Distributed and Robust Tracking by Exploiting Set-Membership and Sparsity

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Shahrokh Farahmand

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Professor Georgios B. Giannakis, Advisor

June 2011

"To my parents, Fereidoun and Shahin"

Acknowledgments

First and foremost, I wish to thank my Ph.D. advisor Prof. Georgios B. Giannakis. Without his guidance, supervision, and ever present support the completion of this thesis would have been impossible. Technically, he taught me many great things but the equally valuable things I learned from him were also non-technical. Among these, I learned how to be organized, professional, and on a well-crafted schedule, while simultaneously being ambitious and high-achieving in research and career goals. I am grateful to him for all of these and for believing in me even at times that I had given up.

I would like to thank my Ph.D. committee members Prof. Jarvis D. Haupt, Prof. Nihar Jindal, Prof. Mostafa Kaveh, and Prof. Stergios I. Roumeliotis for giving me the privilege of being my referees, taking the time and burden to review my Ph.D. thesis, and providing me with their insightful recommendations on this work.

I had the opportunity to collaborate with several individuals and I greatly benefited from their vision, ideas, and insights. Parts of this thesis was a result of these collaborations and I am indebted to Dr. Daniele Angelosante, Prof. Geert Leus, Prof. Stergios I. Roumeliotis, and Prof. Zhi Tian. For my other research works which do not appear in this thesis but nevertheless helped me expand my horizons I had the opportunity of collaborating with Dr. Alfonso Cano, Dr. Xiliang Luo, Prof. Zhi-Quan Luo, and Prof. Xin Wang.

During my stay at SPINCOM I had benefited from the friendship and enjoyed the light, fun, and relaxing gatherings and discussions with all my current and previous labmates. This was aside from our serious research-related discussions which also helped me a great deal. Due thanks are to Bahareh Akhbari, Dr. Daniele Angelosante, Prof. Massoud Babaie-Zadeh, Brian Baingana, Juan-Andres Bazerque, Dr. Alfonso Cano, Dr. Emiliano Dall'Anese, Pedro Andres Forero, Nikolaos Gatsis, Dr. Vassilis Kekatos, Dr. Seung-Jun Kim, Guobing Li, Dr. Qingwen Liu, Dr. Xiliang Luo, Morteza Mardani, Prof. Antonio G. Marques, Gonzalo Mateos, Eric Msechu, Ketan Rajawat, Prof. Alejandro Ribeiro, Yannis Schizas, Dr. Renqiu Wang, Dr. Tairan Wang, Prof. Xin Wang, Dr. Pengfei Xia, Nasim Yahya Soltani, Prof. Liuqing Yang, Dr. Yingqun Yu, Yu Zhang, Dr. Wanlun Zhao, and

Hao Zhu.

I also wish to give special thanks and gratitude to two of my closest friends during my B.Sc. studies in Iran. We shared many good and bad times but the most important thing was that we stayed together and kept in touch at all times even though we were in different corners of the world. As they say, good friendships are like wine, they get better as they age. To my great friends, Dr. Amir Hadi Djahanshahi and Reza Pasand Shanjani.

At last, I wanted to thank the most special people in my life. Their unconditional love and support was the main motivating factor that has taken me this far in life. Though not always in agreement, they stood firmly by my side at all times and gave me the will to carry on no matter how far was the goal or how difficult was the path. To my parents, Fereidoun and Shahin, my sister, Nahid, and my grandmother Katayoun.

Shahrokh Farahmand
Minneapolis, May 26 2011.

Abstract

Target tracking research and development are of major importance and continuously expanding interest to a gamut of traditional and emerging applications, which include radar- and sonar-based systems, surveillance and habitat monitoring using distributed wireless sensors, collision avoidance modules envisioned for modern transportation systems, and mobile robot localization and navigation in static and dynamically changing environments, to name a few. This thesis contributes in several issues pertaining to robustness and distributed operation of modern tracking systems.

The first issue addressed relates to measurement model nonlinearity. It turns out that by adopting a grid to describe the surveillance region, the nonlinear measurement model can be cast as a linear one at the cost of increasing state dimensionality. However, by exploiting sparsity of the state in this higher dimension, novel approaches are developed for tracking target signal strengths on a grid (TSSG). In multi-target settings, the proposed sparsity-aware TSSG trackers can bypass the challenge of data association and do not require knowing the number of targets present. To obtain individual target tracks when needed, simple data association techniques are also introduced. Due to the independence of TSSG trackers from the data association stage, association errors do not influence TSSG tracking performance.

Mitigating the effect of outliers appearing in the state and measurements is the second topic addressed in this thesis. The proposed *robust* algorithm referred to as doubly robust smoother (DRS) jointly estimates the outliers alongside with the state. To enable such joint estimation, sparsity in the outlier domain is exploited by regularizing the adopted criterion with the ℓ_1-norm of the outlier vector. Through novel methods for parameter tuning, DRS is capable of coping with even high levels of outlier contamination. To ensure low-complexity implementation, iterative coordinate descent and method-of-multipliers based solvers are derived for DRS. For settings where the state remains invariant or varies slowly with time, an online robust recursive least-squares (RLS) algorithm referred to as OR-RLS is also derived. Both DRS and OR-RLS are compared against existing robust alternatives and shown to significantly improve the performance.

Finally, a consensus-based *distributed* particle filter referred to as set-membership constrained particle filter (SMC-PF) is introduced for tracking with wireless sensor networks.

Consensus-based algorithms possess well-known merits such as robustness to sensor and link failures, scalability, and only local message exchanges with one-hop neighbors. In addition, SMC-PF provides a consensus-based mechanism for data adaptation based on set-membership ideas. SMC-PF outperforms state-of-the-art distributed particle filters in terms of performance and communication complexity.

Contents

List of Figures

Chapter 1

Introduction

Since their inception more than 50 years ago, dynamic state estimation and target tracking problems have been active areas of research up to this day. Target tracking research and development are of major importance and continuously expanding interest to a gamut of traditional and emerging applications, which include radar- and sonar-based systems, surveillance and habitat monitoring using distributed wireless sensors, collision avoidance modules envisioned for modern transportation systems, and mobile robot localization and navigation in static and dynamically changing environments, to name a few; see e.g., [7], [16], and references therein. The demanding nature of these applications have constantly advanced research and development efforts in this field.

This thesis touches upon several key research issues which are both challenging and of significance to the target tracking community. The first of these deals with measurement nonlinearity and data association for multi-target tracking. Typical measurements in tracking applications are bearing or distance from target(s) which are nonlinear functions of the Cartesian coordinates. If one tries to bypass these nonlinearities by switching to polar coordinates then convenient and popular state models such as constant-velocity

or discrete white noise acceleration (DWNA) model [7] become nonlinear functions of the polar coordinates. In a nutshell, nonlinearities are inherent to target tracking models, and should be dealt with in a proper manner. The approach of this thesis is to introduce and perform operations on a grid adopted to describe the space where tracking is performed. Introduction of the grid, allows for this problem to be cast in a linear framework at the price of increasing state dimensionality. The key enabling factor for successful operation of the proposed algorithms is the attribute of sparsity that is inherent to the grid-based framework and can be fruitfully utilized to improve tracking performance.

The second research issue pertains to outliers, which are defined as measurements, state entries, or noise terms not adhering to the nominal model adopted. They can arise due to many reasons that will be elaborated later on and challenge considerably tracking performance. One such reason is the model nonlinearity which gives rise to outliers stemming from linearization errors. The novel idea for handling outliers is to estimate them as nuisance parameters jointly with the system state. To this end, the proposed approach relies on sparsity-controlling regularization which renders the problem identifiable and well-posed. However, unlike the grid-based approach where sparsity is encountered in the state itself, the proposed robust approaches exploit sparsity in the outlier domain.

The third research issue deals with the development of distributed tracking algorithms. Recent emergence of wireless sensor networks (WSNs) has prompted a paradigm shift from centralized to distributed approaches, which must also exhibit low communication complexity, scalability, and robustness to sensor failures. While consensus-based schemes possess these attributes, they have been derived only for Kalman filters (KFs). But tracking problems entail nonlinear models comprising multi-modal, non-Gaussian random processes that prompts looking into distributed particle filters (PFs) in scenarios where their complexity remains manageable. Toward this objective, a consensus-based distributed PF is derived with improved performance compared to existing decentralized alternatives.

Elaborate discussion of each of these thrusts follows next along with a brief literature review per topic. Finally, contributions of this thesis in each case are pointed out.

1.1 Measurement nonlinearity and data association

At the core of long-standing research issues even for single-target tracking applications is the *nonlinear* dependence of the measurements on the desired state, which challenges the performance of linearized Kalman filter (KF) trackers, including the extended (E)KF [73, Chap. 13], the unscented (U)KF [71, 76], and their iterative variants [7, 16]. As an alternative, deterministic numerical methods can be employed to evaluate the integrals associated with minimum mean-square error (MMSE) state estimates per time step, and yield accurate results [68]. Unfortunately, their complexity increases exponentially with the problem dimension. This has motivated the development of particle filters (PFs), which can cope with nonlinearities but still tend to incur prohibitively high *complexity* in many critical applications. For multi-target tracking, *data association* has been another formidable challenge, especially when the ambient environment is cluttered, and the sensors deployed are unreliable. This challenge amounts to determining the target associated with each measurement, where the noisy measurements typically reflect the candidate target locations acquired through signal detection in gated validation regions; see e.g., [6, 16]. Once data association is established, targets can be tracked separately using the associated measurements.

Chapter 2 investigates the multi-target tracking problem whereby the available measurements comprise the superposition of received target signal strengths of all targets in the sensor field of view. Sensors collecting these measurements are not necessarily radars or high-cost receivers, but can be general-purpose radio units employing simple energy detectors. The measurements are nonlinearly related to target locations but no data association issues arise, because conventional range-gate operations have not yet been employed to de-

tect, separate, and localize the targets of interest [6]. To cope with the nonlinearity issue, Chapter 2 introduces a grid-based dynamical state-space model in which the state describes signal strengths of targets traversing a preselected spatial grid of the tracking field. Because the locations of grid points are preset and known, both the measurement and state equations become linear. Further, data association is avoided by dynamically tracking the TSSG values rather than directly producing the target tracks. Based on TSSG tracking however, data association and track trajectory estimation can be performed as a follow-up step, whereby track association and estimation errors do not propagate back to the TSSG tracker.

