|t+1)$ to its single-hop neighbors and repeats the process described earlier to update the candidate informative sensor subsets $\mathcal{J}_{\rho_\ell,t+1}$.

It is worth mentioning that the kinematic rules implemented in Sec. 3.3 at each sensor are fully distributed since each sensor requires knowledge only of its location and the estimated target position obtained from the head sensor in $\mathcal{S}_{\rho_\ell,t+1}$. Connectivity of the candidate informative subsets $\mathcal{J}_{\rho_\ell,t+1}$ is ensured by construction irrespective of the sensor movement. This way the sensor-to-target association scheme in Sec. 4.2.1 can still be applied in $\mathcal{J}_{\rho_\ell,t+1}$ and determine the informative sensors.

The target-informative sensor selection scheme in Sec. 4.2.1 may also need to be reapplied across the whole sensor network since moving targets may disappear and not

being sensed anymore, while new targets may appear at different regions of the sensor network. The following conditions are checked to determine such events: (i) If any of the sensor subsets $\mathcal{S}_{\rho_\ell,t+1}$ becomes empty, this implies that some of the targets previously sensed are not present anymore; (ii) If at $t+1$, the energy of a sensor, not previously selected exceeds a certain threshold this indicates that most likely a new target enters the sensed field. The two aforementioned conditions signify that the target configuration has changed and the sensor selection scheme in Sec. 3.1 needs to be reapplied in the sensor network to update the sensor-informative subsets. The novel tracking scheme is summarized as Algorithm 1.

4.3.2 Communication and Computational Expenses

The communication cost of the proposed algorithm is studied next. Note that inter-sensor communication takes place during i) the sensor-to-target association scheme in Sec. 4.2.1; ii) carrying out the EKF tracking steps in Sec. 4.2.2; and iii) when applying the kinematic strategy in Sec. 4.2.3 to move the informative sensors. In detail, at time instant t sensor j has to receive $|\mathcal{N}_j|$ scalar measurements from its neighbors, namely $\{x_{j'}(t+1)\}_{j'\in\mathcal{N}_j}$, to update $\hat{\mathbf{\Sigma}}_{x,t+1}(j,j')$. Furthermore, to implement the association scheme in Sec. 4.2.1 each sensor j receives the updates $\{\hat{\mathbf{M}}_{t+1}^{k-1}(j',\ell)\}_{\ell=1}^{L}$ from neighborhood $\mathcal{N}_j$, corresponding to $L|\mathcal{N}_j|$ scalars in total, to form its local updates $\{\hat{\mathbf{M}}_{t+1}^{k}(j,\ell)\}_{\ell=1}^{L}$. Thus, sensor j receives $(L+1)|\mathcal{N}_j|$ scalars in total. Similarly sensor j has to transmit $x_j(t+1)$ and $\{\hat{\mathbf{M}}_{t+1}^{k-1}(j,\ell)\}_{\ell=1}^{L}$, a total of $L+1$ scalars to its neighbors, per iteration k.

After the target-informative sensors are determined, each head sensor has to carry out the estimation process about the corresponding target's states. Thus, the head sensor $\{C_{\rho_\ell,t}\}$ will collect the measurements $x_j(t)$ from the sensors within $\mathcal{S}_{\rho_\ell,t+1}$. This involves $|\mathcal{S}_{\rho_\ell,t+1}|$ scalar exchanges. Further, all sensors in the subset $\mathcal{S}_{\rho_\ell,t+1}$ will receive four scalars corresponding to the current state estimate. Once the state estimation process (Sec. 4.2.2)

Algorithm 5 Multi-Target Tracking Using Sensor Mobility and Informative Sensor Selection

1: **Start-up stage ($t = 0$)/Reconfiguration ($t \neq 0$):** Every sensor j collects Q measurements $x_j(t)$ and the sensors-targets association scheme in Sec. 4.2.1 is applied in the network to determine the subsets $\mathcal{S}_{\rho_\ell,t}$ with $\ell = 1, \ldots, \hat{r}(t)$, and $\hat{r}(t)$ is the estimated number of targets.

2: **for** $\tau = t, \ldots,$ **do**

3: Determine the head sensor $C_{\rho_\ell,\tau}$ in each $\mathcal{S}_{\rho_\ell,\tau}$ for $\ell = 1, \ldots, \hat{r}(t)$.

4: Each head sensor $C_{\rho_\ell,\tau}$ receives measurements $x_j(\tau)$ from $j \in \mathcal{S}_{\rho_\ell,\tau}$ to perform tracking for targets $\rho_\ell = 1, \ldots, \hat{r}(t)$ via the EKF recursions in Sec. 4.2.2.

5: Informative sensors $j \in \mathcal{S}_{\rho_\ell,\tau}$ relocate themselves according to the sensor kinematics introduced in Sec. 4.2.3.

6: Each head sensor $C_{\rho_\ell,\tau}$ propagates the state estimates $\hat{\mathbf{s}}_{\rho_\ell}(\tau)$ to every sensor j that can be reached from $C_{\rho_\ell,\tau}$ by a multi-hop path and satisfies $\|\mathbf{p}_j(\tau) - \hat{\mathbf{p}}_{\rho_\ell}(\tau|\tau)\|_2 < R_s$. Then, the candidate informative sets $\{\mathcal{J}_{\rho_\ell,\tau+1}\}_{\ell=1}^{\hat{r}(t)}$ are formed.

7: The sensor selection scheme in Sec. 4.2.1 is carried out in each subset $\mathcal{J}_{\rho_\ell,\tau+1}$ to identify the target-informative sets $\mathcal{S}_{\rho_\ell,\tau+1}$.

8: If target configuration has changed then go to step 1, otherwise go to step 2.

9: **end for**

is carried out by the head sensors, sensor communication also occurs among the informative sensors when adjusting their new location to avoid collision with closely located sensors (Sec. 4.2.3). Specifically, sensor j receives $2|\mathcal{N}_j|$ scalars from its neighbors, corresponding to their two location coordinates, while it sends out its own location. It is worth mentioning that the communication complexity for each sensor is linear with respect to its neighborhood size $|\mathcal{N}_j|$, and the upper bound number of present targets L. The latter linear cost advocates that the proposed framework is a communication-affordable distributed approach.

When applying the scheme in Sec. 4.2.1 during each coordinate cycle k and time instant t each sensor j has to form the coefficients in Eq:(4.10), Eq:(4.9) with a computational complexity of the order $\mathcal{O}(|\mathcal{N}_j|)$, while determining the roots of the two third-order polynomial in Eq:(4.10), Eq:(4.9) involves determining the eigenvalues of two 3×3 companion matrices whose complexity is fixed and non-dependent on any algorithmic parameters. The EKF in Sec. 4.2.2 can be carried out at a complexity of the order $\mathcal{O}(K^2 + |\mathcal{S}_\rho|)$, where $K = 4$ here and $|\mathcal{S}_\rho|$ corresponds to the size of the target-informative subsets. The kinematic rules implemented at the target-informative sensors in Sec. 4.2.3 have a computational complexity $\mathcal{O}(K)$.

4.4 Simulations

The tracking performance of the novel scheme is tested in a network with $p = 80$ sensors, which are deployed randomly in the region of $[0, 15] \times [0, 15]m^2$. The tracking root mean-square error (RMSE) is studied and compared with the RMSE attained by the tracking schemes in [86, 136]. The comparison is done using one target since the aforementioned existing approaches can handle one target. Target $\rho = 1$ is initialized at location $[1.5, 11.5]$ and moves with velocities of $[0.15, 0.1]m/s$ respectively along the x-axis and y-axis. The tracking process is carried out for a total of $30s$, with the state noise and observation noise variances set to be $\sigma_u^2 = 0.08$ and $\sigma_w^2 = 0.08$ (corresponding to a sensing SNR of 11 dB). Fig.4.1 depicts in logarithmic scale the tracking RMSE (for better display) of i) the novel approach proposed here; ii) the tracking scheme in [136]; and iii) the tracking approach in [86]. Note that for all the three tracking schemes, the initial position of the target is found by applying the sparsity matrix decomposition scheme in Sec. 4.2.1, ensuring the same initial error for all the three different tracking approaches. As corroborated by Fig. 4.1, our tracking scheme exhibits the lowest tracking RMSE. The approach in [136] attains

the worst performance since the sensors can only move on a grid which reduces accuracy. The scheme in [86] performs worse than our approach since it does not have an informative sensor selection scheme, which results all sensors to move and participate in the tracking process which may reduce accuracy when noisy sensors are utilized.

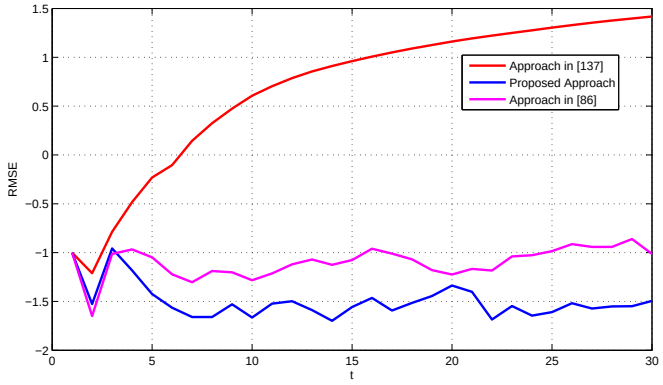

Figure 4.1. Tracking RMSE versus time for a single-target setting..

Fig. 4.2 depicts the distance (in meters) between two moving sensors and the corresponding moving target during the 30s tracking period. It can be seen that the distance is decreasing with time which further implies that the proposed approach allows the informative sensors to closely follow the target.

Next, the performance of our novel tracking scheme is tested in a setting where the number of targets is changing. Specifically, targets $\rho = 1, 2$ start moving at positions $[1.5, 11.5]$, $[5, 7]$ and follow the dynamics in Eq:(4.1), with velocities of $[0.15, 0.1]m/s$ and $[0.4, 0.13]m/s$ along the x-axis and y-axis respectively. Targets $\rho = 1, 2$ move in the field for the time interval $[1, 30]$s and then are not sensed anymore. In the interval $[15, 17]$s, no targets are present in the field. Then, targets $\rho = 3, 4$ start at positions $[6.1, 4.8]$, $[9.0, 4.0]$ and move according to same state model followed by the first two for the time interval

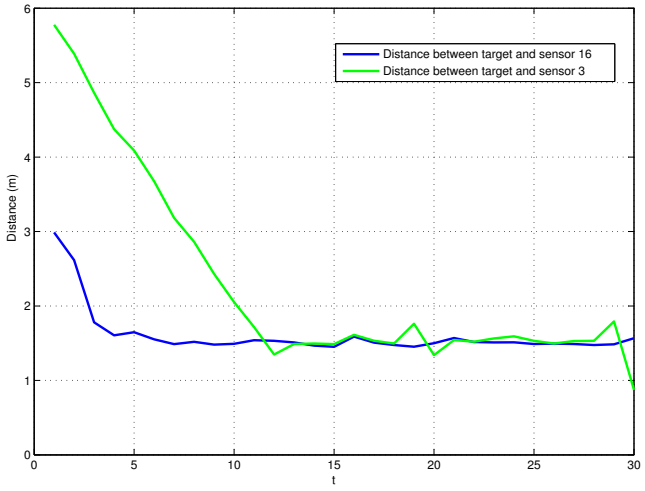

Figure 4.2. Distance between target and moving sensors versus time..

$[32, 50]$s but with velocities $[0.03, 0.35]m/s$ and $[0.25, -0.25]m/s$. Again no targets are present during the interval $[50, 52]$s. Then, three targets $\rho = 5, 6, 7$ start moving at positions $[15, 1.1]$, $[13, 13.5]$ and $[12, 6]$, according to Eq:(4.1), for the time interval $[52, 70]$s and velocities $[-0.1, 0]m/s$ for target $\rho = 5$, and $[0.12, -0.03]$ for both $\rho = 6, 7$ along the x-axis and y-axis respectively. Figure .4.3 depicts the original positions of the sensors represented by blue circles.

Figs. 4.4-4.6 show snapshots of the configuration of the targets and the moving sensors at different time instances. Details for the different curves and coloring on those figures is given in the caption below the figures. From Fig. 4.4 it is clear that for the first two targets, even though the targets move out of the original $[0, 15] \times [0, 15]$ region, both targets are still tracked well as some of the sensors follow them closely. Note that only informative sensors, on average around 10% of the total number of sensors, move according to the proposed kinematic rules in Sec. 4.2.3, while the majority of other sensors which are not close to the moving targets are not moving and maintain their original positions. Similar

conclusions can be extracted from figs. 4.5-4.6, where different sets of targets are moving on the field. It should be emphasized that it is not known that at $t = 30$ and $t = 50$ the target configuration is changing. As discussed in Sec. 4.3, such changes can be determined by having all sensors self-checking the energy level of their measurements, while the head sensors monitor the informative sensor subsets whether they are empty or not.

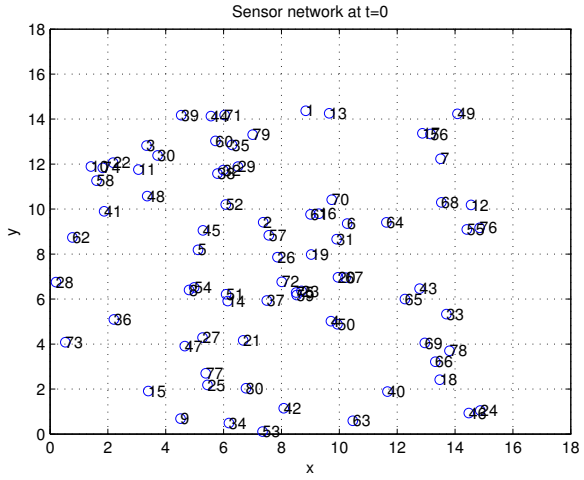

Figure 4.3. Original sensor network topology (before applying kinematic rules to the sensors)..

The tracking RMSE for the above tracking setting is plotted in Fig. 4.7 in logarithmic scale. The proposed tracking framework exploiting sensor mobility is compared with a tracking scheme where sensors are stationary and not moving. In the immobile sensor network, when the targets are moving away from the sensed field sensors willl acquire less and less reliable measurements leading to the dramatic increase of the tracking RMSE (blue dashed curves). In contrast, the proposed framework here enables sensors to follows closely the targets and achieve a much lower tracking RMSE. So even though the targets move out of the original sensed field, there is always a group of sensors keeping adjacent

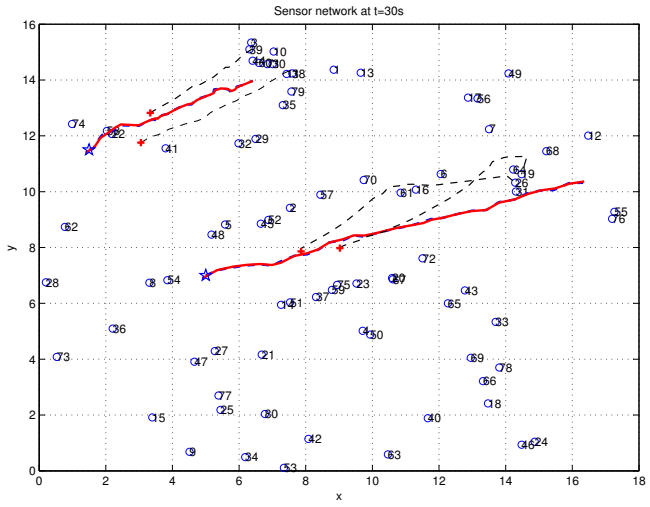

Figure 4.4. Snapshot of the trajectories of targets and moving sensors at time instant $t = 30$s..

to it, and can be selected again to provide accurate measurements. Note that in Fig. 4.7, there are three discontinued curves which corresponds to the error of tracking the three different groups of targets that appear and cease to exist in the monitored field at different time periods. This leads to the RMSE discontinuity at time $t = 31s$ and $t = 51s$ since there are no targets moving during that time interval and no need for tracking.

Next, a tracking setting is considered with two targets where one of targets splits into two targets at a certain time. Similarly to the previous tracking scenario, two targets $\rho = 1, 2$ initialized at positions $[1.6, 11.5]$, $[5.4, 7]$ (indicated by the blue stars) start moving according to the dynamics in (4.1), with velocities of $\mathbf{v}_1 = [v_{1,x}, v_{1,y}] = [0.15, 0.1]m/s$ and $\mathbf{v}_2 = [v_{2,x}, v_{2,y}] = [0.4, 0.13]m/s$, respectively. As $t = 30$, the second target stops moving while the first one splits into two targets. Target $\rho = 3$ continues to move according to the dynamics of target $\rho = 1$, while target $\rho = 4$ moves with velocities $v_x = 0.4m/s$ and $v_y = -0.5m/s$ along the x-axis and y-axis. The two new targets move for the time interval $[31, 42]s$. The splitting point is indicated by the green star in Fig. 4.8. Fig. 4.8

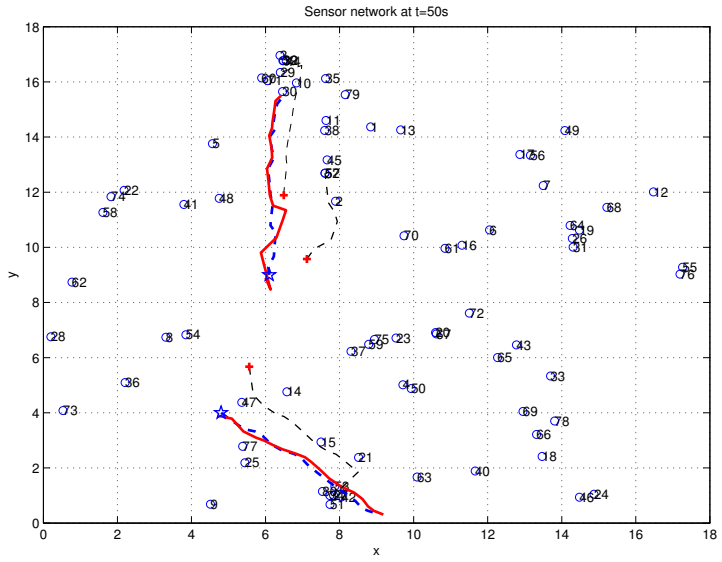

Figure 4.5. Snapshot of the trajectories of targets and moving sensors at time instant $t = 50$s..

shows the trajectories of the targets and some moving sensors, details on the coloring and curve types used can be found in the caption of Fig. 4.8 . The target trajectories in Fig. 4.8 are depicted by blue dashed lines for the time interval $[1, 30]s$ and by blue crossed lines after $t = 30s$. When sensors do not move, the violet estimated trajectories indicate that the split of targets cannot be followed, while target $\rho = 1$ cannot be tracked after a while since is moving away from the immobile sensors. When the kinematic rules in Sec. 4.2.3 are employed, informative sensors follow closely the targets as depicted by the black dashed sensor trajectories. Note that the corresponding estimated red trajectories accurately follow the multiple targets present in the field. As before, the tracking RMSE (logarithm) is compared for the cases where sensors cannot move with the case where the approach in Sec. 4.2.3 is applied. As Fig. 4.9 shows, our active tracking scheme outperforms in terms of tracking accuracy the utilization of stationary sensors. Notice that in Fig. 4.9, after the target splits, tracking using stationary sensors performs much better than before

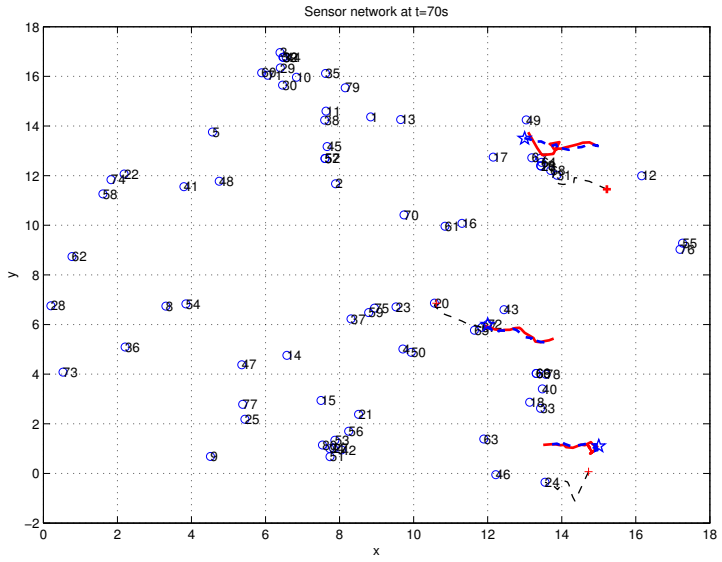

Figure 4.6. Snapshot of the trajectories of targets and moving sensors at time instant $t = 70$s.

splitting. The reason is that when using stationary sensors, for the second tracking phase ($t = [31, 42]s$), there are more sensors originally located close to the trajectories of the targets, compared to the first tracking phase ($t = [1, 30]s$) which makes the tracking error much smaller than the first $30s$, though the performance when using stationary sensors is still worse than tracking using our sensor mobility-based tracking scheme.

4.5 Concluding Remarks

A novel framework combining sparse matrix factorization with proper kinematic rules enable multiple mobile sensors to track multiple targets. An norm-one/norm-two regularized matrix decomposition formulation is utilized to perform sensor-to-target association and select the target informative sensors. Optimal kinematic rules are obtained by minimizing the covariances of parallel extended Kalman filters that track multiple targets using only target-informative sensors. The modified barrier method is utilized to obtain

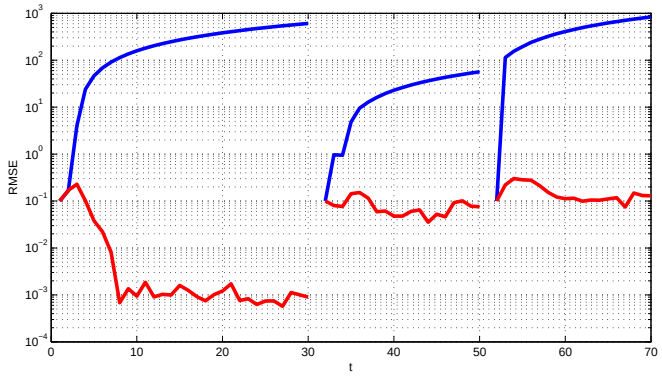

Figure 4.7. Tracking RMSE versus time for tracking the multiple objects in the setting depicted in Figs. 4.4, 4.5 and 4.6..

the sensors' location updates while ensuring that the moving sensors remain connected. Numerical tests in multi-sensor networks, corroborate that our novel scheme outperforms related approaches and accurately tracks multiple targets utilizing only a small percentage of moving sensors that closely follow the targets.

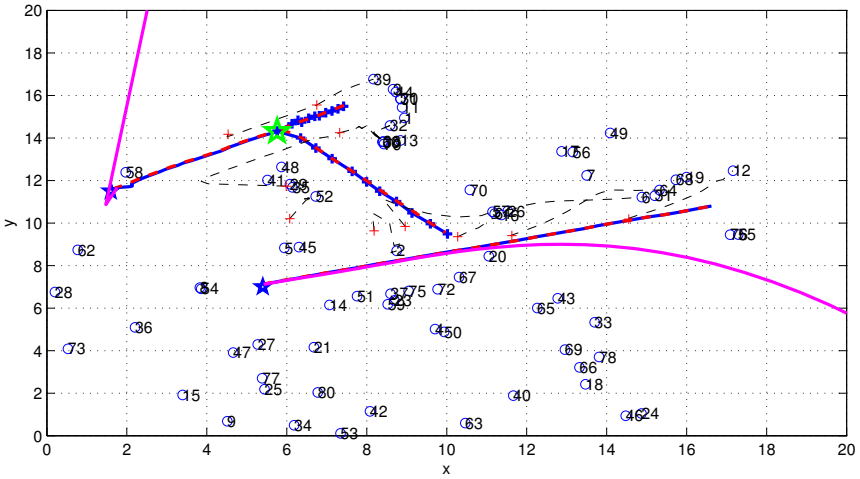

Figure 4.8. Tracking multiple objects .

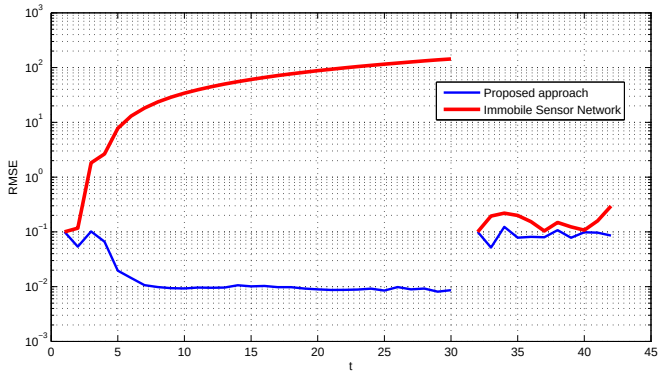

Figure 4.9. Tracking RMSE versus time in the setting in Fig. 4.8..

CHAPTER 5

Regularized Kernel Matrix Decomposition in Thermal Video Multi-Object Tracking

5.1 Problem Setting and Preliminaries

Consider a sequence of frames forming a video in which the frames contain multiple nonstationary/moving objects of interest that need to be detected/identified and tracked. Depending on the model of the used camera, videos may be acquired at different resolution conditions which results in video frames of various dimensionality [96], [11]. Let $\mathbf{F}_t$ the frame of the available video sequence at time instant t of dimensions $f_x \times f_y$. Further, let $x_t^{m,n}(f)$ denote the pixel intensity of the (m,n)-th pixel of frame $\mathbf{F}_t$ at time instant t where $m = 1, \ldots, f_x$ and $n = 1, \ldots, f_y$. For simplicity in exposition $\mathbf{x}_t(\mathbf{f}) \in \mathcal{R}^{p\times 1}$, with $p = f_x \cdot f_y$ denotes a super vector that contains all the pixels of frame $\mathbf{F}_t$ placed in there from top to bottom and left to right. For the sake of simplicity later on we will omit the $\mathbf{f}$ index in $\mathbf{x}_t(\mathbf{f})$.

There is an *unknown* number of objects in the video that we are interested in tracking, and M denotes the maximum number of objects that can be present in a frame. Let $\mathcal{P}_m^t$ denote the set of pixel indices corresponding to the mth object at time instant t, i.e., $\mathcal{P}_m^t := \{[x_{m,1,t}, y_{m,1,t}], \ldots, [x_{m,N_m,t}, y_{m,N_m,t}]\}$ indicate the coordinates of the pixels of the mth object at time instant t. The pixels corresponding to an object at time instant t, say $\mathcal{P}_m^t$, are not known. In order to model the movement of each of the objects we will focus on how the coordinates of the centroid pixel of each object evolve in time. The centroid pixel for the mth object at time instant t is defined as

$$\mathbf{c}_m^t := \lfloor N_m^{-1} \sum_{i=1}^{N_m} [x_{m,n,t}, y_{m,n,t}] \rfloor,$$

with $\lfloor \rfloor$ denoting the floor operator, and intuitively it describes the center point of the mth moving object.

When the video sequence is acquired at a high frame rate, it can be assumed that that objects' centroid pixels move from frame to frame according to a constant velocity moving model [7]. The velocity can be kept constant for a certain number of pixels and then changed if necessary. Specifically, the m-th object's centroid state vector is denoted as $\mathbf{s}_m(t) := [(\mathbf{c}_m^t)^T, \mathbf{v}_m^t]$, where $\mathbf{v}_m^t$ is a 2×1 vector that contains the velocity across the horizontal and vertical axis. The state vector $\mathbf{s}_m(t-1)$ is assumed to evolve according to the following model:

$$\mathbf{s}_m(t) = \mathbf{F}\mathbf{s}_m(t-1) + \mathbf{u}_m(t), \quad m = 1, \ldots, M \tag{5.1}$$

where $\mathbf{F} \in \mathcal{R}^{4\times 4}$ is the state transition matrix, while $\mathbf{u}_m(t)$ denotes zero-mean Gaussian noise with covariance $\mathbf{\Sigma}_u$. The matrices $\mathbf{F}$ and $\mathbf{\Sigma}_u$ have the following structure (e.g., see [7])

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.2}$$

$$\mathbf{\Sigma}_u = \sigma_u^2 \begin{bmatrix} (\Delta T)^3/3 \cdot \mathbf{I}_2 & (\Delta T)^2/2 \cdot \mathbf{I}_2 \\ (\Delta T)^2/2 \cdot \mathbf{I}_2 & \Delta T \cdot \mathbf{I}_2 \end{bmatrix}, \tag{5.3}$$

where ΔT corresponds to the inter-frame interval, σ_u^2 is a nonnegative constant controlling the variance of the noise entries in $\mathbf{u}_m(t)$, while $\mathbf{I}_2$ denotes the 2×2 identity matrix. The pixel coordinates $[x_{m,1,t}, y_{m,1,t}]$ take integer values though the state noise $\mathbf{u}_m(t)$ being Gaussian and subsequently the state can take real values. Though, we can control the state noise standard deviation such that $3\sigma_u$ ($3-\sigma$ bounds) is equal to a small number of pixels

that model any deviation from the constant velocity movement. Thus, the noise will take values within the interval $[-3\sigma_u, 3\sigma_u]$ with probability 99.7%. Despite the fact that the centroid $\mathbf{c}_m(t)$ assumes real values it can model at an acceptable level of accuracy the movement of the video objects from frame to frame.

5.1.1 Kernel-Based Object Pixel Correlations

As stated earlier the pixel corresponding to an object are unknown, thus it is essential to identify the objects before attempting to track them. To cluster the pixels of interest according to an object they belong to, we will utilize statistical correlations that pixels belonging to the same object exhibit. Pixels of an object are expected to have similar intensity (different from the background pixels) which subsequently makes them correlated.

Pixels belonging to the same object exhibit nonlinear dependencies in general [53], thus employing a linear covariance matrix will not identify correlated components. To this end, we account for the nonlinear dependencies in the pixels of frame $\mathbf{F}_t$, namely $\mathbf{x}_t := \text{vec}(\mathbf{F}_t)$ by utilizing nonlinear mappings $\boldsymbol{\phi}_x(\mathbf{x}_t)$ that are applied row-wise across the entries of $\mathbf{x}_t$ and map each pixel in $\mathbf{x}_f$, in a higher dimensional space where linear correlations can be exploited. Specifically, the mapping $\boldsymbol{\phi}_x$ results a $f_x f_y \times D$ matrix

$$\boldsymbol{\phi}_x(\mathbf{x}_t) := [\boldsymbol{\phi}_x(\mathbf{x}_t(1)), \ldots, \boldsymbol{\phi}_x(\mathbf{x}_t(f_x f_y))]^T, \tag{5.4}$$

where D corresponds to the dimensionality of the transformed pixel vector $\boldsymbol{\phi}_x(\mathbf{x}_t(i))$ for $i = 1, \ldots, f_x f_y$.