Similar ideas on bypassing data association at the price of tracking "less informative" estimates have been exploited in recent multi-target tracking schemes, such as the probability hypothesis density (PHD) filter [83, 115] and the Bayesian occupancy filter (BOF) [29]. The PHD filter tracks the so-termed target intensity, while the BOF tracks the probability of a grid point being occupied by any target. A main advantage of the *grid-based* TSSG tracker here is that state estimation becomes possible via KF applied to *linear* state and measurement models, at considerably reduced computational burden relative to the complexity incurred by the PHD and BOF. Further, the TSSG tracker is novel in exploiting the *sparsity* present in the grid-based state vector, which allows one to leverage efficient solvers of (weighted) least-squares (LS) minimization problems regularized by the ℓ_1-norm of the desired state estimate.

Sparsity-aware estimators have been studied for variable selection in *static* linear regression problems, and have recently gained popularity in signal processing and various other fields in the context of compressive sampling (CS); see e.g., [8, 23, 80]. However, few results pertain to the *dynamic* scenario encountered with target tracking. When measurements arrive sequentially in time, a sparsity-aware recursive least-squares scheme was reported in [3], but its tracking capability is confined only to slow model variations; see also [4] for

a sparsity-cognizant smoothing scheme which nevertheless does not lend itself to filtering; as well as [114], where a so-called KF-CS-residual scheme is reported for tracking slowly varying sparsity patterns. Different from existing alternatives, the present work develops sparsity-aware trackers along with their error covariances, without requiring knowledge on the number of (possibly fast-moving) targets or their signal strengths.

1.2 Robustness to state and measurement outliers

A major challenge in tracking applications is deviation from nominal conditions, which gives rise to outliers in the observations and state dynamics. Outliers in the state may come from abrupt changes in the target position due to, e.g., unexpected turbulence, and velocity variations due to target maneuvering. Outliers in the observations typically occur because of clutter, impulsive noise, and glint noise [78]. In addition, both types of outliers can arise after linearizing the emergent nonlinearities, as in the EKF. The clairvoyant KF and Kalman smoother (KS) can not handle state and/or measurement outliers [5, 84], because both can be viewed as minimizers of a weighted least-squares (WLS) criterion, which is known to be sensitive to outliers [66].

Robustification of KF and KS dates back to the '70s [84], but remains an active area of research until today [101,107], continuously leveraging advances in convex optimization [5,9]. Despite these advances, existing robust KF and KS approaches have several limitations. Some consider outliers only in the measurements [101], while others can handle either type of outliers alone but not both simultaneously [84]. Most approaches capitalize on robust e.g., M-estimators [107], which rely on Huber's and other outlier-resilient criteria [62, App. A6.8]. They require knowledge of the nominal distribution, and are effective only when the nominal noise is independent across observations and state entries [67, Chap. 7]. In the presence of correlated Gaussian noise, pre-whitening yields independent noise entries, which is required

for M-estimates to be applicable [107]. However, pre-whitening spreads the outliers to non-contaminated measurements. Approaches to doubly robust fixed-lag smoothing rely on heuristics to determine whether outliers are present in the state or the measurement equation [107].

A recent scheme for robust fixed-interval (but not fix-lag) smoothing is reported in [5], treating nonlinearities in the state and measurement equations separately from robustness issues. In the development, nonlinearities are linearized, and the measurement noise is assumed to follow the so-termed ℓ_1-Laplacian (or a Huber) distribution parameterized by a matrix R. The choice of R (and likewise that of Huber thresholds) critically affects smoothing performance, but systematic means of selecting these parameters was left open in [5]. Finally, a class of robust schemes popular in computer vision for linear regression settings comprises the so-termed random sample consensus (RANSAC)-based algorithms [49, 62].

If the outlier distributions are known and the model is linear and Gaussian (when conditioned on the outliers), efficient sequential Monte Carlo (SMC) smoothers based on Rao-Blackwellization [20] as well as deterministic algorithms based on pruning techniques, such as the interacting multiple model (IMM) method [31], will offer viable alternatives. Unfortunately, accurate description of the outlier distribution can be hard to obtain in practice. In addition, the complexity of SMC methods can be prohibitive for medium-to-large size problems due to the curse of dimensionality [32].

In Chapter 3, outliers are handled through auxiliary unknown variables that are *jointly* estimated along with the state. The resultant estimators rely on constraining the degree of outlier scarcity through ℓ_1-norm regularization, which is imposed on the auxiliary variables to enable *sparsity* control. The proposed robust smoothers: i) can handle both types of outliers simultaneously (hence referred to as doubly robust); ii) are universal, meaning they can operate even when the distributions of the nominal noise and outliers are unknown; iii) possess maximum a posteriori (MAP) optimality under specific assumptions on the outlier

and nominal noise distributions; iv) perform well under nominal conditions (i.e., with no outliers present); and v) outperform RANSAC- and Huber-based robust smoothers.

Unlike ordinary KS, the novel robust estimators are nonlinear functions of the data, and rely on the alternating direction method of multipliers (AD-MoM) or coordinate descent iterations. Closed-form expressions render the bulk of complexity per iteration comparable to that of KS, which is linear in the observation time. Few iterations of the coordinate descent or AD-MoM-based algorithms are required in practice to obtain satisfactory results. Numerical tests demonstrate that the developed methods can reject state and measurement outliers, and outperform RANSAC and Huber-based techniques.

1.2.1 Robustifying recursive least squares

In Chapter 4, outliers arise due to the faulty operation of low-cost unreliable sensors employed by a sensor network. One of the estimators whose performance can be greatly affected by outliers is least-squares (LS) and its recursive (R) version, namely the RLS. RLS is specifically significant due to two main reasons. Firstly, in many applications, measurements arrive at the estimator *sequentially*. If a batch estimator is to be utilized in such an application, its computational power and storage requirements grow with time rendering the estimator impractical. Iterative methods such as RLS are effective algorithms that operate with a fixed computational power and memory requirements. Secondly, through the use of forgetting factor, RLS can effectively track parameters that slowly vary with time.

Efforts to robustify the RLS have been reported in [10, 14, 24, 92, 97], and [121]. As discussed before, the main idea in some of these works has been to replace the non-robust LS error criterion with Huber's M-estimation criteria or other robustness enforcing costs and then solve the resulting optimization problem. Specifically, Huber's convex criterion was considered in [92], the cosine hyperbolic one appeared in [10], Hampel's three-part redescending M-estimate was advocated in [121], and a modified Huber cost was introduced

in [24]. Three major issues arise in all these approaches: i) some of these costs are not even convex, e.g., the modified Huber of [24]; hence, achieving the global optimum is not assured; ii) at every time step an optimization problem needs to be solved where the problem dimensionality grows linearly with time; and iii) correlated ambient noise can not be handled. The colored noise can arise in sensor networks applications, where spatially close sensors experience dependent background noises.

It should be noted that the curse of growing dimensionality is a critical limitation challenging the practicality of the proposed methods. To solve the proposed optimization problem that relies on a modified Huber M-estimate, [24] provides sub-optimal online recursions in the spirit of RLS. However, they do not solve the proposed optimization problem as RLS does for LS. The two most recent works in [14] and [97] provide remedies for the dimensionality growth by following a different path. Both limit the effect of each new datum on the estimate while ensuring low-complexity as RLS-like updates are utilized. However, they can not handle correlated noise, which can be a major limitation. While a simple pre-whitening will do when outliers are absent, any effort to pre-whiten the noise will spread the outliers to non-contaminated measurements rendering the estimation task much more difficult whenever outliers arise.

In Chapter 4, outliers are treated as extra nuisance variables that are jointly estimated with the variables of interest. Due to the extra outlier variables, identifiability issues arise. Building on the sparsity of outliers however, LS cost is regularized by the ℓ_1-norm of the outlier vector, which ensures identifiability. Consequently, one arrives at an optimization problem which handles both outliers and correlated background noise. Unfortunately, dimensionality of this optimization problem grows with time; hence, it can be only utilized at the initialization stage and is referred to as the offline benchmark. To cope with dimensionality, an online variant is developed, which enjoys low-complexity and closed-form updates in the spirit of RLS. Initialized with the offline benchmark and then proceeding

with real-time updates, the proposed online robust RLS (OR-RLS) is compared against RLS and the robust RLS schemes in [14] and [97] via simulations which demonstrate its efficiency and improved performance.

1.3 Distributed consensus-based particle filters

Trading-off accuracy for complexity, non-parametric, nonlinear estimators based on the PF, improve estimation accuracy compared to EKF and UKF while maintaining a reasonable computational burden under certain scenarios. In addition to complexity, when sensors of a WSN are deployed to perform decentralized tracking, coping with the inter-sensor communication overhead presents an extra challenge. Especially when a fusion center (FC) is not available, sensors have to share raw measurements via flooding, which renders the communication overhead prohibitively high. These considerations motivate the context and objective of Chapter 5, which is distributed PF with affordable complexity, and reduced overhead by communicating processed (as opposed to raw) measurements among neighboring sensors.

Tutorial treatment of PFs can be found in [22], [35], and [38]; see also [34, 89] for FC-based approaches. In-network (non-FC based) PF trackers are reported in [1,17,27,54,60,61, 64,77,82,87,98,104] and [105]. Specifically, local PFs are run at individual sensors in [1], and their estimates are spatially smoothed using a consensus filter. A query-response approach is advocated by [98] to exchange measurements among distributed robots. While easy to implement, the distributed schemes in [1] and [98] are ad-hoc, and can not perform close to a centralized PF tracker. Assuming a uniform prior, [17] adopts a mixture of local posteriors as the PF importance density, but does not account for past measurements in the current state estimate, which is tantamount to sub-optimality. Both [54] and [105] use a Gaussian mixture model (GMM) to approximate the posterior. They apply principal component analysis successively in an incremental loop to update GMM parameters. These two approaches as

well as those in [27, 64] and [104] belong to the class of incremental algorithms that rely on GMM or similar parametric models to communicate partial posteriors or likelihoods.

Specifically, [27, 64] and [104] approximate the centralized PF by sequential training of a parametric model. While [64] and [104] sequentially approximate the posterior of interest by a GMM, whose parameters are communicated incrementally from sensor to sensor, [27] sequentially trains a parametric model to match the likelihood. To further improve performance, [64] incorporates measurements from the current sensor along with measurements from the next sensor which is transmitted back to the current sensor in its importance sampling step to generate more efficient particles. All these approaches have the following limitations: i) they require establishment of a Hamiltonian cycle that goes through all sensors in the network, which is an NP-hard problem; ii) affordable sub-optimal paths used in lieu of a Hamiltonian cycle can miss some of the sensors and their data; iii) they incur excessive delays due to sequential GMM training; and iv) they lack robustness to sensor failures e.g., if the sensor who is currently processing fails, the whole estimation task is compromised.