The nonlinear mapping $\boldsymbol{\phi}_x$ should be selected such that the covariance matrix of the transformed frames exhibits a block diagonal structure. For example Fig. 5.1 display the nonzero entries of kernel (Gaussian kernel was used) covariance matrix obtained from a sequence of frames in which a white object moves in black background. Clearly, the

covariance matrix has a block diagonal structure after properly permuting the rows and columns that contain the nonzero entries.

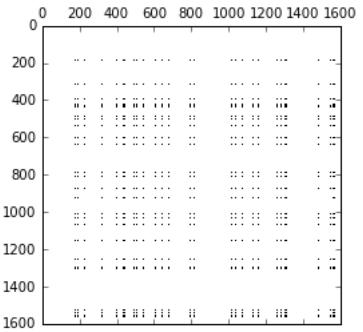

Figure 5.1. Sparse structure of kernel covariance matrix..

Thus, after properly selecting a kernel (see details later on) the covariance of the transformed data $\boldsymbol{\phi}_x(\mathbf{x}_t)$ can be written as

$$\begin{aligned}&\mathbb{E}[(\boldsymbol{\phi}_x(\mathbf{x}_t) - \mathbb{E}[\boldsymbol{\phi}_x(\mathbf{x}_t)])(\boldsymbol{\phi}_x(\mathbf{x}_t) - \mathbb{E}[\boldsymbol{\phi}_x(\mathbf{x}_t)])^T] \\ &= \mathbf{P}_r \text{bdiag}(\mathbf{B}_{1,t}, \mathbf{B}_{2,t}, \ldots, \mathbf{B}_{M,t})\mathbf{P}_c, \end{aligned} \tag{5.5}$$

where $\mathbf{B}_m$ denotes the mth diagonal block of size $N_m \times N_m$ indicating how the N_m pixels of object m are correlated, while pixels belonging to different component are assumed to be uncorrelated. Further, $\mathbf{P}_r$ and $\mathbf{P}_c$ corresponds to arbitrary unknown perturbation matrices of the rows and columns.

The first challenge will be to locate the pixels of each object, which pertains to identifying where the entries of each of the M diagonal blocks are located in the transformed covariance matrix in (5.5), which boils down to estimating the size of each diagonal block N_m, as well as the indices of the pixels that belong to the mth diagonal block. In the following section we will formulate this as a sparse matrix factorization problem, while properly selecting the Gaussian kernel parameters to induce a covariance matrix with block diagonal

structure. Then, once the pixels of an object have been determined we will proceed with tracking the estimated centroid of each of the objects.

5.2 Multi-Object Pixel Clustering

5.2.1 Kernel Covariance Estimation

In order to estimate the covariance matrix of the transformed data in (5.5) we will rely on sample-averaging, where after applying a proper transformation to the pixel vectors to obtain $\boldsymbol{\phi}_x(\mathbf{x}_t)$ we estimate the covariance of the transformed data as follows

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}_{\phi_x,t} &= \frac{1}{F}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}(\boldsymbol{\phi}_x(\mathbf{x}_\tau)-\bar{\boldsymbol{\phi}}_{x,t})\cdot(\boldsymbol{\phi}_x(\mathbf{x}_\tau)-\bar{\boldsymbol{\phi}}_{x,t})^T \\ &= \frac{1}{F}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}[\boldsymbol{\phi}_x(\mathbf{x}_\tau)\boldsymbol{\phi}_x^T(\mathbf{x}_\tau)-\boldsymbol{\phi}(\mathbf{x}_\tau)\bar{\boldsymbol{\phi}}_{x,t}^T \\ &\quad -\bar{\boldsymbol{\phi}}_{x,t}\boldsymbol{\phi}_x^T(\mathbf{x}_\tau)+\bar{\boldsymbol{\phi}}_{x,t}\bar{\boldsymbol{\phi}}_{x,t}^T] \end{aligned} \tag{5.6}$$

where $\bar{\boldsymbol{\phi}}_{x,t} := F^{-1}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}\boldsymbol{\phi}_x(\mathbf{x}_\tau)$ corresponds to the sample-average estimate of the mean of the transformed frame pixels, while F corresponds to the number of frames that the objects are virtually stationary and occupy the same area in the frames. The higher the sampling rate is the larger F can be chosen while assuming the objects are stationary within the time interval $[(t-1)F+1, tF]$.

Applying the kernel trick (assuming a proper nonlinear mapping $\boldsymbol{\phi}_x(\cdot)$ is used; see details in [49,58,95,120]) the inner products involved in calculating the entries of $\boldsymbol{\phi}_x(\mathbf{x}_\tau)\boldsymbol{\phi}_x^T(\mathbf{x}_\tau)$ can be found using a proper positive definite kernel function $K(\mathbf{x}_1(i),\mathbf{x}_2(j))$ whose two arguments correspond to pixels i and j from frame pixel vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively, i.e., the kernel trick implies that the inner product $\langle\boldsymbol{\phi}_x(\mathbf{x}_1(i)),\boldsymbol{\phi}_x(\mathbf{x}_2(j))\rangle$ can be evaluated

from a proper scalar kernel function $K(\mathbf{x}_1, \mathbf{x}_2)$. Utilizing this property the first term in Eq. (5.6) can be rewritten as

$$\frac{1}{F}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}\boldsymbol{\phi}_x(\mathbf{x}_\tau)\boldsymbol{\phi}_x^T(\mathbf{x}_\tau) = \frac{1}{F}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}\mathbf{K}(\mathbf{x}_\tau, \mathbf{x}_\tau), \tag{5.7}$$

where $\mathbf{K}$ is a $f_x f_y \times f_x f_y$ matrix whose (i,j)-th entry is given as $[\mathbf{K}]_{i,j} = K(\mathbf{x}_\tau(i), \mathbf{x}_\tau(j))$. Similarly, the second and third summation terms in Eq. (5.6) can be expressed as

$$\begin{aligned}
&\frac{1}{F}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}\boldsymbol{\phi}_x(\mathbf{x}_\tau)\bar{\boldsymbol{\phi}}_{x,t}^T(\mathbf{x}, t) \\
&= \frac{1}{F}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}\boldsymbol{\phi}_x(\mathbf{x}_\tau)\frac{1}{F}\Sigma_{\tau'=(t-1)\cdot F+1}^{t\cdot F}\boldsymbol{\phi}_x{}^T(\mathbf{x}'_\tau) \\
&= \frac{1}{F}\cdot\frac{1}{F}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}\Sigma_{\tau'=(t-1)\cdot F+1}^{t\cdot F}\boldsymbol{\phi}_x(\mathbf{x}_\tau)\boldsymbol{\phi}_x{}^T(\mathbf{x}'_\tau) \\
&= \frac{1}{F^2}\Sigma_{\tau,\tau'=(t-1)\cdot F+1}^{t\cdot F}\mathbf{K}(\mathbf{x}_\tau, \mathbf{x}_{\tau'}),
\end{aligned} \tag{5.8}$$

while the fourth summation term in Eq. (5.6) gives

$$\begin{aligned}
&\frac{1}{F}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}\bar{\boldsymbol{\phi}}_{x,t}\bar{\boldsymbol{\phi}}_{x,t}^T \\
&= \frac{1}{F^2}\Sigma_{\tau'=(t-1)\cdot F+1}^{t\cdot F}\boldsymbol{\phi}_x(\mathbf{x}_{\tau'})\Sigma_{\tau''=(t-1)\cdot F+1}^{t\cdot F}\boldsymbol{\phi}_x^T(\mathbf{x}_{\tau''}) \\
&= \frac{1}{F^2}\Sigma_{\tau''=(t-1)\cdot F+1}^{t\cdot F}\mathbf{K}(\mathbf{x}_{\tau'}, \mathbf{x}_{\tau''}).
\end{aligned} \tag{5.9}$$

Notice that the matrix in Eq. (5.9) is the same with the one obtained in (5.8), thus the covariance matrix of the transformed data can be calculated with the aid of the kernel function $K(\cdot,\cdot)$ as follows

$$\begin{aligned}
\boldsymbol{\Sigma}_{\boldsymbol{\phi}_x,t} &= \frac{1}{F}\Sigma_{\tau=(t-1)\cdot F+1}^{t\cdot F}\mathbf{K}(\mathbf{x}_\tau, \mathbf{x}_\tau) \\
&\quad - \frac{1}{F^2}\Sigma_{\tau''=(t-1)\cdot F+1}^{t\cdot F}\mathbf{K}(\mathbf{x}_{\tau'}, \mathbf{x}_{\tau''}).
\end{aligned} \tag{5.10}$$

A kernel utilized in image pixel classification successfully [46], [89] is the Gaussian radial basis function (RBF) in which the (i,j) entry of matrix $\mathbf{K}$ used earlier can be expressed as

$$K(\mathbf{x}_\tau(i), \mathbf{x}_\tau(j)) = \exp\left(-\frac{(\mathbf{x}_\tau(i) - \mathbf{x}_\tau(j))^2}{2\sigma^2}\right), \tag{5.11}$$

where the variance σ^2 is a crucial parameter that controls the degree of inter-pixel correlation. Details on how to select this parameters will be given in Sec. 5.5.

5.2.2 Kernel Covariance Sparse Factorization

Given the block diagonal structure that the covariance matrix of the properly transformed data exhibits, we will try to determine sparse factors $\mathbf{g}_1, \ldots, \mathbf{g}_M$ such that $\hat{\mathbf{\Sigma}}_{\boldsymbol{\phi}_x,t} \approx \sum_{m=1}^{M} \mathbf{g}_m \mathbf{g}_m^T$, while the support of each of the factors $\mathbf{g}_m$ will indicate the indices of the entries belonging to a block of correlated pixels in $\hat{\mathbf{\Sigma}}_{\boldsymbol{\phi}_x,t}$ that belong to the same object. The idea of utilizing sparse matrix decomposition to identify correlated data was first proposed in [101] and here it is generalized under the realm of kernel-based nonlinear data transformations.

Single Object:

We start with the case where there is only one moving object in the frames of the video sequence. A standard least-squares based matrix decomposition scheme would minimize the Frobenius norm-based cost $\|\mathbf{\Sigma}_{\boldsymbol{\phi}_\mathbf{x},t} - \mathbf{M}_t\mathbf{M}_t^T\|_F^2$ with respect to the factor estimates $\mathbf{M}_t \in \mathcal{R}^{p\times 1}$. However, such a formulation does not take into account the sparse structure of $\mathbf{M}_t$. To this end, the following minimization framework is proposed:

$$\hat{\mathbf{M}}_t := \arg\min_{\mathbf{M}_t} \|\mathbf{\Sigma}_{\boldsymbol{\phi}_\mathbf{x},t} - \mathbf{M}_t\mathbf{M}_t^T\|_F^2 + \lambda\|\mathbf{M}_t\|_1, \tag{5.12}$$

where the norm-one term $\|\cdot\|_1$ is utilized to induce sparsity in the column vector $\mathbf{M}_t$, see e.g., [114], [137], in the column of $\mathbf{M}_t$ whose support will point to those pixels in a collection of F frames that contain the object of interest within interval $[(t-1)F+1, t\cdot F]$. The parameter λ is the sparsity controlling coefficient that determines the number of zeros in $\mathbf{M}_t$, i.e., the larger λ is, the more zero entries will be contained in the optimal solution $\hat{\mathbf{M}}_t$.

The cost in Eq. (5.12) is nonconvex with respect to (wrt) $\mathbf{M}_t$. To overcome this obstacle an iterative minimization scheme is derived next using coordinate descent strategies [10]. The cost in Eq. (5.12) is minimized recursively wrt one entry of $\mathbf{M}_t$, namely $\mathbf{M}_t(j)$ while keeping all other entries in $\mathbf{M}_t$ fixed to their latest updates.

Minimization of the cost in Eq. (5.12) wrt $\mathbf{M}_t(j)$ while fixing the remaining variables to their latest update during coordinate cycle k gives the following solution for updating $\hat{\mathbf{M}}_t^k(j)$:

$$\hat{\mathbf{M}}_t^k(j) = \arg\min_{\mathbf{M}_t(j)} 2 \cdot \sum_{\mu=1,\mu\neq j}^{p} [\mathbf{\Sigma}_{\boldsymbol{\phi}_x,t}(j,\mu) - \mathbf{M}_t(j)\hat{\mathbf{M}}_t^{k-1}(\mu)]^2 + \lambda|\mathbf{M}_t(j)| + [\mathbf{\Sigma}_{\boldsymbol{\phi}_x,t}(j,j) - \mathbf{M}_t^2(j)]^2. \tag{5.13}$$

Discarding the terms that do not depend on $\mathbf{M}_t(j)$ and applying proper algebraic manipulations, the cost in Eq.(5.13) can be rewritten as:

$$\begin{aligned} J^k(j) = {} & (\mathbf{M}_t(j))^4 + \lambda|\mathbf{M}_t(j)| \\ & + (\mathbf{M}_t(j))^2[2\sum_{\mu=1,\mu\neq j}^{p} [\hat{\mathbf{M}}_t^{k-1}(\mu)]^2 - 2\delta^k(j,j)] \\ & - \mathbf{M}_t(j)[4\sum_{\mu=1,\mu\neq j} \delta^k(j,\mu)\hat{\mathbf{M}}_t^{k-1}(\mu)] \end{aligned} \tag{5.14}$$

where

$$\delta^k(j,\mu) := \mathbf{\Sigma}_{\boldsymbol{\phi}_x,t}(j,\mu) - \hat{\mathbf{M}}_t^{k-1}(j)\hat{\mathbf{M}}_t^{k-1}(\mu) \tag{5.15}$$

for $j,\mu = 1,\ldots,p$. Given the most recent update $\hat{\mathbf{M}}_t^{k-1}$ from coordinate cycle $k-1$, as shown in Appendix A, the update for $\hat{\mathbf{M}}_t^k(j)$ will be the value which achieves the minimum

cost in Eq.(5.14) among the following candidates: i) 0; ii) the real positive roots of the third-degree polynomial:

$$4 \cdot h^3 + 4(\sum_{\mu=1,\mu\neq j} [\hat{\mathbf{M}}_t^{k-1}(\mu)]^2 - \delta^k(j,j)) \cdot h - 4(\sum_{\mu=1,\mu\neq j}^{p} \delta^k(j,\mu)\hat{\mathbf{M}}_t^{k-1}(\mu)) + \lambda = 0 \quad (5.16)$$

iii) the real negative roots of the third-degree polynomial:

$$4 \cdot h^3 + 4(\sum_{\mu=1,\mu\neq j} [\hat{\mathbf{M}}_t^{k-1}(\mu)]^2 - \delta^k(j,j)) \cdot h - 4(\sum_{\mu=1,\mu\neq j}^{p} \delta^k(j,\mu)\hat{\mathbf{M}}_t^{k-1}(\mu)) - \lambda = 0 \quad (5.17)$$

To obtain the roots for the above two third-degree polynomial, we utilized companion matrices, [50]. The proposed sparsity-aware kernel matrix decomposition algorithm is tabulated as Algorithm 1. In fact, convergence to at least a stationary point of the cost in Eq. (5.12) is established in Appendix B.

Algorithm 6 Sparse Kernel Covariance Factorization

1: Using frames within time interval $[(t-1)\cdot F + 1, t \cdot F]$:

2: Form the kernel covariance matrix using Eq. (5.10)

3: Initialize $\mathbf{M}_t(j)$'s as 0's

4: **for** $k = 1, 2, \ldots, \kappa$ **do**

5: Evaluate $\delta^k(j,\mu)$ for $j, \mu = 1, \ldots, p$ via Eq. (5.15).

6: Determine the updates $\{\hat{\mathbf{M}}_t^k(j)\}$ after determining the positive roots of Eq. (5.16) and the negative roots of Eq. (5.17).

7: If $\|\mathbf{M}_t^k - \mathbf{M}_t^{k-1}\| \leq \epsilon$, where ϵ is the desired error threshold then break.

8: **end for**

After determining the sparse factor $\hat{\mathbf{M}}_t^k$, the nonzero entries' indices (support) of $\hat{\mathbf{M}}_t^k$ will point to the moving object pixels within the frame sequence during time interval $[(t-1)F+1, tF]$. Next, we generalize the pixel classification framework in the presence of multiple objects.

5.2.3 Multiple Objects

In the presence of multiple objects in a frame sequence the sparse factorization formulation in Eq. (5.12) can be employed by introducing multiple columns in $\mathbf{M}_t$ and employing the same coordinate descent process described earlier. One challenge in the presence of multiple objects is the correlation among objects that have similar pixel intensities and/or texture. In this case, the sparse factorization framework may return sparse factors $\hat{\mathbf{M}}_t$ that contain nonzero values in entries corresponding to pixels of more than one objects. Thus, it may be necessary to do some extra clustering among these pixels to separate them according to the object they correspond to. This process will enable to split the objects that may appear at the same sparse factor and enable us to track them individually.

To split the objects that may be present in a sparse factor returned by the sparse factorization algorithm we rely on the property that pixels corresponding to the same object present in a sparse factor $\hat{\mathbf{M}}_t$ should be neighboring and thus closer (in terms of Euclidean distance) compared to pixels corresponding a different object (that in general is placed at a different part of the frame).

Let $\mathcal{P}^t$ denote the nonzero entries of $\hat{\mathbf{M}}_t^k$ which indicates the moving objects' pixels, from which the coordinates $\mathbf{z}_i$ for each pixel $i \in \mathcal{P}^t$ can be further extracted. Then, we employ K-means clustering, see [47], aiming at partitioning the $\mathcal{P}^t$ pixels into Z_t clus-

ters $\{\Xi_1, \ldots, \Xi_{Z_t}\}$ according to the similarity their corresponding coordinates $\mathbf{z}_i, i \in \mathcal{P}^t$ exhibit. K-means clusters the pixels by minimizing the following formulation

$$\arg\min_{\Xi} \sum_{j=1}^{Z_t} \sum_{\mathbf{z}_i \in \Xi_j} \|\mathbf{z}_i - \xi_j\|^2 \tag{5.18}$$

where ξ_j corresponds to the mean of cluster Ξ_j

In this way, $\mathcal{P}^t$ pixels will be clustered into Z_t clusters, centered at $\xi_j, j = 1, \ldots, Z_t$ corresponding to the different moving objects contained within a sparse factor obtained via Alg. 1. If two clusters' centroid coordinates are too close to each other:

$$\|\xi_j - \xi_{j'}\|^2 \leq \epsilon_d, \tag{5.19}$$

where ϵ_d is a predefined distance, we will decrease the initial number of clusters to $Z_t - 1$. By setting an upper limit M_{up} on the number of moving objects, we will adjust the required number of clusters Z_t in the aforementioned way. So that eventually Z_t would equal the real but unknown number of objects, here M in the video.

5.3 Frame Object Tracking

Once the pixels $\mathcal{P}_m^t$ corresponding to object m have been determined, the objects centroid pixel can be determined as described earlier and Kalman filtering will be utilized to accurately track the location of each detected object within the video sequence. Recall that the state vector $\mathbf{s}_m(t)$ contains the location coordinates, as well as the velocity at which the object's centroid is moving along each of the two dimensions present in each frame. It should be emphasized that there may be some errors when clustering the pixels according to the objects they belong too, in which case let $\hat{\mathcal{P}}_m^t$ denote the estimated pixel locations corresponds to object m, while $\hat{\mathbf{c}}_m^t$ corresponds to the corresponding estimate of the object's

centroid pixel. The following measurement model can be utilized to associate $\hat{\mathbf{c}}_m^t$ with the state vector $\mathbf{s}_m(t)$ as follows

$$\hat{\mathbf{c}}_m^t = \mathbf{H}(t)\mathbf{s}_m(t) + \mathbf{w}_m(t) = \mathbf{c}_m^t + \mathbf{w}_m(t),\ m = 1, \ldots, M, \tag{5.20}$$

where

$$\mathbf{H}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

while $\mathbf{w}_m(t)$ corresponds to the localization error that may be present in $\hat{\mathbf{c}}_m^t$ when utilizing the sparse factorization approach in Sec. III. It is assumed here that the noise $\mathbf{w}_m(t)$ is zero mean with variance $\sigma_w^2 \cdot \mathbf{I}_{2\times 2}$. After numerical testing, the noise variance is found to lie below 3 pixel distance. Thus, the variance is set as $\sigma_w^2 = 3$.

Although, the distribution of the noise $w_m(t)$ is unknown in (5.20) and not necessarily Gaussian, the Kalman filter will still provide the linear minimum mean-square estimation for the state and observation models in (5.1) and (5.20). The object state estimator and corresponding error covariance matrix, obtained by the Kalman filter for object m are denoted here as $\hat{\mathbf{s}}_m(t|t)$ and $\mathbf{P}_m(t|t)$, respectively. The prediction step in the Kalman filter used here, see e.g. [63], involves the following updating recursions for the state estimator and corresponding covariance at time instant t

$$\hat{\mathbf{s}}_m(t|t-1) = \mathbf{F}\hat{\mathbf{s}}_m(t-1|t-1) \tag{5.21}$$

$$\hat{\mathbf{P}}_m(t|t-1) = \mathbf{F}\hat{\mathbf{P}}_m(t-1|t-1)\mathbf{F}^T + \mathbf{\Sigma}_u. \tag{5.22}$$

The estimated centroid $\hat{\mathbf{c}}_m^t$ will then be used to carry out the correction step of the Kalman filter which involves the following updating recursions:

$$\hat{\mathbf{s}}_m(t|t) = \hat{\mathbf{s}}_m(t|t-1) + \mathbf{G}_m(t) \cdot [\hat{\mathbf{c}}_m^t - \mathbf{H}(t)\hat{\mathbf{s}}_m(t|t-1)] \tag{5.23}$$

$$\mathbf{P}_m(t|t) = (\mathbf{I} - \mathbf{G}_m(t)\mathbf{H}(t))\mathbf{P}_m(t|t-1) \tag{5.24}$$

for $m = 1, \ldots, M$), while the matrix $\mathbf{G}_m(t)$ which corresponds to the Kalman gain can be evaluated as

$$\mathbf{G}_m(t) = \mathbf{P}_m(t|t-1)\mathbf{H}^T(t)(\sigma_w^2 \cdot \mathbf{I}_{2\times 2} + \mathbf{H}(t)\mathbf{P}_m(t|t-1)\mathbf{H}^T(t))^{-1}. \tag{5.25}$$

A separate Kalman filter is implemented for each of the M objects determined using Alg. 1. Each of these filters is using the estimated centroid $\hat{\mathbf{c}}_m^t$ found at every time instant t.

5.4 Real-Time Object Identification and Tracking

5.4.1 Dealing with Large Frames

One may notice that the proposed matrix decomposition scheme may involve high dimension computations in the initialization stage when determining $\mathbf{M}_t \in \mathcal{R}^{p*1}$, especially when the video resolution is very high. To deal with this issue, we resort to a divide and conquer strategy. We split the $f_x f_y \times f_x f_y$ kernel covariance matrix into smaller parts corresponding to smaller regions of a frame with size $\varrho \times \varrho$ where $\varrho \ll f_x f_y$. For each of these smaller regions we obtain $\hat{\mathbf{M}}_t^j$ for $j = 1, \ldots, J$ using Alg. 1 on the kernel covariance matrix $\hat{\mathbf{\Sigma}}_{\boldsymbol{\phi}_{\mathbf{x}^j,t}}$ that corresponds to subframe $\mathbf{x}_j^t$ that corresponds to a smaller region of the frame $\mathbf{x}_t$ at time instant t, and $J = \frac{f_x f_y}{\varrho^2}$. Then, smaller sparse factor $\hat{\mathbf{M}}_t^j$s are stacked as follows

$$\hat{\mathbf{M}}_t = [\{\hat{\mathbf{M}}_t^1\}^T, \ldots, \{\hat{\mathbf{M}}_t^J\}^T]^T, \tag{5.26}$$

to construct the sparse factor $\hat{\mathbf{M}}_t \in \mathcal{R}^{p*1}$ corresponding to the kernel covariance matrix $\hat{\mathbf{\Sigma}}_{\boldsymbol{\phi}_x,t}$ of the entire frames $\mathbf{x}_t$. It is worth noting that $\hat{\mathbf{M}}_t$ is acquired without the need of factorizing the much larger in size $f_x f_y \times f_x f_y$ kernel covariance matrix $\hat{\mathbf{\Sigma}}_{\boldsymbol{\phi}_x,t}$. Proceeding as before, the nonzeros entries of $\hat{\mathbf{M}}_t$ can be utilized to estimate the $\mathcal{P}_t$ object pixels. After implementing the K-means clustering method on the $\mathcal{P}_t$ pixels, cluster centriods will serve as the initialization positions of the objects in the filtering stage. Further, the pixel subsets

$\mathcal{P}_m^t$ are also used to estimate the width and height of a rectangular subframe, say width w_s^0 and height h_s^0, that surrounds the pixels of each moving object in the frame.

5.4.2 Object Identification and Tracking

Next, it is outlined how the Kalman filter and the sparse kernel factorization algorithm interact with each other to track multiple moving objects in a given video sequence. During the start-up stage, a number of F_s frames will be utilized to evaluate the kernel covariance matrix. After applying Alg. 1 using the divide and conquer implementation in Sec. (5.4.1), the nonzero entries in the acquired sparse vector $\hat{\mathbf{M}}_0$ will point to the pixels of the moving objects in the video.

From $\hat{\mathbf{M}}_0$, we can extract the pixels which form the detected moving objects. Here the object size in Sec. (5.4.1), namely w_s^0 and h_s^0 will be rectified further to w_s and h_s which satisfies $w_s \geq w_s^0, \text{mod}(w_s, \varrho) = 0$ and $h_s \geq h_s^0, \text{mod}(h_s, \varrho) = 0$.. This results in a slightly larger sub-region for each object in which smaller region $\varrho \times \varrho$ can be further divided. In the next time instance, we will just incorporate the pixels around the predicted centroid position $\hat{\mathbf{s}}_m(t|t-1)$ from Kalman filter Eq. (5.21) to form the object kernel covariance matrix $\hat{\mathbf{\Sigma}}_{\boldsymbol{\phi}_{\mathbf{x}},t,m}$ following Eq. (5.6) in which $\mathbf{x}$ contains the pixels $x^{i,j}$ with x- and y-coordinates within the intervals

$$
\begin{aligned}
[\hat{s}_m(t|t-1)]_1 - w_s/2 \leq i \leq [\hat{s}_m(t|t-1)]_1 + w_s/2 \\
[\hat{s}_m(t|t-1)]_2 - h_s/2 \leq j \leq [\hat{s}_m(t|t-1)]_2 + h_s/2.
\end{aligned} \tag{5.27}
$$

Similarly to the initialization stage, the divide and conquer implementation is carried out in the kernel covariance matrix $\hat{\mathbf{\Sigma}}_{\boldsymbol{\phi}_{\mathbf{x}},t,m}$ for each object separately to acquire the pixels which corresponds to the moving object m at the current time instant. This approach reduces the computational complexity since each object allocated areas is further split into smaller regions.

To account for newborn or disappearing objects in the video resulting a time-varying number of objects we can periodically generate and factorize the kernel covariance matrix in the entire video frame. The complete algorithm is outline below:

Algorithm 7 Real-Time Multi-Object Identification and Tracking

1: **Start-up stage ($t = 0$)/Reconfiguration ($t\%Tc = 0$):** For F_s consecutive frames, the kernel covariance matrix is formed according to Eq. (5.6) and Algorithm 1 is applied in the entire frames to determine moving objects in the input frame sequence.

2: **for** $t = 1, 2, \ldots,$ **do**

3: Gather frames within time interval $[(t-1) \cdot F + 1, t \cdot F]$

4: Using the F acquired frames form the kernel covariance matrix $\mathbf{\Sigma}_{\boldsymbol{\phi}_x, t, m}$ with pixels in a rectangular region of size $w_s \times h_s$ with centroid of $[\hat{s}_m(t|t-1)]_{1:2}$ for $m = 1, \ldots, M$

5: Apply Algorithm 1 to $\mathbf{\Sigma}_{\boldsymbol{\phi}_x, t, m}$ and acquire the nonzero entries in $\hat{\mathbf{M}}_t^m$

6: Apply Kalman filter for each of the M objects, i.e., Eqs. (5.21)-(5.25) to track each of the M objects' centroid pixel.

7: **end for**

5.5 Synthetic Numerical Tests

The performance of proposed scheme is first tested on a synthetic frame sequence that contains three rectangular objects that move independently of each other. Video background contains randomly generated Gaussian noise with variance 20, while the objects consist of pixel with intensity varying between 240 to 255. The synthetic video contains 60 frames, during which object 1 moves from the left to the right, object 2 moves from the top to the bottom of the frame and object 3 moves from the right to the left. The frame size is 100 by 100, while all the objects are of size 10 by 10. The true coordinates of each object's centroid pixel are recorded as the ground truth to evaluate our proposed tracking scheme.

To select a proper kernel variance, firstly we empirically choose a variance range $[0.001, 1]$, by checking the kernel covariance matrix formed with different variances in the variance range, we select the variance value which manifests the desired block diagonal structure in the kernel covariance matrix. The value is set as $\sigma^2 = 0.01$. Three sample images are given in Figs. 5.2, 5.3, 5.4, with frame indices 20, 40, and 60. The objects are marked out by a dark-colored circle to show how it captures the momentary location of each of the the rectangular moving objects. The tracking root mean squared-error (RMSE) which quantifies the number of pixels by which the estimated centroid pixel coordinates is missing the true centroid of each object is depicted in Fig. 5.5. It is clear that both the proposed tracking scheme and the scheme in [135] localizes the moving objects within 3 pixels of accuracy. Though, the approach in [135] requires prior knowledge of objects' initial coordinates and a proper search window size which our scheme can generate by itself and without the need of any prior information.

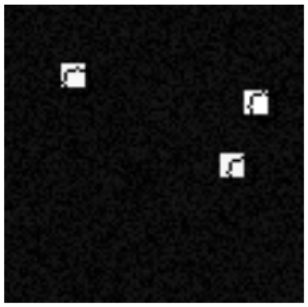

Figure 5.2. Tracking result for frame 24 in synthetic video sequence..