Such limitations can be avoided if consensus-based approaches are utilized instead [1, 60, 61, 82, 87]. In [60], sensor data are used to train a GMM in a distributed fashion based on which particles are generated per sensor. In [61], sensors consent on the average mean and covariance of their local particle filters and use them to generate particles. Support vector machines were utilized in [82] to consent on the average of local functions yielding a reduced subset of particles. Being suboptimal, the performance of [1, 60, 61, 82] cannot approach the performance of centralized PF. Re-formulating the posterior as the geometric mean of locally known Gaussian terms, [87] invokes consensus to make the parameters of the resultant Gaussian posterior available across sensors. While this approach has potential to approach the performance of centralized PF, it cannot cope with multi-modal posteriors.

Another major challenge in distributed PF is whether to communicate a GMM approx-

imation of the posterior density or raw particles and weights. For low-dimensional data with a small number of modes, GMMs are reasonable. However, communicating GMMs amounts to transmitting covariance matrices which are proportional to the square of the state dimension. Therefore, GMMs can become costly for problems with large dimensions or a large number of peaks which can arise in multi-target tracking scenarios. Under such conditions, [77] proposes a Markov chain based approach, where particles and weights are communicated among sensors. The communication of such schemes can be greatly reduced if one can afford to communicate particle weights (that are scalars) only, and not the particles themselves. It will be demonstrated that this is possible at the cost of enforcing synchronism among sensors. A major contribution in this thrust is a synchronous non-parametric distributed PF operating either in an incremental- or a consensus-based mode. The proposed distributed PF can approach the performance of centralized PF, and requires communication of particle weights only.

Affordable inter-sensor communications are enabled through a novel distributed adaptation scheme, which considerably reduces the number of particles needed to achieve a given performance. Adaptation amounts to taking into account the current measurements in the importance density employed by the PF. The particles drawn according to a data-adapted importance density are more efficient, meaning they represent the posterior more accurately than particles drawn from a non-adapted density. Hence, one can afford fewer particles while alleviating the particle depletion problem.

Many works are available on adaptation methods for PFs [33, 36, 40, 50, 57, 59, 72, 86, 88, 94, 111]. However, none is developed to ensure affordable distributed implementation. In the centralized PF setting, existing approaches fall under two categories. The first includes parametric schemes which fit a Gaussian to the true posterior, and aim to find the associated mean and covariance. The latter can be obtained by an EKF iteration [36] or a UKF iteration [86], after equating terms in a Taylor series expansion of the un-normalized

posterior [88], or, via support vector regression [72]. Application-specific methods, such as the one in [18] for visual tracking, also fall under the first category. On the other hand, prior-editing [59], likelihood sampling [50] and its modifications [111], belong to the second category. Additional non-parametric approaches here include the particle flow using log-homotopy [33], where particles representing the prior are deterministically migrated to the region where the posterior has most of its probability mass; and its "stochastic" counterparts, which rely on a combination of bridging densities, and either Markov chain Monte Carlo methods [57], or, adaptive importance sampling [40]. The auxiliary PF benefits from adaptation as well [94]. A second contribution in this thrust of distributed tracking is a novel adaptation method that does not assume Gaussian posteriors. From this vantage point, it belongs to the second category of centralized adaptation methods, but is particularly attractive for distributed implementation.

1.4 Published results

This Ph.D. research on distributed and robust tracking has resulted in 4 journal papers to the Institute of Electrical and Electronic Engineers (IEEE) Transactions on Signal Processing [42, 44, 46, 48]. The work has also been disseminated at pertinent conferences where 4 articles have been presented [41, 43, 45, 47].

Chapter 2

Tracking target signal strengths on a grid using sparsity

Multi-target tracking is mainly challenged by the nonlinearity present in the measurement equation, and the difficulty in fast and accurate data association. To overcome these challenges, the present chapter introduces a grid-based model in which the state captures target signal strengths on a known spatial grid (TSSG). This model leads to *linear* state and measurement equations, which bypass data association and can afford state estimation via sparsity-aware Kalman filtering (KF). Leveraging the grid-induced sparsity of the novel model, two types of sparsity-cognizant TSSG-KF trackers are developed: one effects sparsity through ℓ_1-norm regularization, and the other invokes sparsity as an extra measurement. Iterative extended KF and Gauss-Newton algorithms are developed for reduced-complexity tracking, along with accurate error covariance updates for assessing performance of the resultant sparsity-aware state estimators. Based on TSSG state estimates, more informative target position and track estimates can be obtained in a follow-up step, ensuring that track association and position estimation errors do not propagate back into TSSG

state estimates. The novel TSSG trackers do not require knowing the number of targets or their signal strengths, and exhibit considerably lower complexity than the benchmark hidden Markov model filter, especially for a large number of targets. Numerical simulations demonstrate that sparsity-cognizant trackers enjoy improved root mean-square error performance at reduced complexity when compared to their sparsity-agnostic counterparts.

2.1 Grid-based state space model

Consider the problem of tracking M moving targets using N active (e.g., radar) or passive (e.g., acoustic) sensors deployed to provide situational awareness over a geographical area. Targets emit power either because they passively reflect the energy of other transmitters such as radar, or, because they are active sources such as cell-phones or transmitters mounted on smart cars. Associated with each target, say the mth one, is its position vector $\mathbf{p}_k^{(m)}$ per time k, and the signal of strength $s^{(m)}$ that the target reflects or emits. Sensor n measures the superposition of received target signal strengths,

$$y_{n,k} = \sum_{m=1}^{M} h\left(d_k^{m\to n}\right) s^{(m)} + \nu_{n,k}\,, \quad n = 1, \ldots, N\,, \quad k = 1, 2, \ldots \tag{2.1}$$

where $h(\cdot)$ denotes the distance-dependent propagation function; $d_k^{m\to n} := \|\mathbf{p}_k^{(m)} - \mathbf{q}_n\|_2$ is the distance between the known position $\mathbf{q}_n$ of sensor n and the unknown position vector $\mathbf{p}_k^{(m)}$ of target m; and $\nu_{n,k}$ is zero-mean Gaussian noise at sensor n. Function $h(\cdot)$ satisfies $h(0) = 1$, is non-negative, decreasing, and is either assumed known from the physics of propagation or acquired through training [80].

At each time k, a centralized processor has available the measurement vector $\mathbf{y}_k := [y_{1,k}, \ldots, y_{N,k}]^T$, based on which the target positions $\{\mathbf{p}_k^{(m)}\}_{m=1}^M$ are to be tracked. Note that the measurement model (2.1) differs from the one typically considered in radar applications, where a measurement either comes from a single target or a clutter, usually in the form of

position information obtained from range gate operations [6]. Each measurement in (2.1) comes from a sensor, and comprises the superposition of received signal strengths emitted by or reflected from all targets in the sensor field of view. This model considers the localization and tracking problems jointly, and avoids the measurement-target association issue.

One major challenge in tracking and localization problems is that the measurements in (2.1) are nonlinear functions of the wanted target position vectors. A neat approach to arrive at a linear measurement model is to adopt a set of G (possibly regularly spaced) grid points at known positions $\{\mathbf{g}_i\}_{i=1}^{G}$, where target(s) could be potentially located; see also e.g., [29], [23], and [8]. Using a sufficiently dense grid, it is possible to capture the target locations at a prescribed spatial resolution using a $G \times 1$ vector $\mathbf{x}_k$ having most entries equal to zero except for the $\{i_k^{(m)}\}_{m=1}^{M}$ entries given by $\{x_k^{(i_k^{(m)})}\}_{m=1}^{M}$, which represent the target signal strengths at time k if and only if the m-th target is located at the $i_k^{(m)}$-th grid point, that is $\mathbf{p}_k^{(m)} = \mathbf{g}_{i_k^{(m)}}$. Note that if target m is located exactly on a grid point $i_k^{(m)}$, then $x_k^{(i_k^{(m)})} \equiv s^{(m)} \neq 0$ will be the only nonzero entry of $\mathbf{x}_k$ corresponding to this target. However, to account for target presence off the preselected grid points, it will be allowed for the unknown target signal strength $s^{(m)}$ to "spill over" grid points around $i_k^{(m)}$ and thus render nonzero a few neighboring entries of $\mathbf{x}_k$. Let $\mathcal{G}_k^{(m)}$ denote the spill-over region on the grid corresponding to target m at time k, such that $x_k^{(i)} \neq 0$ is associated with $s^{(m)}$, $\forall i \in \mathcal{G}_k^{(m)}$. The following assumption on this target occupancy model is imposed:

(as1) *Each grid point i can be occupied by at most one target m at any given time k.*

This assumption can be easily satisfied in practice by selecting a sufficiently dense grid [29, 39]. Under as1), each grid point i is associated with a unique target index $m_k^{(i)}$ at time k; that is, $i \in \mathcal{G}_k^{(m_k^{(i)})}$, where $m_k^{(i)} \in [1, M]$ if it is occupied by one of the M targets; or, $m_k^{(i)} = 0$ if it is not occupied, meaning it is associated with a dummy target $m = 0$ with strength $s^{(0)} \equiv 0$. Apparently, $\{\mathcal{G}_k^{(m)}\}_{m=0}^{M}$ are mutually exclusive across m and their

union spans the entire grid in the sense $\cup_{m=0}^{M}\mathcal{G}_k^{(m)} = \cup_{i=1}^{G} i$, which leads to a measurement equation [cf. (2.1)]

$$y_{n,k} = \sum_{m=0}^{M} \sum_{i\in\mathcal{G}_k^{(m)}} h\left(d^{(i\to n)}\right) x_k^{(i)} + v_{n,k} = \mathbf{h}_n^T \mathbf{x}_k + v_{n,k}. \tag{2.2}$$

Here $\mathbf{h}_n^T := [h(d^{1\to n}), h(d^{2\to n}), \ldots, h(d^{G\to n})]$; $d^{i\to n} := \|\mathbf{q}_n - \mathbf{g}_i\|_2$ now denotes the known time-invariant distance between the nth sensor and the ith grid point; and the noise $v_{n,k}$ replacing $\nu_{n,k}$ in (2.1) captures the unmodeled dynamics in the aforementioned spill-over effect. Notwithstanding, thanks to the grid-based model, the measurements in (2.2) have become linear functions of the unknown $\mathbf{x}_k$ whose nonzero entries reveal the grid points where target signal strengths are present at time k.