5.6 Multi-Object Detection and Tracking in Thermal Video

Next, the proposed tracking scheme is tested on a video sequences extracted from the datasets available on the OTCBVS website [75], where a Raytheon L-3 Thermal-Eye

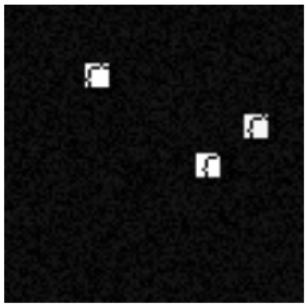

Figure 5.3. Tracking result for frame 44 in synthetic video sequence..

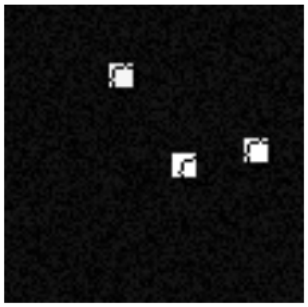

Figure 5.4. Tracking result for frame 64 in synthetic video sequence..

2000AS infrared sensor is utilized to acquire 8-bit grayscale images (with a resolution of 320×240 pixels per frame). The first image sequence is extracted from the OTCBVS dataset 05, i.e, terravic motion infrared database. In this video sequence, a man with a weapon moves from the left to the right with deformation. Six sample frames are presented in Fig. 5.6, 5.7, 5.8. Even though the shape of the target varies with the non-rigid movement of the limbs, our novel algorithm is able to detect the person and tracks the target accurately (the white box surrounding the objects is the estimated area where the algorithm thinks there is an object). Frames 243 and 282 are zoomed in to show the accuracy of the bounding box our proposed tracking scheme generated, see Figs. 5.9, 5.10.

Another experiment is conducted on another video sequence extracted from the OTCBVS database. In this sequence, there are two pedestrians one of which moves from the left to the right, while the other pedestrian moves from the right to the left. Six sample frames are

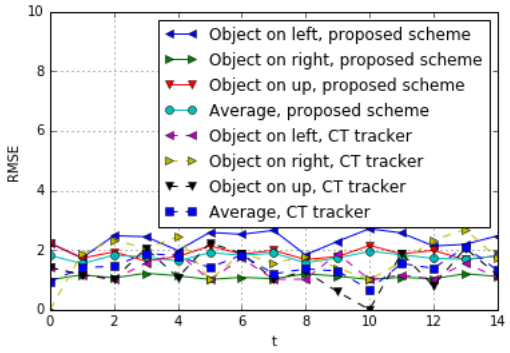

Figure 5.5. Tracking root mean squared-error (RMSE) for the synthetic video sequence..

Figure 5.6. Tracking results for frames 231 and 240 in sequence 1..

displayed in Fig. 5.11, 5.12, 5.13. For better view of how our proposed tracking scheme manages to localize both pedestrians, frames 251 and 360 are zoomed in and displayed in Fig. 5.14, 5.15, 5.16, and5.17. The tracking RMSE for our proposed method and the tracking scheme in [135] is compared in Fig. 5.18. It can be observed that our tracking scheme outperforms the scheme in [135] for both pedestrians, and it is worth noting that for our scheme, the average tracking error for most of the tracking time is below 4 pixels.

5.7 Tracking with Missing Pixels

Oftentimes, videos may be corrupted due to camera or storage issues, resulting missing pixels in the video frames. Here, our tracking scheme is tested in the scenario where a portion of the frame pixels are missing (their intensity is set to 0). Here, two sample frames $224, 252$ with a 5% random pixel loss are provided to display the tracking result. Notice that the tracking performance of the pedestrian on the left is not as good as the pedestrian

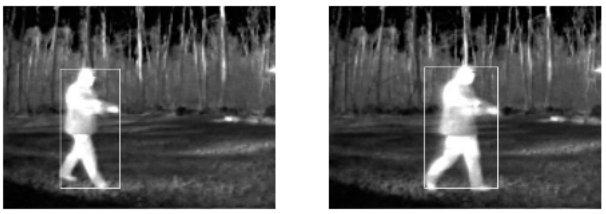

Figure 5.7. Tracking results for frames 243 and 267 in sequence 1..

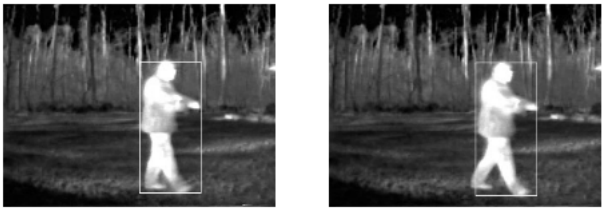

Figure 5.8. Tracking results for frames 282 and 288 in sequence 1..

on the right, since the left sided pedestrian occupies a smaller number of pixels which results a larger portion of the object to disappear in the presence of missing pixels. For the pedestrian on the right, despite the loss of several pixels, our tracking scheme enables precise localization.

5.8 Concluding Remarks

A novel multi-object detection and tracking algorithm was put forth in video sequences. The task of identifying objects in a sequence of frames was transformed in a sparse kernel covariance factorization problem, where the support of the estimated sparse factors point to the pixels of each object present in a frame. To this end, a sparsity-aware kernel covariance matrix factorization scheme, based on norm-1 regularization was proposed and minimized utilizing a coordinate descent approach. After objects are successfully determined, Kalman filtering is implemented cooperatively with the sparse kernel covariance factorization scheme to allow accurate tracking of each object's centroid pixels. Numerical tests on different video datasets validate the effectiveness of our proposed

Figure 5.9. Frames 243 in sequence 1 zoomed in to demonstrate the accurate tracking of the area where the moving object resides in the frame..

Figure 5.10. Frames 282 in sequence 1 zoomed in to demonstrate the accurate tracking of the area where the moving object resides in the frame..

video tracking mechanism in the presence of multiple objects, and corroborate the improved tracking performance over existing alternatives.

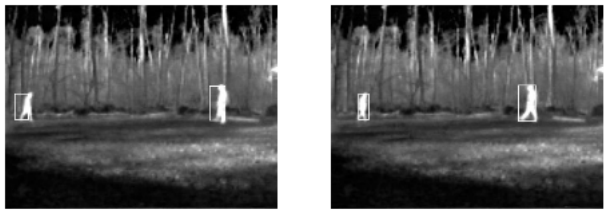

Figure 5.11. Tracking results for frames 240 and 264 in sequence 2..

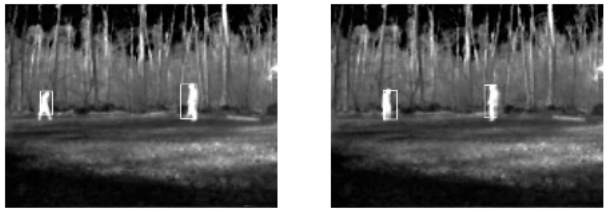

Figure 5.12. Tracking results for frames 282 and 318 in sequence 2..

Figure 5.13. Tracking results for frames 336 and 360 in sequence 2..

Figure 5.14. Frame 251 in sequence 2 zoomed in for pedestrian on the left..

Figure 5.15. Frame 251 in sequence 2 zoomed in for pedestrian on the right..

Figure 5.16. Frame 360 in sequence 2 zoomed in for pedestrian on the left..

Figure 5.17. Frame 360 in sequence 2 zoomed in for pedestrian on the right..

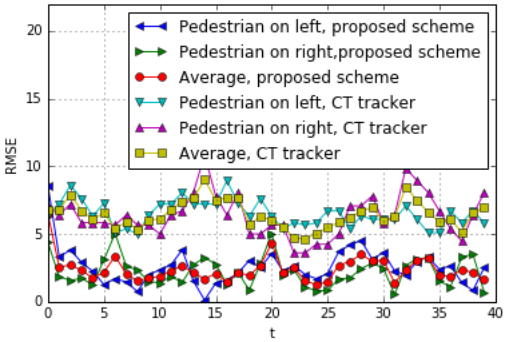

Figure 5.18. OTCBVS Thermal Video tracking RMSE..

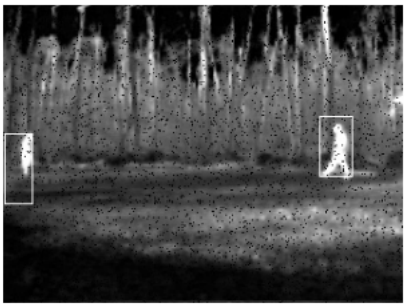

Figure 5.19. Tracking in the presence of missing pixels for frame 224..

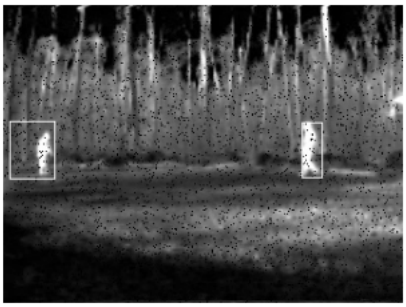

Figure 5.20. Tracking in the presence of missing pixels for frame 252..

CHAPTER 6

FUTURE RESEARCH

6.1 Tracking Multiple Maneuvering Targets

In our previous works, it is assumed that the targets are moving in a constant velocity model. However in real cases, targets may not adhere to a single moving model, possibly targets may move or evolve under different moving models at different time instances, e.g., constant velocity, constant acceleration, constant turn, etc. There are a plethora of works coping with the problem of tracking maneuvering targets. The most popular method is the interacting multiple model method in which several pre-assumed models are mixed together by transitioning from model to model with a transition probability matrix, the state estimator is evaluated as the combination of the state estimators across all the aforementioned pre-assumed model, see in [6, 39, 92]. Limitations of this IMM method involve: i) They do not account for the time instances when exactly the maneuver happens, whereas the maneuver time may be vital in some applications; and ii) Several possible models has to be assumed first, however the assumed models may not cover the real moving trajectories of the targets. To address the aforementioned two issues, we will propose a method which can detect model changes and later make changes to the filtering model accordingly, thus achieving more accurate tracking performance.

6.2 Object Tracking using Multi-modal Video Sequencies

Thermal cameras normally output gray scale imaging, which results a lower data processing complexity, while RGB cameras output triple loads of information that corresponds to red, green, blue color data. Combining both thermal heat features and visible colors from

RGB video could potentially lead to more robust tracking. There are already some research efforts that propose fusion of thermal and RGB visible data, e.g, [48], [45], [34]. The work in [34] relies on the contour saliency map, to fuse together object locations and contours from both thermal and color sensors and eventually extract the object silhouette features, thus obtaining improved tracking performance. However, the method is computationally expensive since it aims at constructing a complete object contour. In [48], data fusion is implemented to fuse thermal and visible data, resulting in an illumination-invariant face image. In the latter work, decision fusion combines the matching score generated from individual face recognition models. We would like to exploit the inter-pixel correlations from both visible and thermal data so as to take advantage of heat and color information of moving objects in a video. Moreover, we are interested developing our thermal video tracking scheme so that it will be capable of handling tough but common tracking scenarios, *e.g.*, object occlusion and scaling.

APPENDIX A

Proof of Eqs. (2.9), (2.10) and (2.12)

Let $\mathbf{M}_t(j,\ell) = y$, while setting the rest minimization variables in (4.8) to their most up-to-date values at the end of cycle $k-1$. It follows that $\hat{\mathbf{M}}_t^k(j,\ell)$ is the minimizer of

$$\arg\min_y y^4 + c_1 \cdot y^2 + c_2 \cdot y + \lambda_\rho t, \;\; \text{s. to } |y| \leq t, \tag{A.1}$$

where

$$c_1 = 2\sum_{i\in\mathcal{N}_j}[\hat{\mathbf{M}}_t^{k-1}(i,\ell)]^2 - 2\zeta_{t,\Sigma}^k(j,j,\ell) + \phi, \text{ and } \quad c_2 = -4\sum_{i\in\mathcal{N}_j}\zeta_{t,\Sigma}^k(j,i,\ell)\hat{\mathbf{M}}_t^{k-1}(i,\ell). \tag{A.2}$$

After evaluating the derivatives of the cost in (G.1) wrt y and t and applying the Karush-Kuhn-Tucker optimality conditions [10] it follows that $y^* := \hat{\mathbf{M}}_t^k(j,\ell)$ should satisfy $4(y^*)^3 + 2c_1y^* + c_2 + \mu_1^* - \mu_2^* = 0$ and $-\mu_1^* - \mu_2^* + \lambda_\ell = 0$, where μ_1^* and μ_2^* are the optimal multipliers corresponding to the inequality constraints of (G.1). Note that $\mu_1^* \geq 0, \mu_2^* \geq 0$, while the complementary slackness conditions impose that $\mu_1^*(y^* - t^*) = \mu_2^*(-t^* - y^*) = 0$. If $y^* > 0$ the slackness conditions imply that $\mu_2^* = 0$ from which it follows that $\mu_1^* = \lambda_\ell$. Substituting the latter values in $4(y^*)^3 + 2c_1y^* + c_2 + \mu_1^* - \mu_2^* = 0$ gives (2.9). Similarly, the negative candidate minimizers of (G.1) can be obtained by the roots of (2.10). Differentiating the cost in (4.8) with respect to $\sigma_{j,t}$ and setting the derivative equal to zero we can obtain (2.12). □

APPENDIX B

Proof of Convergence of Alg. 1 in Ch. 2.2

Let $h(\{\mathbf{M}_t(j,\rho)\}_{j=1,\rho=1}^{m,L}, \{\sigma_{j,t}\}_{j=1}^{m})$ denote the cost given in (4.8) which is defined over $\mathbb{R}^{m(L+1)\times 1}$, and let's define

$$h_0(\{\mathbf{M}_t(j,\rho)\}_{j=1,\rho=1}^{m,L}, \{\sigma_{j,t}\}_{j=1}^{m}) := \sum_{j=1}^{m} \sum_{j'\in\mathcal{N}_j} [\hat{\mathbf{\Sigma}}_{x,t}(j,j') - \sum_{l=1}^{L} \mathbf{M}_t(j,l)\mathbf{M}_t(j',l)]^2 + \phi \sum_{\ell=1}^{L} \|\mathbf{M}_{t,:\ell}\|_2^2.$$

Further, consider the level set

$$\mathcal{H}_t^0 := \{\{\mathbf{M}_t(j,\rho)\}_{j,\rho=1}^{m,L} : h(\{\mathbf{M}_t(j,\rho)\}_{j=1,\rho=1}^{m,L}, \{\sigma_{j,t}\}_{j=1}^{m}) \leq h(\hat{\mathbf{M}}_t^0)\}, \tag{B.1}$$

where $\hat{\mathbf{M}}_t^0$ is the $m \times L$ matrix used to initialize Alg. 1 and selected such that $\|\hat{\mathbf{M}}_t^0\|_1 < \infty$ from which it follows that $h(\hat{\mathbf{H}}^0) < \infty$, while the noise variances $\sigma_{j,t,0} = 0$ for $j = 1, \ldots, m$. Then, from (H.1) and the form of $h(\cdot)$ it follows that the member matrices $\mathbf{M}_t$ of $\mathcal{H}_t^0$ satisfy

$$\sum_{\ell=1}^{L} \sum_{j=1}^{m} \lambda_\ell |\mathbf{M}_t(j,\ell)| \leq h(\hat{\mathbf{M}}_t^0) < \infty.$$

Thus, the set $\mathcal{H}^0$ is closed and bounded (compact). Also, $h(\cdot)$ is continuous on $\mathcal{H}^0$.