The next step is to model the evolution of $\mathbf{x}_k$ in time as the targets move across the grid. Regarding their movement pattern, targets obey the following assumption:

(as2) *All targets move according to identical transition probabilities* $\{f_k^{(ji)}\}_{i,j=1}^{G}$*, where* $f_k^{(ji)} := p(x_k^{(j)} \neq 0 | x_{k-1}^{(i)} \neq 0; j \in \mathcal{G}_k^{(m)}, i \in \mathcal{G}_{k-1}^{(m)})$*,* $m = 1, \ldots, M$*.*

In words, the homogeneity of targets under as2) refers to the probability that a target m moves from grid point i at time $k-1$ to point j at time k.

Consider now expressing each entry of $\mathbf{x}_k$ as $x_k^{(j)} = s_k^{(j)} \cdot p(x_k^{(j)} \neq 0)$, where $s_k^{(j)} = s^{(m_k^{(j)})} \in \{s^{(0)}, s^{(1)}, \ldots, s^{(M)}\}$ denotes a nonnegative proportionality constant, and $p(x_k^{(j)} \neq 0)$ stands for the probability of a target to be present on grid point j at time k. Essentially, each $x_k^{(j)}$ is associated with only one of the $(M+1)$ targets (including the dummy target $m = 0$) indexed by $m_k^{(j)}$, and $s_k^{(j)}$ is a proportionality constant in the sense that it takes on $(M+1)$ possible values $s^{(m)} = \sum_{j\in\mathcal{G}_k^{(m)}} x_k^{(j)}$, for $m = 0, 1, \ldots M$.

Under as1) and as2), it is shown in the Appendix that the state obeys the following

recursion

$$x_k^{(j)} = \sum_{i=1}^{G} f_k^{(ji)} x_{k-1}^{(i)}, \quad \forall j \in [1, G]. \tag{2.3}$$

Concatenating (2.3) for $j = 1, \ldots, G$, and (2.2) for $n = 1, \ldots, N$, one arrives at the *grid-based* model

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k \tag{2.4a}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{2.4b}$$

where the $G \times G$ state transition matrix $\mathbf{F}_k$ has its (i, j)-th entry given by $f_k^{(ji)}$; the measurement matrix is defined as $\mathbf{H} := [\mathbf{h}_1, \ldots, \mathbf{h}_n]^T$; likewise for the measurement noise vector $\mathbf{v}_k := [v_{1,k}, \ldots, v_{N,k}]$; and $\mathbf{w}_k$ is a zero-mean process noise vector with a positive-definite covariance matrix $\mathbf{Q}_k$ added to account for both as1) and the natural non-negativity constraints on $\mathbf{x}_k$ whose entries represent target signal strengths (magnitudes or power).

A distinct feature of model (2.4) is that the unknown $\mathbf{x}_k$ is *sparse* $\forall k$, since only few of its G entries are nonzero (in fact exactly M nonzero entries if all the M targets are located on grid points). Although (2.3) describes the linear evolution of each $\mathbf{x}_k$ entry under as1), using these recursions alone does not guarantee that the predicted or estimated $\mathbf{x}_k$ adheres to as1). Indeed, starting with a target at an arbitrary entry in $\mathbf{x}_0 \neq \mathbf{0}$ and running (2.3) up to a large enough k, the signal strength of this target will "spill-over" to all entries of $\mathbf{x}_k$, and will possibly overlap with other targets present. Such a state transition pattern is expected, because uncertainty of any dynamically evolving state grows over time if no corrections are made based on real-time measurements. Therefore, $\mathbf{x}_k$ predictions based on (2.4a) will be non-sparse, but the true state vector $\mathbf{x}_k$ at any time k is sparse with only a few nonzero entries around the target locations. Posterior to processing the measurements, filtered and predicted renditions of $\mathbf{x}_k$ will remain sparse as well. The noise term $\mathbf{w}_k$ reflects the uncertainty in the state transition model under as1).

This sparsity attribute will prove to be instrumental for enhancing tracking performance. Also, it is worth noting that the state transition matrix $\mathbf{F}_k$ reflects the transition behavior of target positions only, without revealing full information of the target movement model that may be dependent on velocity or other factors as well. In fact, $\mathbf{F}_k$ is derived from the target movement model but does not fully reveal it, which differs from most existing track state models.

Given $\mathbf{y}_{1:k} := \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$, the goal of this paper is to track $\mathbf{x}_k$ using a *sparsity-aware* Kalman filter (KF). Since $\mathbf{x}_k$ represents the target signal strength on the grid (TSSG), the KF-like algorithms proposed in Sections 2.2 and 2.3 will be referred to as TSSG–KF trackers, while the iterated extended Kalman filter (IEKF) algorithms of Section 2.4 will be referred to as TSSG–IEKF trackers. Having available $\hat{x}_k^{(j)}$ estimates, and recalling that $x_k^{(j)} = s^{(m_k^{(j)})} p(s_k^{(j)} \neq 0)$, one can estimate the constant $s^{(m)}$ capturing the signal strength of the m-th target at time k as

$$\hat{s}_k^{(m)} = \sum_{j \in \mathcal{G}_k^{(m)}} \hat{x}_k^{(j)}, \quad \forall\ k \tag{2.5}$$

and the corresponding target position vector at time k as

$$\hat{\mathbf{p}}_k^{(m)} = (1/\hat{s}_k^{(m)}) \sum_{j \in \mathcal{G}_k^{(m)}} \mathbf{g}_j \hat{x}_k^{(j)}, \qquad m = 1, \ldots, M. \tag{2.6}$$

The following remark makes useful observations regarding the position estimate in (2.6).

Remark 1. A TSSG filter for tracking $\mathbf{x}_k$ avoids data association, because the TSSG-based state and measurement equations in (2.4) hold for any target-grid association $\{\mathcal{G}_k^{(m)}\}_m$, so long as as1) and as2) are satisfied. On the other hand, finding the target positions via (2.6) requires knowledge of $\{\mathcal{G}_k^{(m)}\}_m$, and hence calls for associating targets with TSSG entries. Solution to such an association problem will be provided in Section 2.5. Nonetheless, it is worth stressing that the association errors and resultant position estimation errors do not affect TSSG tracking that is independent of target position estimation, similar to the PHD and BOF in [83] and [29], respectively.

In addition to reduced complexity, an attractive feature of the present formulation relative to e.g., [29] is that even for finite G, there is no need to assume that targets are located on grid points since (2.6) allows for interpolating the target position vectors regardless, after knowing that grid point j is associated with the target $m_k^{(j)}$ occupying it. The next remark is useful to further appreciate this point.

Remark 2. Given measurements $\mathbf{y}_{1:k}$, and supposing that the number of targets M and their signal strengths $\{s^{(1)}, \ldots, s^{(M)}\}$ are known, the maximum a posteriori (MAP) and minimum mean-square error (MMSE) optimal trackers can be derived from a hidden Markov model (HMM) filter implementing the following recursions derived from Bayes' rule (cf. (2.34) and (2.35) in the Appendix)

$$\begin{aligned} p\Big(x_k^{(j)} \neq 0 \Big| \mathbf{y}_{1:k-1}\Big) &= \sum\nolimits_{i \in \mathcal{G}_{k-1}^{(m_k^{(j)})}} f_k^{(ji)}\; p\Big(x_{k-1}^{(i)} \neq 0 \Big| \mathbf{y}_{1:k-1}\Big) \\ p\left(x_k^{(j)} \neq 0 \Big| \mathbf{y}_{1:k}\right) &= \frac{p(\mathbf{y}_k | x_k^{(j)} \neq 0; s^{(m_k^{(j)})}) p(x_k^{(j)} \neq 0 | \mathbf{y}_{1:k-1})}{\sum_{i \in \mathcal{G}_k^{(m_k^{(j)})}} p(\mathbf{y}_k | x_k^{(i)} \neq 0; s^{(m_k^{(i)})}) p(x_k^{(i)} \neq 0 | \mathbf{y}_{1:k-1})} \end{aligned} \tag{2.7}$$

where $f_k^{(ji)}$ is the transition probability as in (2.3). These HMM recursions hinge on prior knowledge of the target-grid association $\{\mathcal{G}_k^{(m)}\}_{m=0}^M$, which need to be figured out among a total of $(M+1)^{G-M} G!/(G-M)!$ possible combinations. A large G increases grid density and hence spatial resolution, at the expense of increasing complexity. In addition, M and $\{s^{(m)}\}_{m=1}^M$ need to be known beforehand.

One additional remark is now in order.

Remark 3. Although $\mathbf{y}_k$ in (2.4b) comprises scalar measurements from N geographically distributed sensors per time k, it is possible to form $\mathbf{y}_k$ with samples of the continuous-time signal received at a *single* (e.g., a radar or sonar) sensor by over-sampling at a rate faster than the rate $\mathbf{x}_k$ changes, so long as the state-space model (2.4) is guaranteed to be observable (and thus $\mathbf{x}_k$ is ensured to be identifiable).