Recall that the cost involved in updating $\hat{\mathbf{M}}_t^k(j,\ell)$ can be written as $J_t^k(j,\ell) := y^4 + c_1 y^2 + c_2 y + \lambda_\rho |y|$, [cf. (G.1)]. If $c_2 \neq 0$ then after determining the monotonicity of $J_t^k(j,\ell)$ it follows that it has a unique minimizer. If $c_2 = 0$, then $J^k(j,\rho)$ is symmetric around zero. In that case if $c_1 > 0$ then the unique minimizer of $J_t^k(j,\rho)$ is 0. Though, if $c_1 < 0$ then $J_t^k(j,\rho)$ has two minimizers with the same magnitude but different sign. In that case we can consistently select the positive (or negative) minimizer ensuring a unique minimizer per iteration. Function $h(\cdot)$ satisfies the regularization conditions outlined in [112, (A1)]. In detail, the domain of $h_0(\cdot)$ is formed by matrices whose entries satisfy $\mathbf{M}_t(j,\ell) \in (-\infty, +\infty)$. Then, domain$(h_0) = (-\infty, \infty)^{m(L+1)\times 1}$ is an open set. Further, $h_0(\cdot)$ is Gâteaux differentiable over domain(h_0). The Gâteaux derivative is

$$h_0'(\mathbf{M}; \mathbf{\Delta}_M) := \lim_{\epsilon\to 0} [h_0(\mathbf{M} + \epsilon \mathbf{\Delta}_M) - h_0(\mathbf{M})]/\epsilon. \tag{B.2}$$

After carrying out the necessary algebraic operations it follows readily that $h_0'(\mathbf{M}; \boldsymbol{\Delta}_M)$ exists for all $\boldsymbol{\Delta}_M \in \text{domain}(h_0)$, and it is equal to

$$-2\text{tr}[(\mathbf{E} \odot (\hat{\boldsymbol{\Sigma}}_{x,t} - \mathbf{M}_t\mathbf{M}_t^T))(\mathbf{E} \odot (\mathbf{M}_t\boldsymbol{\Delta}_M^T + \boldsymbol{\Delta}_M\mathbf{M}_t^T)] + \mathbf{1}^T(\mathbf{M}_t \odot \boldsymbol{\Delta}_M)\mathbf{1}.$$

The aforementioned properties ensure convergence of the Alg. 1 iterates to a stationary point of $h(\cdot)$ [112, Thm. 4.1 (c)]. □

APPENDIX C

Derivation for (3.9)

Starting from the cost equation in (3.8), and using a common $\mathbf{H}$ for all $\mathbf{H}_\tau$ as justified earlier the cost can be rewritten as follows:

$$\sum_{\tau=t-K+1}^{t}(\mathbf{x}_\tau-\mathbf{H}\mathbf{s}_\tau)^T\mathbf{\Sigma}_w^{-1}(\mathbf{x}_\tau-\mathbf{H}\mathbf{s}_\tau)+\sum_{\tau=t-K+1}^{t}(\mathbf{s}_\tau-\mathbf{F}\mathbf{s}_{\tau-1})^T\mathbf{Q}^{-1}(\mathbf{s}_\tau-\mathbf{F}\mathbf{s}_{\tau-1})$$

$$+(\mathbf{s}_{t-K}-\hat{\mathbf{s}}_{t-K})^T\mathbf{M}_{t-K|t-K}^{-1}(\mathbf{s}_{t-K}-\hat{\mathbf{s}}_{t-K})+\textstyle\sum_{\rho=1}^{r}\lambda_\rho\|\mathbf{h}_\rho\|_1$$

$$=(\mathbf{s}_{t-K}-\hat{\mathbf{s}}_{t-K})^T\mathbf{M}_{t-K|t-K}^{-1}(\mathbf{s}_{t-K}-\hat{\mathbf{s}}_{t-K})+\textstyle\sum_{\rho=1}^{r}\lambda_\rho\|\mathbf{h}_\rho\|_1$$

$$+(\mathbf{F}\mathbf{s}_{t-K}-\mathbf{s}_{t-K+1})^T\mathbf{Q}^{-1}(\mathbf{F}\mathbf{s}_{t-K}-\mathbf{s}_{t-K+1})+\ldots+(\mathbf{F}\mathbf{s}_{t-1}-\mathbf{s}_t)^T\mathbf{Q}^{-1}(\mathbf{F}\mathbf{s}_{t-1}-\mathbf{s}_t)$$

$$+(\mathbf{x}_{t-K+1}-\mathbf{H}\mathbf{s}_{t-K+1})^T\mathbf{\Sigma}_w^{-1}(\mathbf{x}_{t-K+1}-\mathbf{H}\mathbf{s}_{t-K+1})+\ldots+(\mathbf{x}_t-\mathbf{H}\mathbf{s}_t)^T\mathbf{\Sigma}_w^{-1}(\mathbf{x}_t-\mathbf{H}\mathbf{s}_t)$$

$$=(\mathbf{x}_\alpha-\mathbf{F}_\alpha\mathbf{s}_\alpha)^T\mathbf{Q}_\alpha^{-1}(\mathbf{x}_\alpha-\mathbf{F}_\alpha\mathbf{s}_\alpha)+\textstyle\sum_{\rho=1}^{r}\lambda_\rho\|\mathbf{h}_\rho\|_1 \tag{C.1}$$

where $\mathbf{s}_\alpha$, $\mathbf{x}_\alpha$, $\mathbf{Q}_\alpha$, and $\mathbf{F}_\alpha$ are given in equations (3.9), (3.10), and (3.12), respectively. The gradient of (C.1) w.r.t $\mathbf{s}_\alpha$ is $\nabla_{\mathbf{s}_\alpha}J_{t,K}(\{\mathbf{s}_\tau\}_{\tau=t-K}^{t},\mathbf{H})=-2\mathbf{F}_\alpha^T\mathbf{Q}_\alpha^{-1}(\mathbf{x}_\alpha-\mathbf{F}_\alpha\mathbf{s}_\alpha)$. By setting the derivative to zero, the result in (3.9) follows. □

APPENDIX D

Derivation for (3.15)

When it comes to update matrix $\mathbf{H}_t$, while fixing the state vectors to their most up-to-date values only the first and last term in (3.7) are relevant. The matrix $\mathbf{H}_t$ will be updated by minimizing (3.7) on an entry-wise manner. To this end, the vector $\mathbf{y}_{i,t}^{\varkappa-1}$ in (3.14) is formed to help isolate the optimization variable $h \equiv \hat{\mathbf{H}}(i,\rho)$ from same-row entries which are fixed to their most up-to-date values $\hat{\mathbf{H}}_t^{\varkappa-1}(i,\ell), \ell \neq \rho, \ell = 1,\ldots,r$. Thus, (3.13) is minimized on an entry-by-entry coordinate fashion.

The minimization problem in (3.13) is equal to

$$\hat{\mathbf{H}}_t^{\varkappa}(i,\rho) := \arg\min_{h} \sigma_w^{-2}\|\mathbf{y}_{i,t}^{\varkappa-1} - \check{\mathbf{s}}_{\rho,t}\cdot h)\|_2^2 + \lambda_\rho t, \text{s. to} - t \leq h \leq t \tag{D.1}$$

Introducing the Lagrangian multipliers ν_1 and ν_2, the derivative of the Lagrangian function w.r.t. h is given by

$$\nabla_h \mathcal{L}(h,\nu_1,\nu_2) = 2\sigma_w^{-2} h\check{\mathbf{s}}_{\rho,t}\cdot\check{\mathbf{s}}_{\rho,t}^T - 2\sigma_w^{-2}\mathbf{y}_{i,t}^{\varkappa-1}\cdot\check{\mathbf{s}}_{\rho,t}^T + \nu_1 - \nu_2 \tag{D.2}$$

After applying the KKT optimality conditions [10], it follows readily that the optimal solution of (D.1) is given by (3.15). □

APPENDIX E

Derivation for (3.23)

The cost function in (3.23) can be rewritten as follows:

$$
\begin{aligned}
&\sum_{j=1}^{p}\sum_{\tau=t-K}^{t}(\xi_1(\tau)x_j(\tau)-\xi_1(\tau)\mathbf{H}_{j:}\mathbf{s}_{\tau,j})^T\boldsymbol{\Sigma}_w^{-1}(\xi_1(\tau)\mathbf{x}_j(\tau)-\xi_1(\tau)\mathbf{H}_{j:}\mathbf{s}_{\tau,j})\\
&+\frac{1}{p}\sum_{j=1}^{p}\sum_{\tau=t-K}^{t}(\xi_1(\tau)\mathbf{s}_{\tau,j}-\xi_1(\tau)\mathbf{F}\mathbf{s}_{\tau-1,j})^T\mathbf{Q}^{-1}(\xi_1(\tau)\mathbf{s}_{\tau,j}-\xi_1(\tau)\mathbf{F}\mathbf{s}_{\tau-1,j})\\
&+\frac{1}{p}\sum_{j=1}^{p}\sum_{\tau=t-K}^{t}(\xi_2(\tau)\mathbf{s}_{\tau,j}-\xi_2(\tau)\hat{\mathbf{s}}_{t-K,j})^T\mathbf{M}_{t-K|t-K,j}^{-1}(\xi_2(\tau)\mathbf{s}_{\tau,j}-\xi_2(\tau)\hat{\mathbf{s}}_{t-K,j})\\
&=\sum_{j=1}^{p}\sum_{\tau=t-K}^{t}(\xi_1(\tau)x_j(\tau)-\xi_1(\tau)\mathbf{H}_{j:}\mathbf{s}_{\tau,j})^2\sigma_w^{-2}\\
&+\sum_{j=1}^{p}\sum_{\tau=t-K}^{t}\|p^{-1/2}\mathbf{Q}^{-1/2}(\xi_1(\tau)\mathbf{s}_{\tau,j}-\xi_1(\tau)\mathbf{F}\mathbf{s}_{\tau-1,j})\|_2^2 \qquad \text{(E.1)}\\
&+\sum_{j=1}^{p}\sum_{\tau=t-K}^{t}\|p^{-1/2}\mathbf{M}_{t-K|t-K,j}^{-1/2}(\xi_2(\tau)\mathbf{s}_{\tau,j}-\xi_2(\tau)\hat{\mathbf{s}}_{t-K,j})\|_2^2\\
&=\sum_{j=1}^{p}\sum_{\tau=t-K}^{t}\left\|\begin{bmatrix}\xi_1(\tau)\sigma_w^{-1}x_j(\tau)\\ \xi_1(\tau)(p\mathbf{Q})^{-\frac{1}{2}}\mathbf{F}\mathbf{s}_{\tau-1,j}\\ \xi_2(\tau)p^{-1/2}\mathbf{M}_{t-K|t-K,j}^{-\frac{1}{2}}\hat{\mathbf{s}}_{t-K,j}\end{bmatrix}-\begin{bmatrix}\xi_1(\tau)\sigma_w^{-1}\mathbf{H}_{j:}\\ \xi_1(\tau)(p\mathbf{Q})^{-\frac{1}{2}}\\ \xi_2(\tau)p^{-1/2}\mathbf{M}_{t-K|t-K,j}^{-\frac{1}{2}}\end{bmatrix}\cdot\mathbf{s}_{\tau,j}\right\|_2^2
\end{aligned}
$$

APPENDIX F

Derivation for ADMM updating equations in Ch. 3.3

Differentiating the augmented Lagrangian in (3.27), w.r.t $\mathbf{s}_{\tau,j}$ gives

$$\nabla_{\mathbf{s}_{\tau,j}}\mathcal{L}_a = -2\mathbf{G}_{\tau,j}^T(\boldsymbol{\xi}_{\tau,j} - \mathbf{G}_{\tau,j}\mathbf{s}_{\tau,j}) \tag{F.1}$$
$$+\sum_{j'\in\mathcal{N}_j}(\mathbf{v}_{\tau,j}^{j',k+1}(\kappa) + \boldsymbol{\omega}_{\tau,j}^{j',k+1}(\kappa)) + c\sum_{j'\in\mathcal{N}_j}(\mathbf{s}_{\tau,j} - \mathbf{z}_{\tau,j}^{j',k+1}(\kappa)) + c\sum_{j'\in\mathcal{N}_j}(\mathbf{s}_{\tau,j} - \mathbf{z}_{\tau,j'}^{j,k+1}(\kappa))$$

Setting the latter gradient equal to $\mathbf{0}$ (first-order optimality conditions), it follows for $j = 1,\ldots,p$ and $\tau = t-K,\ldots,t$ that

$$\mathbf{s}_{\tau,j}^{k+1}(\kappa+1) = (2\mathbf{G}_{\tau,j}^T\mathbf{G}_{\tau,j} + 2c|\mathcal{N}_j|\mathbf{I}_{r\times r})^{-1} \tag{F.2}$$
$$\times\,[2\mathbf{G}_{\tau,j}^T\boldsymbol{\xi}_j - \sum_{j'\in\mathcal{N}_j}(\mathbf{v}_{\tau,j}^{j',k+1}(\kappa) + \boldsymbol{\omega}_{\tau,j}^{j',k+1}(\kappa)) + c\sum_{j'\in\mathcal{N}_j}(\mathbf{z}_{\tau,j}^{j',k+1}(\kappa) + \mathbf{z}_{\tau,j'}^{j,k+1}(\kappa))]$$

The gradient of the augmented Lagrangian function in (3.27) w.r.t $\mathbf{z}_{\tau,j}^{j'}$ is

$$\nabla_{\mathbf{z}_{\tau,j}^{j'}}\mathcal{L}_a = c[\mathbf{z}_{\tau,j}^{j'} - \mathbf{s}_{\tau,j}^{k+1}(\kappa+1)] + c[\mathbf{z}_{\tau,j}^{j'} - \mathbf{s}_{\tau,j'}^{k+1}(\kappa+1)] - [\mathbf{v}_{\tau,j}^{j',k+1}(\kappa) + \boldsymbol{\omega}_{\tau,j'}^{j,k+1}(\kappa)]$$

Thus, by setting the latter gradient equal to zero we obtain the updating formula in (3.34) for $j = 1,\ldots,p$, $j' \in \mathcal{N}_j$ and $\tau = t-K,\ldots,t$. □

APPENDIX G

Proof of Eqs. (5.16), (5.17)

Let $\mathbf{M}_t(j) = h$, while setting the rest minimization variables in (5.12) to their most up-to-date values at the end of cycle $k-1$. It follows that $\hat{\mathbf{M}}_t^k(j)$ is the minimizer of

$$\arg\min_h h^4 + c_1 \cdot h^2 + c_2 \cdot h + \lambda t, \quad \text{s. to } |h| \leq t, \tag{G.1}$$

where

$$c_1 = 2\sum_{i,i\neq j}^{p} [\hat{\mathbf{M}}_t^{k-1}(i)]^2 - 2\delta^k(j,j) + \phi, \text{ and} \tag{G.2}$$

$$c_2 = -4\sum_{i,i\neq j}^{p} \delta^k(j,i)\hat{\mathbf{M}}_t^{k-1}(i). \tag{G.3}$$

After evaluating the derivatives of the cost in (G.1) wrt h and t and applying the Karush-Kuhn-Tucker optimality conditions [10] it follows that $h^* := \hat{\mathbf{M}}_t^k(j)$ should satisfy $4(h^*)^3 + 2c_1h^* + c_2 + \mu_1^* - \mu_2^* = 0$ and $-\mu_1^* - \mu_2^* + \lambda = 0$, where μ_1^* and μ_2^* are the optimal multipliers corresponding to the inequality constraints of (G.1). Note that $\mu_1^* \geq 0, \mu_2^* \geq 0$, while the complementary slackness conditions impose that $\mu_1^*(h^* - t^*) = \mu_2^*(-t^* - h^*) = 0$. If $h^* > 0$ the slackness conditions imply that $\mu_2^* = 0$ from which it follows that $\mu_1^* = \lambda$. Substituting the latter values in $4(h^*)^3 + 2c_1h^* + c_2 + \mu_1^* - \mu_2^* = 0$ gives (5.16). Similarly, the negative candidate minimizers of (G.1) can be obtained by the roots of (5.17). □

APPENDIX H

Convergence of Alg. 6 in Ch. 5.2

Let $\ell(\{\mathbf{M}_t(j)\}_{j=1}^p)$ denote the cost in (5.12) which is defined over $\mathbb{R}^{p\times 1}$, and let's define

$$\ell_0(\{\mathbf{M}_t(j)\}_{j=1}^p,) := \sum_{j=1}^p \sum_{j'=1}^p [\hat{\mathbf{\Sigma}}_{\phi_x,t}(j,j') - \mathbf{M}_t(j)\mathbf{M}_t(j')]^2$$

Further, consider the following level set:

$$\mathcal{L}_t^0 := \{\{\mathbf{M}_t(j)\}_j^p : \ell(\{\mathbf{M}_t(j)\}_{j=1}^p) \leq \ell(\hat{\mathbf{M}}_t^0)\}, \tag{H.1}$$

where $\hat{\mathbf{M}}_t^0$ is the $p \times 1$ matrix used to initialize Alg. 1 and selected such that $\|\hat{\mathbf{M}}_t^0\|_1 < \infty$, from which it follows that $h(\hat{\mathbf{H}}^0) < \infty$. Then, from (H.1) and the form of $\ell(\cdot)$ it follows that the member matrices $\mathbf{M}_t$ of $\mathcal{H}_t^0$ satisfy

$$\sum_{j=1}^p \lambda_\ell |\mathbf{M}_t(j)| \leq \ell(\hat{\mathbf{M}}_t^0) < \infty.$$

Thus, the level set $\mathcal{L}^0$ is closed and bounded (compact). Also, $\ell(\cdot)$ is continuous on $\mathcal{L}^0$. Recall from [cf. (G.1)] that the cost involved in updating $\hat{\mathbf{M}}_t^k(j)$ can be written as $J_t^k(j) := h^4 + c_1 h^2 + c_2 h + \lambda|h|$. If $c_2 \neq 0$ then after determining the monotonicity of $J_t^k(j)$, it has a unique minimizer. And if $c_2 = 0$, then $J^k(j)$ is symmetric around zero. In that case if $c_1 > 0$ then the unique minimizer of $J_t^k(j)$ is 0. Though, if $c_1 < 0$ then $J_t^k(j)$ has two minimizers with the same magnitude but different sign. In that case we can consistently select the positive (or negative) minimizer ensuring a unique minimizer per iteration. Function $\ell(\cdot)$ satisfies the regularization conditions outlined in [112, (A1)]. In detail, the domain of $\ell_0(\cdot)$ is formed by matrices whose entries satisfy $\mathbf{M}_t(j) \in (-\infty, +\infty)$. Then, domain$(\ell_0) = (-\infty, \infty)^{p\times 1}$ is an open set. Further, $\ell_0(\cdot)$ is Gâteaux differentiable over domain(ℓ_0). The Gâteaux derivative is

$$\ell_0'(\mathbf{M}; \mathbf{\Delta}_M) := \lim_{\epsilon\to 0} [\ell_0(\mathbf{M} + \epsilon\mathbf{\Delta}_M) - \ell_0(\mathbf{M})]/\epsilon. \tag{H.2}$$

After carrying out the necessary algebraic operations it follows readily that $\ell_0'(\mathbf{M}; \mathbf{\Delta}_M)$ exists for all $\mathbf{\Delta}_M \in$ domain(ℓ_0), and it equals

$$-2\text{tr}[(\hat{\mathbf{\Sigma}}_{x,t} - \mathbf{M}_t\mathbf{M}_t^T)(\mathbf{M}_t\mathbf{\Delta}_M^T + \mathbf{\Delta}_M\mathbf{M}_t^T)] + \mathbf{1}^T(\mathbf{M}_t \odot \mathbf{\Delta}_M)\mathbf{1}.$$

The aforementioned properties ensure that Alg. 1 iterates to converge to a stationary point of $\ell(\cdot)$ [112, Thm. 4.1 (c)]. $\square$

REFERENCES

[1] N. Ahmed, M. Rutten, T. Bessell, S. S. Kanhere, N. Gordon, and S. Jha, "Detection and Tracking Using Particle-Filter-Based Wireless Sensor Networks," *IEEE Trans. on Mobile Computing,* vol. 9, no. 9, pp. 1332–1345, Sep. 2010.

[2] A. Gorji and M. B. Menhaj, "Multiple Target Tracking for Mobile Robots Using the JPDAF Algorithm," *Proc. of IEEE Intl. Conf. on Tools with Artificial Intelligence,* Patras, Greece, Oct. 2007, pp. 137–145.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks,* Elsevier, vol. 38, pp. 393–422, 2002.

[4] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice Hall, Englewood Cliffs, NJ; 1979.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Proc.*, vol. 50, no. 2, 174–188, 2002.

[6] S. M. Aly, R. E. Fouly, and H. Braka. "Extended Kalman filtering and Interacting Multiple Model for tracking maneuvering targets in sensor netwotrks." Intelligent solutions in Embedded Systems, 2009 Seventh Workshop on. IEEE, 2009.

[7] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. New York: Wiley, 2001.

[8] M. Baum and U. D. Hanebeck, "The Kernel-SME Filter for Multiple Target Tracking," *Proc. of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, July 2013, pp. 288–295.

[9] M. Baum and U. D. Hanebeck,"Extended Object Tracking Based on Set-Theoretic and Stochastic Fusion," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3103–3115, Oct. 2012.

[10] D. P. Bertsekas, *Nonlinear Programming*, Second Edition, Athena Scientific, 2003.

[11] R. G Baraniuk, T. Goldstein, A.C. Sankaranarayanan, C. Studer, A. Veeraraghavan and M.B. Wakin, "CS-Video: Algorithms, Architectures, and Applications for Compressive Video Sensing," *IEEE Signal Processing Magazine,* vol. 34, no .1, pp. 52–66, 2017.

[12] M. J. Black, and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[13] L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," *Advances in Neural Information Processing Systems,* Vancouver, CAN, 2010.

[14] I. K. Chen, S. L. Hsu, C.Y. Chi, and L.G. Chen, "Automatic video segmentation and object tracking with real-time rgb-d data," *Proc. of the IEEE Intl. Conf. on Consumer Electronics,* Las Vegas, NV, pp. 486-487, 2014.

[15] D P. Bertsekas, and J N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* vol. 23. Englewood Cliffs, NJ: Prentice hall, 1989.

[16] S. Boyd, and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2009.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." *Foundations and Trends in Machine Learning 3*, no. 1, pp. 1–122, 2011.

[18] J. Borenstein, L. Feng, and H. R. Everett. Navigating mobile robots: Systems and techniques. AK Peters, Ltd., 1996.

[19] B. Barshan, and H. F. Durrant-Whyte. "Inertial navigation systems for mobile robots." IEEE Transactions on Robotics and Automation 11.3 (1995): 328-342.

[20] F. S. Cattivelli., and A. H. Sayed. "Diffusion strategies for distributed Kalman filtering and smoothing." IEEE Transactions on automatic control 55, no. 9 (2010): 2069-2084.

[21] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray. "On a decentralized active sensing strategy using mobile sensor platforms in a network." In Decision and Control, 2004. CDC. 43rd IEEE Conference on, vol. 2, pp. 1914-1919. IEEE, 2004.

[22] A. S. Charles and C. J. Rozell, "Dynamic Filtering of Sparse Signals using Reweighted ℓ_1," *Proc. of the Intl. Confe. on Acoust., Speech and Sig. Proc.,* Vancouver, CAN, pp. 6451–6455, May 2013.

[23] S. Cui, J. J. Xiao, A. J. Goldsmith, Z. Q. Luo, and H. V. Poor, "Estimation Diversity and Energy Efficiency in Distributed Sensing." *IEEE Transactions on Signal Processing*, vol. 55,no. 9, pp. 4683-4695, 2007.

[24] G. Cielniak, T. Duckett, and A. J. Lilienthal, "Improved data association and occlusion handling for vision-based people tracking by mobile robots," *Proc. of IEEE Intl. Conf. on Intelligent Robots and Systems*, San Diego, CA, 2007.

[25] E. Candès, J. Romberg, and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information," *IEEE Trans. on Info. Theory*, pp. 489–509, Feb. 2006.

[26] M. Coates, "Distributed Particle Filters for Sensor Networks," *Proc. of the 3rd Intl. Symposium on Information Proc. in Sensor Networks (IPSN 04)*, pp. 99-107, April 2004.

[27] V. P. Dayan and K. N. Vijeyakumar, "Target Tracking in Sensor Networks Using Energy Efficient Prediction Based Clustering Algorithm", *Procedia Engineering*, vol. 38, pp. 2070–2076, 2012.

[28] S. S. Dias and M. G. S. Bruno, "Cooperative Target Tracking using Decentralized Particle Filtering and RSS Sensors," *IEEE Trans. on Sig. Proc.,* vol. 61, no. 14, pp. 3632–3646, July 2013.

[29] P. Djuric, J. H. Kotecha, J. Zhang, Y. Hang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle Filtering," *IEEE Signal Processing Magazine,* vol. 20, no. 5, pp. 19–38, Sep. 2003.

[30] A. Doucet, N. Freitas, and N. Gordon (eds.), *Sequential Monte Carlo Methods in Practice*, Springer, NY, 2001.

[31] A. Doucet, B.-N. Vo, C. Andrieu, and M. Davy, "Particle Filtering for Multi-Target Tracking and Sensor Management," *Proc. of the Fifth Intl. Conf. on Information Fusion*, Annapolis, Maryland, July 2002, pp. 474–481.

[32] K. Dogancay, "UAV path planning for passive emitter localization." IEEE Transactions on Aerospace and Electronic systems 48.2 (2012): 1150-1166.

[33] K. Dogancay. "Online optimization of receiver trajectories for scan-based emitter localization." IEEE Transactions on Aerospace and Electronic Systems 43, no. 3 (2007).

[34] J. W. Davis and V. Sharma, "Background-subtraction using contour-based fusion of thermal and visible imagery," *Computer Vision and Image Understanding*, vol. 106, no. 2 pp. 162–182, 2007.

[35] R. D. Ennulat, and D. Pommerrenig, "Uncooled high resolution infrared imaging plane," U.S. Patent No. 4,754,139. 28 Jun. 1988.

[36] A. M. Elmogy, and F. O. Karray. "COOPERATIVE MULTI TARGET TRACKING USING MULTI SENSOR NETWORK." Int. J. Smart Sens. Intell. Syst 1, no. 3 (2008): 716-734.

[37] Y. Fu, and L. Yang. "Sensor mobility control for multitarget tracking in mobile sensor networks." International Journal of Distributed Sensor Networks 10.3 (2014): 278179.

[38] J. A. Fuemmeler and V. V. Veeravalli, "Energy Efficient Multi-Object Tracking in Sensor Networks," *IEEE Trans. on Sig. Proc.*, vol. 58, no. 7, pp. 3742–3750, 2010.

[39] A. F. Genovese "The interacting multiple model algorithm for accurate state estimation of maneuvering targets." Johns Hopkins APL technical digest 22.4 (2001): 614-623.

[40] A. Goldsmith, "Wireless Communications," *Cambridge University Press,* 2005.

[41] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proc. for Radar and Sig. Processing,* vol 140, no. 2, pp. 107–113, 1993.

[42] A. Gorji and M. B. Menhaj, "Multiple Target Tracking for Mobile Robots Using the JPDAF Algorithm," *Proc. of IEEE Intl. Conf. on Tools with Artificial Intelligence,* Patras, Greece, Oct. 2007, pp. 137–145.

[43] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An Optimal Gradient Method for Nonnegative Matrix Factorization," *IEEE Trans. on Sig. Proc.*, vol. 60, no. 6, pp. 2882–2898, June 2012.

[44] G. H. Golub and C. F. V. Loan, *Matrix Computations,* 3rd ed. Baltimore, MD: The John Hopkins Univ. Press, 1996.

[45] M. Hanif and U. Ali, "Optimized visual and thermal image fusion for efficient face recognition," *Proc. of IEEE Intl. Conf. on Information Fusion*, Florence, Italy, 2006.

[46] Z. Harchaoui and F. Bach, "Image classification with segmentation graph kernels," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.

[47] J. A. Hartigan, and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics),* vol. 28, no. 1, pp. 100–108,1979.

[48] J. Heo, S. G. Kong, B. R. Abidi, and M. A. Abidi, "Fusion of visual and thermal signatures with eyeglass removal for robust face recognition," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition Workshop*, Washington, DC, pp. 122–122, 2004.

[49] T. Hofmann, B. Schölkopf, and A. J. Smola. "Kernel methods in machine learning," *The Annals of Statistics,* pp. 1171-1220, 2008.

[50] R. A. Horn and C. R. Johnson, Matrix Analysis. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[51] P. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," *Journal of Machine Learn. Res.*, vol. 5, pp. 1457–1469, 2004.

[52] C. Hue, J.-P. Le Cadre, and P. Perez, "Tracking Multiple Objects with Particle Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 791–812, 2002.

[53] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. "Crisp boundary detection using pointwise mutual information." In European Conference on Computer Vision, pp. 799-814. Springer International Publishing, 2014.

[54] F. Iutzeler, P. Ciblat, and J. Jakubowicz, "Analysis of Max-Consensus Algorithms in Wireless Channels," *IEEE Trans. on Signal Processing,* vol. 60, no. 11, pp. 6103–6107, 2012.

[55] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation,"*Proc. of the IEEE,* vol. 92, no.3, pp. 401–422, 2004.

[56] F. Jiang, J. Chen, and A. L. Swindlehurst, "Optimal Power Allocation for Parameter Tracking in a Distributed Amplify-and-Forward Sensor Network". arXiv preprint arXiv: 1309.3591, 2013.

[57] M. Jiang, Z. Pan, and Z. Tang, "Visual object tracking based on Cross-modality Gaussian-Bernoulli deep Boltzmann machines with RGB-D sensors," *Sensors*, vol. 121, 2017.

[58] N. Kwak, "Kernel discriminant analysis for regression problems," *Pattern Recognition*, vol. 45, no. 5 pp. 2019–2031,2012.

[59] D. Y. Kim and M. Jeon,"Robust Distributed Kalman Filter for Wireless Sensor Networks with Uncertain Communication Channels," *Mathematical Problems in Engineering,* vol. 2012, Article ID 238597.

[60] E. Karseras, K. Leung and W. Dai, "Tracking Dynamic Sparse Signals using Hierarchical Bayesian Kalman Filters," *Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP),* Vancouver, CAN, pp. 6546–6550, May 2013.

[61] K. Kang and V. Maroulas, "Drift Homotopy Methods for a NonGaussian Filter," *Proc. of the 16th Intl. Conf. on Information Fusion (FUSION)*, pp. 1088–1094, 2013.