2.2 KF for tracking TSSG

If the non-negativity constraints for $\mathbf{x}_k$ were absent, the optimal state estimator for (2.4) in the MAP, MMSE, or least-squares (LS) error sense would be the clairvoyant linear KF. A pertinent state estimator is pursued here in the presence of non-negativity constraints. Suppose that the estimate $\hat{\mathbf{x}}_{k-1|k-1}$ and its error covariance matrix $\mathbf{P}_{k-1|k-1}$ are available from the previous time step. At time k, the KF state predictor and its error covariance are obtained as

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k. \end{aligned} \tag{2.8}$$

For the KF corrector update, consider the LS formulation of the KF; see e.g., [107]. The corrector update can be derived as a regularized LS criterion, which will also be useful to account for the sparsity attribute. To show this, view $\hat{\mathbf{x}}_{k|k-1}$ as a noisy measurement of $\mathbf{x}_k$. It follows readily from (2.8) that $\hat{\mathbf{x}}_{k|k-1} = \mathbf{x}_k + \mathbf{e}_{k|k-1}$, where $\mathbf{e}_{k|k-1}$ has covariance matrix $\mathbf{P}_{k|k-1}$. Stacking $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{y}_k$ to form an augmented measurement vector, yields the following linear regression model

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{I}_G \\ \mathbf{H} \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \mathbf{e}_{k|k-1} \\ \mathbf{v}_k \end{bmatrix}$$

where the augmented noise vector has block diagonal covariance matrix denoted as $\text{diag}(\mathbf{P}_{k|k-1}, \mathbf{R}_k)$. The weighted (W)LS estimator for this linear regression problem is given by

$$\hat{\mathbf{x}}_{k|k} = \arg \min_{\mathbf{x}_k \geq \mathbf{0}} \|\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_k\|^2_{\mathbf{P}^{-1}_{k|k-1}} + \|\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\|^2_{\mathbf{R}_k^{-1}} \tag{2.9}$$

where $\|\mathbf{x}\|^2_{\mathbf{A}} := \mathbf{x}^T \mathbf{A} \mathbf{x}$. In the absence of non-negativity constraints, the optimal state corrector $\hat{\mathbf{x}}_{k|k}$ can be found in closed form as the cost is quadratic, and likewise its error

covariance can be updated as

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R}_k)^{-1}\mathbf{H}\mathbf{P}_{k|k-1}. \tag{2.10}$$

A gradient projection algorithm will be developed in Section 2.3 to solve (2.9) under non-negativity constraints on the state vector. However, (2.10) will still be used bearing in mind that this update is approximate now. The TSSG–KF tracker implemented by (2.8)-(2.10) is sparsity-agnostic, as it does not explicitly utilize the prior knowledge that $\mathbf{x}_k$ is sparse.

2.3 Sparsity-aware KF trackers

Taking into account sparsity, this section develops sparsity-cognizant trackers. To this end, the degree of sparsity quantified by the number of nonzero entries of $\mathbf{x}_k$, namely the ℓ_0-norm $\|\mathbf{x}_k\|_0$, can be used to regularize the LS cost of the previous section. Unfortunately, similar to compressed sensing formulations for solving under-determined linear systems of equations [21], such a regularization results in a non-convex optimization problem that is NP-hard to solve, and motivates relaxing the ℓ_0-norm with its closest convex approximation, namely the ℓ_1-norm. Thus, the proposed sparsity-cognizant tracker is based on the state corrector minimizing the following ℓ_1-regularized WLS cost function

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= \arg\min_{\mathbf{x}_k \geq \mathbf{0}} J(\mathbf{x}_k) \\ J(\mathbf{x}_k) &:= \|\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_k\|^2_{\mathbf{P}^{-1}_{k|k-1}} + \|\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\|^2_{\mathbf{R}_k^{-1}} + 2\lambda_k\|\mathbf{x}_k\|_1. \end{aligned} \tag{2.11}$$

The state corrector minimizing (2.11), together with the covariance update[1] in (2.10) and the prediction step in (2.8), form the recursions of the sparsity-aware TSSG–KF tracker. Relevant design choices and algorithms for minimizing (2.11) will be elaborated in the next subsection.

[1] A more accurate covariance update will be derived in (2.28).

The TSSG-KF trackers in (2.9) and (2.11) involve both prediction and correction steps, which interestingly can be combined into a single estimation step. Considering that both $\mathbf{x}_{k-1}$ and $\mathbf{x}_k$ are sparse and non-negative, and combining the LS terms for both the prediction and correction steps, the following optimization problem arises for some non-negative λ_{k-1} and λ_k parameters:

$$\hat{\mathbf{x}}_{k|k} = \arg \min_{\mathbf{x}_{k-1},\mathbf{x}_k \geq \mathbf{0}} \Big\{ \|\hat{\mathbf{x}}_{k-1|k-1} - \mathbf{x}_{k-1}\|^2_{\mathbf{P}^{-1}_{k-1|k-1}} + \|\mathbf{x}_k - \mathbf{F}_k \mathbf{x}_{k-1}\|^2_{\mathbf{Q}_k^{-1}} + \|\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\|^2_{\mathbf{R}_k^{-1}} + \lambda_{k-1}\|\mathbf{x}_{k-1}\|_1 + \lambda_k \|\mathbf{x}_k\|_1 \Big\} . \tag{2.12}$$

The performance gain of this tracker was evaluated via simulations and no substantial improvement over the TSSG-KF tracker was observed. For this reason, focus henceforth will be placed on the TSSG-KF tracker in (2.11).

2.3.1 Parameter selection

The scalar parameter λ_k in (2.11) controls the sparsity-bias tradeoff [63]. The corrector $\hat{\mathbf{x}}_{k|k}$ becomes increasingly sparse as λ_k increases, and eventually vanishes, i.e., $\hat{\mathbf{x}}_{k|k} = \mathbf{0}$, when λ_k exceeds an upper bound $\bar{\lambda}_k$. There are two systematic means of selecting λ_k. The first one popular for variable selection in linear regressions is cross-validation [63, pp. 241-249]. The second one is the so-termed absolute variance deviation based selection that has been advocated in the context of outlier rejection setups [41]. Both approaches require solving (2.11) for different trial values of λ_k. Even though warm starts reduce the computational burden considerably, this can be certainly affordable for offline solvers of a linear regression problem or a fixed-interval smoothing scenario, but may incur prohibitive delays for real-time applications. For the tracking problem at hand, the simple rule advocated is to set $\lambda_k = \alpha\bar{\lambda}_k$, where $\alpha \in (0, 1)$ is a fixed scaling value to avoid the trivial solution $\hat{\mathbf{x}}_{k|k} = \mathbf{0}$. The bound $\bar{\lambda}_k$ is derived next.

Proposition 1. *The solution to (2.11) reduces to* $\hat{\mathbf{x}}_{k|k}=\mathbf{0}$ *for any scalar* $\lambda_k \geq \bar{\lambda}_k$*, where*

$$\bar{\lambda}_k = \|\mathbf{P}_{k|k-1}^{-1}\hat{\mathbf{x}}_{k|k-1} + \mathbf{H}^T\mathbf{R}_k^{-1}\mathbf{y}_k\|_\infty. \tag{2.13}$$

Proof: Since $\mathbf{x}_k \geq \mathbf{0}$, it holds that $\|\mathbf{x}_k\|_1 = \mathbf{x}_k^T\mathbf{1}$, where $\mathbf{1}$ denotes the all-one vector. Therefore, $J(\mathbf{x})$ in (2.11) is differentiable and results in a convex problem. The necessary and sufficient optimality condition states that $\mathbf{x}^*$ is an optimum point iff $(\mathbf{y}-\mathbf{x}^*)^T\nabla J(\mathbf{x}^*) \geq 0,\ \forall \mathbf{y} \geq \mathbf{0}$. For $\mathbf{x}^*=\mathbf{0}$, this condition holds iff $\nabla J(\mathbf{x}^*) \geq \mathbf{0}$. It then follows from (2.11) that

$$\nabla J(\mathbf{x}) = 2\Big(-\mathbf{P}_{k|k-1}^{-1}(\hat{\mathbf{x}}_{k|k-1}-\mathbf{x}) - \mathbf{H}^T\mathbf{R}_k^{-1}(\mathbf{y}_k-\mathbf{H}\mathbf{x}) + \lambda_k\mathbf{1}\Big). \tag{2.14}$$

Therefore, $\mathbf{x}^*=\mathbf{0}$ is an optimal solution iff (2.13) holds. □

2.3.2 Gradient projection algorithms

As (2.11) is a convex problem, convex optimization software such as SeDuMi [103] can be utilized to solve it efficiently. In addition to these solvers, low-complexity iterative methods are developed here, by adopting the gradient projection (GP) algorithms in [13, pp. 212-217]. Note that the proposed algorithms can be used to obtain the sparsity-agnostic tracker from (2.9) too, since the latter is obtained by minimizing a special case of (2.11) corresponding to $\lambda_k = 0$.

At each time k, the GP is initialized with $\hat{\mathbf{x}}_{k|k}(0) = \hat{\mathbf{x}}_{k|k-1}$ at iteration $l=0$. The state corrector iterates from l to $(l+1)$ as follows

$$\hat{\mathbf{x}}_{k|k}(l+1) = \left[\hat{\mathbf{x}}_{k|k}(l) - \gamma\nabla J\left(\hat{\mathbf{x}}_{k|k}(l)\right)\right]^+ \tag{2.15}$$

where $[x]^+$ denotes the projection onto the non-negative orthant, γ is the step size, and ∇J is as in (2.14). Here $J(\mathbf{x}_k)$ is differentiable because $\|\mathbf{x}_k\|_1 = \mathbf{x}_k^T\mathbf{1}$ when $\mathbf{x}_k \geq \mathbf{0}$.

While (2.15) amounts to a Jacobi-type iteration updating all the entries at once, one can also devise Gauss-Seidel variants, where entries are updated one at a time [13, pp.

218-219]. This is possible because the non-negative orthant is a constraint set expressible as the Cartesian product of one-dimensional sets, allowing entry-wise updates per iteration $(l+1)$ as

$$\hat{x}_{k|k}^{(j)}(l+1) = \max\left\{0,\ \hat{x}_{k|k}^{(j)}(l) - \gamma \nabla_j J\left(\tilde{\mathbf{x}}_{k|k}^{(j)}(l)\right)\right\} \tag{2.16}$$

where $\tilde{\mathbf{x}}_{k|k}^{(j)}(l) := \left\{\hat{\mathbf{x}}_{k|k}^{(1:j-1)}(l+1), \hat{\mathbf{x}}_{k|k}^{(j:G)}(l)\right\}$ has its first $(j-1)$ entries already updated in the $(l+1)$st iteration. Convergence of the iterations in (2.16) to the optimum solution of (2.11) is guaranteed under mild conditions by the results in [13, p. 219]. Specifically, $J(\mathbf{x}_k)$ should be non-negative and its gradient should be Lipschitz continuous, both of which hold for the objective in (2.11).

Proposition 2. *Any limit point of the sequence generated by (2.16), with arbitrary initialization* $\hat{\mathbf{x}}_{k|k}^{(0)}$*, is an optimal solution of (2.11) provided that the step size* γ *is chosen small enough.*

In practice, only a few gradient-projection iterations are run per time step k to allow for real-time sparsity-aware KF tracking.