[62] B. Kailkhura, S. Brahma, and P. K. Varshney. "Consensus based Detection in the Presence of Data Falsification Attacks." arXiv preprint arXiv:1504.03413 (2015).

[63] S. M. Kay, *Fundamental of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1993.

[64] V. Krishnamurthy, M. Maskery, and G. Yin, "Decentralized Adaptive Filtering Algorithms for Sensor Activation in an Unattended Ground Sensor Network," *IEEE Trans. on Sig. Proc.*, vol. 56, pp. 6086–6101, 2008.

[65] G. Y. Keung, B. Li, Q. Zhang, and H. Yang. "The target tracking in mobile sensor networks." In Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, pp. 1-5. IEEE, 2011.

[66] H. M. La, and W. Sheng. "Dynamic target tracking and observing in a mobile sensor network." Robotics and Autonomous Systems 60.7 (2012): 996-1009.

[67] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman. ”Decentralized environmental modeling by mobile sensor networks.” IEEE Transactions on Robotics 24, no. 3 (2008): 710-724.

[68] D. D. Lee and H. S. Seung, “Algorithms for Non-negative Matrix Factorization,” *Proc. of 13th Conference on Advanced Neural Information Processing Systems (NIPS)*, Denver, CO, 2000, pp. 556–562.

[69] C. J. Lin, “Projected Gradient Methods for Nonnegative Matrix Factorization,” *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[70] J. Lin, W. Xiao, F. L. Lewis, and L. Xie, “Energy-Efficient Distributed Adaptive Multisensor Scheduling for Target Tracking in Wireless Sensor Networks,” *IEEE Tr. on Instr. and Meas.*, vol. 58, pp. 1886–1896, 2009.

[71] J. S. Liu and R. Chen, “Sequential Monte Carlo Methods for Dynamic Systems,” *Journal of the American Stat. Association*, vol. 93, no. 443, pp. 1032-1044, 1993.

[72] H. Q. Liu, H. C. So, F. K. W. Chan, and K. W. K. Lui, “Distributed Particle Filter for Target Tracking in Sensor Networks,” *Progress In Electromagnetics Research C*, vol. 11, pp. 171-182, 2009.

[73] X. Lihua, S. Yeng-Chai, and C. E. De Souza, “Robust Kalman Filtering for Uncertain Discrete-Time Systems,” *IEEE Trans. on Automatic Control,* vol. 39, no. 6, pp. 1310–1314, Jun. 1994.

[74] M. Luber, L. Spinello, and K. O. Arras, “People tracking in RGB-D data with on-line boosted target models,” *Proc. of IEEE Intl. Conf. on Intelligent Robots and Systems*, San Fransisco, CA, 2011.

[75] IEEE OTCBVS WS Series Bench; Roland Miezianko, Terravic Research Infrared Database. Available: http://vcipl-okstate.org/pbvs/bench/

[76] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*, Artech House Publishers MA, 2007.

[77] R. P. S. Mahler, "Multitarget Bayes Filtering via First-order Multitarget Moments," *IEEE Trans. Aero. Elect. Sys.*, vol. 39, no. 4, pp. 1152–1178, 2003.

[78] V. Maroulas and P. Stinis, "Improved Particle Filters for Multi-Target Tracking," *Journal of Computational Physics*, vol. 231, no. 2, pp. 602–611, 2012.

[79] W. Ng, J. Li, S. Godsill, and J. Vermaak, "A Hybrid Approach for Online Joint Detection and Tracking for Multiple Targets," *Proc. of the IEEE Aerospace Conference*, Big Sky, MT, March 2005, pp. 2126–2141.

[80] T. Mukai, and M. Ishikawa. "An active sensing method using estimated errors for multisensor fusion systems." IEEE Transactions on Industrial 43.3 (1996): 380-386.

[81] Y. Oshman, and P. Davidson. "Optimization of observer trajectories for bearings-only target localization." IEEE Transactions on Aerospace and Electronic Systems 35.3 (1999): 892-902.

[82] R. Olfati-Saber, "Distributed Kalman Filter with Embedded Consensus Filters," *Proc. 44th Conf. Dec., the Eur. Control Conf.*, Seville, Spain, Dec. 2005, pp. 8179–8184.

[83] S. Oh, "A Scalable Multi-Target Tracking Algorithm for Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, 2012.

[84] O. Ozdemir, R. Niu, and P. K. Varshney, "Tracking in Wireless Sensor Networks Using Particle Filtering: Physical Layer Considerations," *IEEE Trans. on Signal Processing*, vol. 57, no. 5, pp. 1987–1999, May 2009.

[85] S. Oh, S. Russell, and Shankar Sastry, "Markov Chain Monte Carlo Data Association for Multi-Target Tracking," *IEEE Transactions on Automatic Control*, vol. 54, pp. 481–497, 2009.

[86] Y. Peng, R. A. Freeman, and K. M. Lynch. "Distributed cooperative active sensing using consensus filters." Robotics and Automation, 2007 IEEE International Conference on. IEEE, 2007.

[87] J. G. Proakis, M. Salehi. "Digital Communications 5e." (2007).

[88] C. N. Padole, and L. A. Alexandre, "Motion based particle filter for human tracking with thermal imaging," *Proc. of IEEE Intl. Conf. on Emerging Trends in Engineering and Technology (ICETET)*, Goa, India, pp. 158–162, 2010.

[89] J. Peng, Y. Zhou, and CL. P. Chen, "Region-kernel-based support vector machines for hyperspectral image classification" *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no .9, pp. 4810–4824, 2015.

[90] W. J. Qi, P. Zhang and Z. L. Deng,"Weighted Fusion Robust Steady-State Kalman Filters for Multisensor System with Uncertain Noise Variances," *Journal of Applied Mathematics,* vol. 2014, Article ID 369252.

[91] G. Ren and I. D. Schizas, "Distributed Sensor-Informative Tracking of Targets," *Proc. of the IEEE Intl. Workshop on Computational Advances in Multi-Sensor Adaptive Processing,* Saint Martin, Dec. 15-18, 2013.

[92] Y. Ruan, and L. Hong. "Use of the interacting multiple model algorithm with multiple sensors." Mathematical and computer modelling 44.3 (2006): 332-341.

[93] G. Ren, V. Maroulas, and I. D. Schizas. "Distributed spatio-temporal association and tracking of multiple targets using multiple sensors." IEEE Transactions on Aerospace and Electronic Systems 51, no. 4 (2015): 2570-2589.

[94] G. Ren and I. D. Schizas, "Joint Sensors-Sources Association and Tracking under Power Constraint,"*Proc. of the IEEE Global Conf. on Signal and Information Processing,* Atlanta, GA, pp. 754–758, Dec. 2014.

[95] R. Rosipal, M. Girolami, L. J. Trejo, and A. Cichocki,"Kernel PCA for feature extraction and de-noising in nonlinear regression," *Neural Computing and Applications,* vol. 10, no. 3, pp. 231-243, 2001.

[96] A. C. Sankaranarayanan, L. Xu, C. Studer, Y. Li, K. F. Kelly, and R. G. Baraniuk, "Video compressive sensing for spatial multiplexing cameras using motion-flow models," *SIAM Journal on Imaging Sciences*, vol. 8, no. 3, pp. 1489–1518, 2015.

[97] I. D. Schizas, A. Ribeiro, and G B. Giannakis. "Consensus in Ad Hoc WSNs with Noisy LinksPart I: Distributed Estimation of Deterministic Signals." *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008.

[98] R. F. Souto, J. Y. Ishihara, and G. Araujo Borges, "A Robust Extended Kalman Filter for Discrete-Time Systems with Uncertain Dynamics, Measurements and Correlated Noise," *Proc. of the 2009 American Control Conference*, St. Louis, MS, pp. 1888–1893, June 2009.

[99] N. F. Sandell and R. Olfati-Saber, "Distributed Data Association for Multi-Target Tracking in Sensor Networks," *IEEE Conference on Decision and Control*, Cancun, Mexico, pp. 1085–1090, 2008.

[100] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[101] I. D. Schizas, "Distributed Informative-Sensor Identification via Sparsity-Aware Matrix Factorization," *IEEE Trans. on Sig. Proc.*, vol. 61, no. 18, pp. 46104624, Sep. 2013.

[102] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to High-dimensional Particle Filtering," *Mon. Wea. Rev.*, vol. 136, pp. 4629–4640, 2008.

[103] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance.* Prentice Hall, 1995.

[104] P. Stinis, "Conditional Path Sampling for Stochastic Differential Equations by Drift Relaxation," *Comm. Appl. Math. Comput. Sci.* vol. 6, no. 1, 63–78, 2011.

[105] R. Tharmarasa, T. Kirubarajan, A. Sinha, and T. Lang, "Decentralized Sensor Selection for Large-Scale Multisensor-Multitarget Tracking," *IEEE Trans. on Aerospace and Electronic Systems,* vol. 47, no. 2, pp. 1307–1324, April 2011.

[106] A. J. Sidi, and A. J. Weiss. "Tracking of a moving emitter based on delay and Doppler shift using a particle filter." Electrical and Electronics Engineers in Israel (IEEEI), 2010 IEEE 26th Convention of. IEEE, 2010.

[107] R. Silva, J. S. Silva, and F. Boavida. ”Mobility in wireless sensor networksSurvey and proposal.” Computer Communications 52 (2014): 1-20.

[108] A. Teichman, J. T. Lussier, and S. Thrun. ”Learning to Segment and Track in RGBD.” IEEE Transactions on Automation Science and Engineering 10.4 (2013): 841-852.

[109] A. Treptow, G. Cielniak, and T. Duckett, “Active people recognition using thermal and grey images on a mobile security robot,” *Proc. of IEEE Intl. Conf. on Intelligent Robots and Systems*, Edmonton, CAN, 2005.

[110] M. Toulouse, B. Q. Minh, and P. Curtis. ”A consensus based network intrusion detection system.” IT Convergence and Security (ICITCS), 2015 5th International Conference on. IEEE, 2015.

[111] R. Tibshirani, “Regression Shrinkage and Selection via the Lasso,” *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[112] P. Tseng, “Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization,” *Journal of Opt. Theory and Applications*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

[113] Y. Tian and H. Qi, “Target Detection and Classification Using Seismic Signal Processing in Unattended Ground Sensor Systems,”*Proc. of the Intl. Conf. on Acoust., Speech and Sig. Proc.,* Orlando, FL, 2002.

[114] R. Tibshirani, “Regression Shrinkage and Selection via the Lasso,” *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[115] M. O. Ulfarsson and V. Solo, “Sparse Variable PCA Using Geodesic Steepest Descent,” *IEEE Transactions on Signal Processing*, vol. 10, no. 12, pp. 5823–5832, 2008.

[116] T. G. Gutowski and C. L. Dym, “Propagation of Ground Vibrations: A Review,” *Journal of Sound and Vibration,* vol. 49, no. 2, pp. 179–193, 1976.

[117] D. Li, K. D. Wong, Y. H. Hu and A. M. Sayeed, "Detection, Classification, and Tracking of Targets," *IEEE Sig. Proc. Mag.,* vol. 19, no. 2, pp. 17–29, March 2002.

[118] J. Vermaak, S. J. Godsill, and P. Perez, "Monte Carlo Filtering for Multi Target Tracking and Data Association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no, 1, pp. 309–332, 2005.

[119] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo Methods for Multi-Target Filtering with Random Finite Sets," *IEEE Trans. Aero. Elect. Sys.*, vol. 41, no. 4, pp. 1224–1245, 2005.

[120] J. P. Vert, K. Tsuda, and B. Schölkopf, "A primer on kernel methods," *Kernel Methods in Computational Biology*, pp. 35–70, 2004.

[121] L. Wang, Y. Zhao, W. Wang, "Study on Theory of Multi-target Tracking and Data Association Algorithms in Phased Array Radar,"*Proc. of 6th Intl. Conf. on ITS,* Chengdu, CH, pp. 1232–1235, 2006.

[122] N. Vaswani, "Kalman Filtered Compressed Sensing," *Proc. of Intl. Conf. on Imag. Proc.,* San Diego, CA, pp. 893896, 2008.

[123] X. Wang, D. Musicki, R. Ellem, and F. Fletcher, "Efficient and Enhanced Multi-Target Tracking with Doppler Measurements," *IEEE Trans. on Aerospace and Electronic Letters,* vol. 45, no. 4, pp. 1400–1417, Oct. 2009.

[124] L. Xie, Y. C. Soh, "Robust Kalman Filtering for Uncertain Systems," *Systems & Control Letters*, vol. 22, no. 2, pp. 123–129, Feb. 1994.

[125] L. Xiao, and S. Boyd. "Fast linear iterations for distributed averaging." Systems & Control Letters 53.1 (2004): 65-78.

[126] Y. Xia and H. Jingqing, "Robust Kalman Filtering for Systems Under Norm Bounded Uncertainties in all System Matrices and Error Covariance Constraints," *Journal of Systems Science and Complexity,* vol. 18, no. 4, pp. 339–445, Oct. 2005.

[127] F. Xu, X. Liu, and K. Fujimura, "Pedestrian detection and tracking with night vision," *IEEE Transactions on Intelligent Transportation Systems*, pp. 63–71,2005.

[128] M. Yasuno, N. Yasuda, and M. Aoki, "Pedestrian detection and tracking in far infrared images," *IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria, pp. 131–136, 2005.

[129] M. Yuan, and Y. Lin. "Model selection and estimation in regression with grouped variables." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68, no. 1 (2006): 49-67.

[130] W. Yi, M. Morelande, L. Kong, and J. Yang, "A Computationally Efficient Particle Filter for Multi-Target Tracking Using an Independence Approximation," *IEEE Tran. on Signal Proc.*, vol. 61, no. 4, pp. 843–856, 2013.

[131] H. Zou, "The Adaptive Lasso and its Oracle Properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[132] W. Zhang and G. Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," *IEEE Trans.on Wir. Com.*, vol. 3, no. 5, pp. 1689–1701, Sep. 2004.

[133] H. Zhu, I. D. Schizas, and G. B. Giannakis, "Power-Efficient Dimensionality Reduction for Distributed Channel-Aware Kalman Tracking Using Wireless Sensor Networks," *IEEE Tran. on Sig. Proc.,* vol. 57, no. 8, pp. 3193–3207, 2009.

[134] K. Zhou, and S. I. Roumeliotis. "Multirobot active target tracking with combinations of relative observations." IEEE Transactions on Robotics 27.4 (2011): 678-695.

[135] K. Zhang, L. Zhang, and M. Yang,"Real-time compressive tracking," *European Conference on Computer Vision,* pp. 864–877, Springer, 2012.

[136] Y. Zou, and K. Chakrabarty. "Distributed mobility management for target tracking in mobile sensor networks." Mobile Computing, IEEE Transactions on 6.8 (2007): 872-887.

[137] H. Zou, T. Hastie, and R. Tibshirani, "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, 2006.

[138] Y. K. Zhan and A. K. S. Jardine, "Adaptive Autoregressive Modeling of Non-Stationary Vibration Signals Under Distinct Gear States. Part I: Modeling," *Journal of Sound and Vibration,* vol. 286, pp. 429–450, 2005.

BIOGRAPHICAL STATEMENT

Guohua Ren was born in Lvliang, China, in 1994. He received his B.S. degree from University of Electronic Science and Technology of China, China, in 2012. He is currently a PhD student in the University of Texas at Arlington, majoring in Electrical Engineering. His research interests are on signal processing, sensor networks and machine learning.