2.4 Enhanced sparsity-aware IEKF tracking

The proposed sparsity-aware tracker employs the KF covariance recursion in (2.10) to update the error covariance of the corrector state estimate. As it does not account for the ℓ_1-norm regularization, this update is approximate. In order to incorporate the prior knowledge of sparsity when updating the corrector covariance, this section develops an EKF-based approach, which leads to enhanced tracking performance.

Toward this objective, the prior information on sparsity is viewed as an extra measurement $\mu_k = \|\mathbf{x}_k\|_0$, rather than as a regularizing term in the LS cost function. When the number of targets M is known, an apparent choice is to set $\mu_k = M$. Accordingly, tracking

will be carried out based on an augmented $(N+1)\times 1$ measurement vector, given by

$$\bar{\mathbf{y}}_k := [\mathbf{y}_k^T \ \mu_k]^T.$$

2.4.1 Viewing sparsity as an extra measurement

The added measurement can be modeled in a general form as

$$\mu_k = \rho(\mathbf{x}_k) + u_k$$

where $\rho(\mathbf{x}_k)$ is a differentiable function approximating the sparsity-inducing ℓ_0-norm, and u_k denotes zero-mean noise with variance σ_k^2. The noise term captures both the uncertainty in approximating $\|\mathbf{x}_k\|_0$, as well as the error in attaining the desired degree of sparsity. As to $\rho(\mathbf{x}_k)$, three well-known approximants of the ℓ_0-norm are the ℓ_1-norm, the logarithm, and the inverse Gaussian functions:

$$\begin{aligned}
(\ell_1\text{-norm}) \qquad & \rho(\mathbf{x}_k) = \mathbf{x}_k^T\mathbf{1} \\
(\text{logarithm}) \qquad & \rho(\mathbf{x}_k) = \textstyle\sum_{j=1}^{G} \log\left(x_k^{(j)} + \delta\right) \\
(\text{inverse Gaussian}) \qquad & \rho(\mathbf{x}_k) = \textstyle\sum_{j=1}^{G} \left(1 - \exp\left(-\frac{(x_k^{(j)})^2}{2\sigma_p^2}\right)\right)
\end{aligned}$$

where δ and σ_p are tuning parameters, and only $\mathbf{x}_k \geq \mathbf{0}$ is considered. These nonlinear functions are plotted along with the ℓ_0-norm function for a scalar $\mathbf{x}_k$ in Fig. 2.1. It can be seen that they all have relatively sharp edges around the origin to approximate the ℓ_0-norm.

Adding the extra measurement μ_k, the state space model in (2.4) is augmented to

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} \tag{2.17a}$$

$$\bar{\mathbf{y}}_k = \bar{\mathbf{h}}(\mathbf{x}_k) + \bar{\mathbf{v}}_k \tag{2.17b}$$

where $\bar{\mathbf{h}}(\mathbf{x}_k) := [(\mathbf{H}\mathbf{x}_k)^T, \ \rho_k(\mathbf{x}_k)]^T$ consists of $N+1$ scalar measurement functions that can be nonlinear in general, and $\bar{\mathbf{v}}_k := [\mathbf{v}_k^T, \ u_k]^T$ has covariance $\bar{\mathbf{R}}_k := \text{diag}(\mathbf{R}_k, \sigma_k^2)$. Similar to

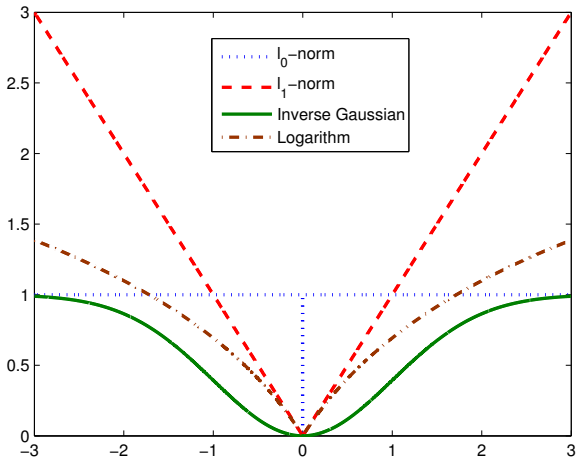

Figure 2.1: The ℓ_0-norm and its three approximations.

(2.11), the model in (2.17) leads to a nonlinear (N)LS problem

$$\hat{\mathbf{x}}_{k|k} = \arg \min_{\mathbf{x}_k \geq \mathbf{0}} J_1(\mathbf{x}_k) \tag{2.18}$$
$$J_1(\mathbf{x}_k) := \|\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_k\|^2_{\mathbf{P}^{-1}_{k|k-1}} + \|\mathbf{y}_k - \mathbf{H}\mathbf{x}_k\|^2_{\mathbf{R}^{-1}_k} + \sigma_k^{-2}\left(\mu_k - \rho(\mathbf{x}_k)\right)^2 .$$

Compared with (2.11), (2.18) replaces the ℓ_1-norm of $\mathbf{x}_k$ with an alternative LS-error regularization involving the extra measurement which accounts for the sparsity present. Because (2.18) directly results from (2.17), the error covariance of state estimates can be updated using the KF-like recursions developed next.

2.4.2 IEKF algorithm for nonlinear measurement models

Since the augmented $\bar{\mathbf{y}}_k$ in (2.17b) is a nonlinear function of the wanted TSSG state, the EKF approach is adopted here to update the error covariance along the lines of e.g., [7, Chap. 10]. Specifically, an iterated (I)EKF algorithm is employed, which is tantamount to applying Gauss-Newton iterations to a relevant NLS regression problem [11].

The prediction step of the IEKF is similar to KF, hence $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\mathbf{P}}_{k|k-1}$ follow directly from the state space model in (2.17), and coincide with (2.8). For the correction step per time k, IEKF recursions are initialized with $\hat{\mathbf{x}}_{k|k}(0) = \hat{\mathbf{x}}_{k|k-1}$ for $l = 0$, and subsequent iterations proceed as follows [109, Appendix C]

$$\begin{aligned} \hat{\mathbf{x}}_{k|k}(l+1) &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(l)\Big(\bar{\mathbf{y}}_k - \bar{\mathbf{h}}(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{\Phi}(l)(\hat{\mathbf{x}}_{k|k}(l) - \hat{\mathbf{x}}_{k|k-1})\Big) \\ \mathbf{K}(l) &= \mathbf{P}_{k|k-1}\mathbf{\Phi}^T(l)\big(\mathbf{\Phi}(l)\mathbf{P}_{k|k-1}\mathbf{\Phi}^T(l) + \bar{\mathbf{R}}_k\big)^{-1} \end{aligned} \tag{2.19}$$

where $\mathbf{\Phi}(l) := \nabla\bar{\mathbf{h}}\big(\hat{\mathbf{x}}_{k|k}(l)\big)^T$ denotes the Jacobian matrix of $\bar{\mathbf{h}}(\cdot)$ evaluated at $\hat{\mathbf{x}}_{k|k}(l)$. After the IEKF iterations are completed at $l = L$, the corrector's error covariance matrix is updated as

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}(L)\mathbf{\Phi}(L)\mathbf{P}_{k|k-1}. \tag{2.20}$$

The ensuing proposition establishes the link between IEKF and Gauss-Newton iterations for the related NLS problem.

Proposition 3. *Consider the NLS problem [cf. (2.17) and (2.18)]*

$$\hat{\mathbf{x}}_{k|k} = \arg\ \min_{\mathbf{x}_k} \|\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_k\|^2_{\mathbf{P}^{-1}_{k|k-1}} + \|\bar{\mathbf{y}}_k - \bar{\mathbf{h}}(\mathbf{x}_k)\|^2_{\bar{\mathbf{R}}^{-1}_k}. \tag{2.21}$$

Solving (2.21) via Gauss-Newton iterations initialized with $\hat{\mathbf{x}}_{k|k}(0) = \hat{\mathbf{x}}_{k|k-1}$, amounts to the IEKF recursions in (2.19).

Proof: The quadratic terms in (2.21) can be rewritten as

$$\hat{\mathbf{x}}_{k|k} = \arg\ \min_{\mathbf{x}_k} \|\mathbf{g}(\mathbf{x}_k)\|_2^2 \tag{2.22}$$

$$\text{where} \quad \mathbf{g}(\mathbf{x}_k) = \begin{bmatrix} \mathbf{P}^{-1/2}_{k|k-1}(\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_k) \\ \\ \bar{\mathbf{R}}_k^{-1/2}(\bar{\mathbf{y}}_k - \bar{\mathbf{h}}(\mathbf{x}_k)) \end{bmatrix}. \tag{2.23}$$

Gauss-Newton iterations for (2.22) become

$$\hat{\mathbf{x}}_{k|k}(l+1) = \hat{\mathbf{x}}_{k|k}(l) - \big(\mathbf{\Psi}(l)\mathbf{\Psi}^T(l)\big)^{-1}\mathbf{\Psi}(l)\mathbf{g}(\hat{\mathbf{x}}_{k|k}(l)) \tag{2.24}$$

where $\mathbf{\Psi}(l) := \nabla \mathbf{g}(\hat{\mathbf{x}}_{k|k}(l))$ is the Jacobian transpose evaluated at $\hat{\mathbf{x}}_{k|k}(l)$. Substituting $\mathbf{g}(.)$ from (2.23) into (2.24), and applying the matrix inversion lemma to invert the matrix in (2.24), yields (2.19) after straightforward algebraic manipulations. □

When Gauss-Newton iterations in (2.24) are adopted in lieu of IEKF, the resulting error covariance matrix is a function of $\nabla \mathbf{g}$ at the last iteration L given by

$$\mathbf{P}_{k|k} = \left(\mathbf{\Psi}(L)\mathbf{\Psi}^T(L)\right)^{-1}. \tag{2.25}$$

The sparsity-aware EKF formulation in (2.18) is a special case of the general NLS problem in (2.21) corresponding to $\bar{\mathbf{h}}(\mathbf{x}_k) := [(\mathbf{H}\mathbf{x}_k)^T,\ \rho_k(\mathbf{x}_k)]^T$. As a result, the error covariance for the state estimate of (2.18) can be derived from (2.25) as

$$\mathbf{P}_{k|k} = \left(\mathbf{P}_{k|k-1}^{-1} + \mathbf{H}^T\mathbf{R}_k^{-1}\mathbf{H} + \frac{1}{\sigma_k^2}\nabla\rho(\hat{\mathbf{x}}_{k|k}(L))\nabla\rho(\hat{\mathbf{x}}_{k|k}(L))^T\right)^{-1}. \tag{2.26}$$

Compared with (2.10) for the sparsity-agnostic KF, the last summand in (2.26) captures the effect of the sparsity-promoting penalty term on the error covariance.

To enforce the non-negativity constraints in (2.18), one can project each Gauss-Newton iterate in (2.24) onto the non-negative orthant. Unfortunately, this may not generate a convergent sequence [13, p. 215]. To ensure convergence, the projection should be with respect to a different distance metric than the usual Euclidean distance. Upon defining $\mathbf{B}(l) := (\mathbf{\Psi}(l)\mathbf{\Psi}^T(l))^{-1}$, one implements

$$\hat{\mathbf{x}}_{k|k}(l+1) = \left[\hat{\mathbf{x}}_{k|k}(l) - \left(\mathbf{\Psi}(l)\mathbf{\Psi}^T(l)\right)^{-1}\mathbf{\Psi}(l)\mathbf{g}(\hat{\mathbf{x}}_{k|k}(l))\right]_{\mathbf{B}(l)}^{+} \tag{2.27}$$

where $[.]_{\mathbf{B}}^{+}$ denotes projection onto the non-negative orthant, which minimizes the $\|.\|_{\mathbf{B}}^2$ distance instead of the usual $\|.\|_2^2$. If $\rho(\mathbf{x}_k) = \mathbf{x}_k^T\mathbf{1}$, which is equivalent to the ℓ_1-norm for $\mathbf{x}_k \geq \mathbf{0}$, then (2.18) becomes convex and general-purpose convex solvers such as SeDuMi can also be utilized to solve it [103].

The iterative updates in (2.27) and (2.26), along with the prediction step (2.8), constitute the *sparsity-aware TSSG–IEKF tracker*.

2.4.3 Enhanced sparsity-aware KF tracker

As a final note, the sparsity-aware TSSG–KF tracker in Section 2.3 can be enhanced by also casting the ℓ_1-regularized WLS cost in (2.11) as an NLS cost. The ℓ_1-norm term in (2.11) can be equivalently expressed as an extra LS error term for the extra measurement $0 = \sqrt{2\lambda}\sqrt{\mathbf{x}_k^T\mathbf{1}} + u_k$, where u_k is zero-mean noise with unit-variance. The corresponding covariance update can be derived from (2.25) as

$$\mathbf{P}_{k|k} = \left(\mathbf{P}_{k|k-1}^{-1} + \mathbf{H}^T\mathbf{R}_k^{-1}\mathbf{H} + \frac{\lambda}{2\mathbf{x}_k^T\mathbf{1}}\mathbf{1}\mathbf{1}^T\right)^{-1}. \tag{2.28}$$

In all, the state update in (2.11), together with the prediction step in (2.8) and the refined covariance update in (2.28), form the recursions of the *enhanced sparsity-aware TSSG–KF tracker*.

2.5 Position estimation and track formation

The TSSG filters developed so far produce a dynamic TSS map of the operational environment. Such information is adequate to describe the targets' distribution and spatial occupancy over the sensing field of interest, similar in spirit to the PHD filter which portrays the targets' intensity function and the BOF that depicts their occupancy map. In many tracking applications however, more informative estimates such as target positions and trajectories are desired. This section provides TSSG-based solutions to these estimation tasks too.

For the PHD approach, methods performing these extra steps have been reported using particle PHD filters [25,79,91], or Gaussian mixture (GM)-PHD filters [90]. Target positions are typically identified by peak-picking the target intensity function being tracked, and the estimated target positions are treated as measurements for the ensuing data association and track recovery tasks. PHD filters view each particle or each Gaussian component involved

as a target [83,115], and employ conventional target movement models to describe the state transition. As a result, most of the well-known data association methods can be run after PHD filtering [6], [16, Chapters 6-7]. Examples include the auction algorithm proposed in [79], and the joint probabilistic data association (JPDAF) algorithm [85]. Likewise for the BOF, the target movement model is employed in updating the HMM filter as well, which makes it feasible to be combined with a well-established data association method such as the JPDAF [85].

In contrast, the TSSG state equation only models the dynamic behavior of the TSS distribution on the grid, in which grid points are not treated as targets, and hence do not directly obey the conventional target movement model. As remarked in Section 2.1, only partial information about position changes is explicitly captured by the state transition matrix $\mathbf{F}_k$, while other factors such as velocity are implicit. Due to this major difference, conventional data association methods cannot be directly adopted as a follow-up to TSSG filtering. This section develops estimators of target positions and tracks for multi-target scenarios, based solely on the limited information regarding target transition probabilities on the grid.

2.5.1 Target position estimation

Given the output $\hat{\mathbf{x}}_{k|k}$ of the TSSG filter, target positions can be obtained from (2.6) provided that the subset of grid points associated with each target is known in the form of $\mathcal{G}_k^{(m)}$, $\forall m$.

Starting from $\hat{\mathbf{x}}_{k|k}$, one can apply appropriate clustering techniques to identify $\mathcal{G}_k^{(m)}$. When the number of targets M is known, simple parametric clustering methods such as the k-means can be used [15, pp. 424–429]. When M is unknown, one can perform joint clustering and model order selection. Such algorithms utilize some global model order selection criteria such as Akaike's information criterion to determine the best number of clusters

$\hat{M}$, as well as the clusters $\{\hat{\mathcal{G}}_k^{(m)}\}_{m=1}^{\hat{M}}$ themselves [119]. Other nonparametric clustering methods can be employed as well, without assuming or estimating the number of clusters. For example, hierarchical clustering techniques either aggregate or divide the data based on some proximity measure, while density estimation-based nonparametric approaches identify clusters and their number from the modes of the empirical density function of the unknowns; see e.g., [69] for a survey.

Having acquired $\hat{M}$ and $\{\hat{\mathcal{G}}_k^{(m)}\}_{m=1}^{\hat{M}}$, and based on (2.6), the target positions can be obtained individually from the TSSG estimates on the associated clusters of grid points $\forall i \in \hat{\mathcal{G}}_k^{(m)}$, as follows:

$$\hat{\mathbf{p}}_k^{(m)} = \frac{\sum_{i\in\hat{\mathcal{G}}_k^{(m)}} \mathbf{g}_i \hat{x}_{k|k}^{(i)}}{\sum_{i\in\hat{\mathcal{G}}_k^{(m)}} \hat{x}_{k|k}^{(i)}}, \quad m = 1, 2, \ldots, \hat{M}. \tag{2.29}$$

2.5.2 Position-to-track association

Suppose that there are M_t tracks from time slot 1 up to $k-1$, and $\hat{\mathbf{p}}_{k-1}^{(m)}$ has been associated with track t and hence alternatively expressed as $\hat{\mathbf{p}}_{k-1}^{(t)}$, $t = 1, \ldots, M_t$. The goal of track association is to assign the position estimates $\left\{\hat{\mathbf{p}}_k^{(m)}\right\}_{m=1}^{M}$ of the M targets at time k to one of the established M_t tracks. For clarity in exposition, suppose first that $M = M_t$ and there is no target birth or death. This assumption will be removed later on. Evidently, there are $M!$ different assignments, which must be examined to find the best possible association.

Given $\mathbf{y}_{1:k-1}$, the first step is to establish a track prediction model to be used for computing the predicted track positions $\{\hat{\mathbf{p}}_{k|k-1}^{(t)}\}_{t=1}^{M_t}$ and their error covariances. Note from (2.29) that the target position estimates conditioned on the TSSG are independent of the per-sensor measurements. Hence, it suffices to predict $\{\hat{\mathbf{p}}_{k|k-1}^{(t)}\}_t$ solely from the TSSG vector $\hat{\mathbf{x}}_{k-1|k-1}$. To do so, focus on track t and form a $G \times 1$ vector $\check{\mathbf{x}}_{k-1,t}$ that only retains the entries of $\hat{\mathbf{x}}_{k-1|k-1}$ belonging to the t-th cluster of grid points in $\mathcal{G}_{k-1}^{(t)}$; that is, $\check{x}_{k-1,t}^{(j)} = \hat{x}_{k-1}^{(j)}$ for $j \in \mathcal{G}_{k-1}^{(t)}$ and $\check{x}_{k-1,t}^{(j)} = 0$ otherwise, $\forall j$.

Given $\check{\mathbf{x}}_{k-1,t}$ at time $k-1$, the predicted TSSG belonging to track t at time k becomes

$$\check{\mathbf{x}}_{k|k-1,t} = \mathbf{F}_k \check{\mathbf{x}}_{k-1,t}$$

and correspondingly, the predicted track position is

$$\hat{\mathbf{p}}_{k|k-1}^{(t)} = \frac{\sum_{j=1}^{G} \mathbf{g}_j \check{x}_{k|k-1,t}^{(j)}}{\sum_{j=1}^{G} \check{x}_{k|k-1,t}^{(j)}} \,. \tag{2.30}$$

The normalized quantities $\check{x}_{k|k-1,t}^{(j)} / (\sum_{j=1}^{G} \check{x}_{k|k-1,t}^{(j)})$ in (2.30) play the role of fractional weights when the corresponding grid positions $\mathbf{g}_j$ are used to estimate the track position. Viewing $\hat{\mathbf{p}}_{k|k-1}^{(t)}$ as the weighted average of G position-samples $\{\mathbf{g}_j\}_{j=1}^{G}$, it is straightforward to estimate the covariance of $\hat{\mathbf{p}}_{k|k-1}^{(t)}$ using the sample covariance, as

$$\hat{\mathbf{P}}_{k|k-1}^{(t)} = \frac{\sum_{j=1}^{G} \check{x}_{k|k-1,t}^{(j)} (\mathbf{g}_j - \hat{\mathbf{p}}_{k|k-1}^{(t)})(\mathbf{g}_j - \hat{\mathbf{p}}_{k|k-1}^{(t)})^T}{\sum_{j=1}^{G} \check{x}_{k|k-1,t}^{(j)}} \,. \tag{2.31}$$

The process in (2.30)-(2.31) is repeated for all target tracks $t = 1, \ldots, M_t$, so that the prediction estimates and covariances become available for all tracks.

Now, the aim is to associate the predicted track positions $\{\hat{\mathbf{p}}_{k|k-1}^{(t)}\}_t$ in (2.30) with the target position estimates $\{\hat{\mathbf{p}}_k^{(t)}\}_m$ in (2.29). To this end, define the decision variables $a(t,m) \in \{0,1\}$ for $t = 1, \ldots, M_t$ and $m = 1, \ldots, M$, where $a(t,m) = 1$ amounts to deciding that target m measured at $\hat{\mathbf{p}}_k^{(m)}$ is assigned to track t. The pairwise-association cost can be quantified using the Mahalanobis distance between track t's prediction and $\hat{\mathbf{p}}_k^{(m)}$ as a measurement, that is

$$\mathrm{MD}(t,m) := \left(\hat{\mathbf{p}}_{k|k-1}^{(t)} - \hat{\mathbf{p}}_k^{(m)}\right)^T (\hat{\mathbf{P}}_{k|k-1}^{(t)})^{-1} \left(\hat{\mathbf{p}}_{k|k-1}^{(t)} - \hat{\mathbf{p}}_k^{(m)}\right) \,. \tag{2.32}$$

The following optimization problem is formulated to minimize the total association cost subject to linear constraints that ensure one-to-one track-to-measurement mapping:

$$\min_{a(t,m)\in\{0,1\}} \quad \sum_{t=1}^{M} \sum_{m=1}^{M} a(t,m) \mathrm{MD}(t,m) \tag{2.33}$$

$$\text{such that} \quad \sum_{m=1}^{M} a(t,m) = 1, \ \forall t = 1, \ldots, M_t,$$
$$\sum_{t=1}^{M} a(t,m) = 1, \ \forall m = 1, \ldots, M.$$

It is worth mentioning that (2.33) is a special case of the so called assignment problem, which is a well-known data association algorithm [16, pp. 342–349]. Its solution can be efficiently computed in polynomial time using integer programming solvers such as the Hungarian algorithm [75].

The track association problem in (2.33) can be modified to handle track birth and death scenaria [16]. Toward this objective, introduce a dummy target $m = 0$ and a dummy track $t = 0$. The one-to-one constraints in (2.33) are modified as follows: each track is assigned to at most one target position measurement, but the dummy track can be associated with any number of targets; meanwhile, each position measurement is assigned to at most one track, but the dummy measurement can be assigned to multiple tracks; further, the dummy target cannot be associated with the dummy track. Such a modified association problem resembles the auction algorithm [16, 79], along with the corresponding association costs defined in (2.32). The computational burden of this combinatorial problem can be reduced by removing some unlikely association pairs in advance. Essentially, if for a track t all the association costs $\{\mathrm{MD}(t,m)\}_m$ exceed a large threshold, then this track is considered "dead", and is associated with the dummy target. Similarly for a target m, if all the association costs $\{\mathrm{MD}(t,m)\}_t$ are too large, then this target is considered "born", and is associated with the dummy track.

Once the position-to-track association is completed, velocity estimates can be obtained too. This is possible by subtracting target position at time $k-1$ from its position at time k and dividing by the sampling period.

Finally, it is worth noting that in formulating (2.33), only the state transition probability

matrix $\mathbf{F}_k$ is needed, regardless of the underlying target movement model. It is possible however to utilize each target's movement model to develop other (more effective) data association schemes, and refine the track estimates as well. Such association and track refinement steps will take place after every TSSG update, using the output of the TSSG tracker to form the position-measurements (2.29) for the ensuing parallel target trackers, one for each target. The results will not be fed back to the TSSG trackers, thus ensuring resilience of TSSG estimates to data mis-association and track estimation errors.

2.6 Numerical tests

Consider a 300×300 square-meter surveillance region along with a 10×10 rectangular grid with equally-spaced grid points. Therefore, each grid cell is of size 30×30. Simulations are performed for both single- and multi-target scenarios.

2.6.1 Single-target case

A single target starts at the south-west corner of the grid at time $k = 1$, and moves northeast according to a constant velocity model

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \bar{\mathbf{v}} T_s + \mathbf{n}_k$$

where $\bar{\mathbf{v}}$ denotes the target's constant velocity assumed known and given by $\bar{\mathbf{v}} = (15, 15)$ meters per second; $\mathbf{p}_{k-1}$ is the previous target position; $T_s = 1$ is the sampling time in seconds; and $\mathbf{n}_k$ represents modeling noise of zero-mean and variance $\sigma_n^2 \mathbf{I}_2$. Given this model and ignoring $\mathbf{n}_k$, if the target starts at the center of the grid cell it is currently in, then at the next time instant, it will arrive at the northeast corner of this grid cell conjoining the north, east, and northeast grid cells. Due to the symmetrically distributed noise, the target will have equal probability of falling inside each of the 4 grid cells. It is assumed

that σ_n is small enough so that the probability of a target moving into grid cells other than its four adjacent ones is negligible. The resultant movement model is as follows: a target stays on the current grid point with probability 1/4, and moves north, east, or northeast with probability 1/4. Whenever the target moves outside the boundaries of the surveillance region, tracking stops. One random realization of this movement model is plotted in Fig. 2.2 and is considered for the ensuing simulations starting with the single-target case. The target's signal strength is $s = 10$, and there are $N = 20$ sensors distributed randomly over the surveillance region measuring the received TSS. The measurement noise $\mathbf{v}_k$ is zero-mean Gaussian white with unit variance. The propagation function $h(x)$ in (2.1) is given by $h(x) = c/(c + x^2)$ for $x \geq 0$, where c is chosen so that $h(60) = 0.5$. Apparently, $h(0) = 1$ and $h(x)$ is monotonically decreasing as x increases.

The proposed sparsity-agnostic and sparsity-aware TSSG-KF trackers in Sections 2.2-2.3 are employed to estimate the target signal strengths and position vectors over time. The position estimation accuracy is measured by the average root mean-square error (RMSE) in the form of RMSE $= \sqrt{\frac{1}{K_{\max}} \sum_{k=1}^{K_{\max}} \|\hat{\mathbf{p}}_k - \mathbf{p}_k\|_2^2}$, where $K_{\max}$ is the tracking duration and $\hat{\mathbf{p}}_k$ is obtained as in (2.6). The covariance matrix of the process noise $\mathbf{w}_k$ is set to $\mathbf{Q}_k = \mathbf{I}_G$ in (2.8), and $1,000$ Monte Carlo runs over the random measurement noise are performed to compute the RMSE. To shed light on the role of the ℓ_1-norm sparsity penalty term in (2.11), Fig. 2.3 depicts the RMSE performance with respect to the sparsity-controlling coefficient λ_k as a fraction of $\bar{\lambda}_k$ in (2.13). The sparsity-agnostic tracker corresponds to setting $\lambda = 0$ in (2.11), and is also plotted for comparison. It is seen that the sparsity-aware KF tracker outperforms the sparsity-agnostic one for a large range of $\lambda_k \neq 0$ values, and $\lambda_k = 0.1\bar{\lambda}_k$ appears to yield the lowest RMSE for this test. The optimal HMM filter exhibits the best performance, but requires accurate knowledge of the target signal strength.

Fig. 2.4 depicts the RMSE of the sparsity-aware TSSG-IEKF tracker of (2.18), with $\mu_k = 1$ and for different values of σ_k. This tracker incorporates sparsity as an extra mea-

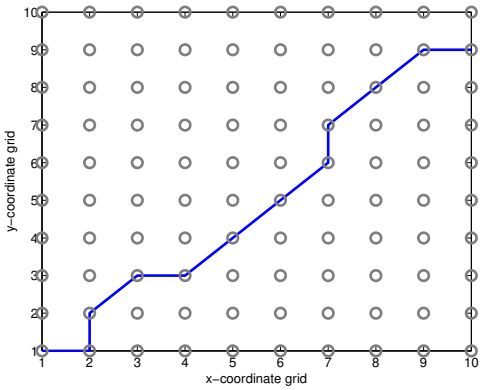

Figure 2.2: True target track on the grid.

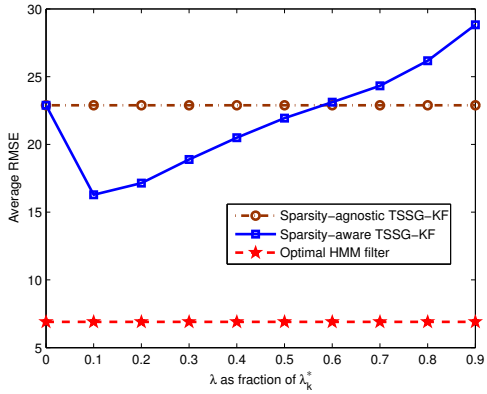

Figure 2.3: Sparsity-agnostic and sparsity-aware TSSG-KF trackers.

surement, and selects the sparsity model $\rho(\mathbf{x}_k)$ as the ℓ_1-norm function. Evidently, this extra measurement is effective in promoting sparsity, which leads to improved performance relative to the sparsity-agnostic tracker. The noise variance σ_k^2 of the sparsity measurement in (2.17b) is a design parameter chosen in accordance with the sensor measurements (here having unit variance). As Fig. 2.4 indicates, there is an optimal value of σ_k that attains the most effective tradeoff between the sensor measurements and the sparsity-induced measurement. As σ_k becomes larger, the tracker collects less information from the extra measurement, and eventually becomes sparsity-agnostic when σ_k is too large. On the other hand, when σ_k is too small, the tracker is predominantly enforcing a sparse solution without considering much the sensor measurements, which also degrades tracking performance.

Both sparsity-aware TSSG trackers, the TSSG-KF tracker with $\lambda_k = 0.1\bar{\lambda}_k$ and the TSSG-IEKF tracker with $\sigma_k = 2$, are compared in Fig. 2.5 in terms of their RMSE performance versus time. The curves are generated using $1,000$ Monte Carlo runs. These two sparsity-aware trackers exhibit similar performance, both outperforming the sparsity-agnostic tracker. The clairvoyant optimal HMM filter is also tested as the benchmark.

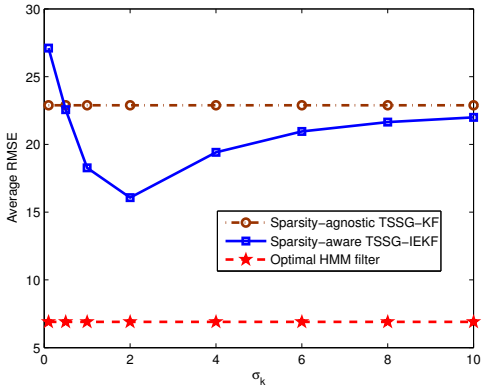

Figure 2.4: TSSG-IEKF tracker with an extra sparsity measurement.

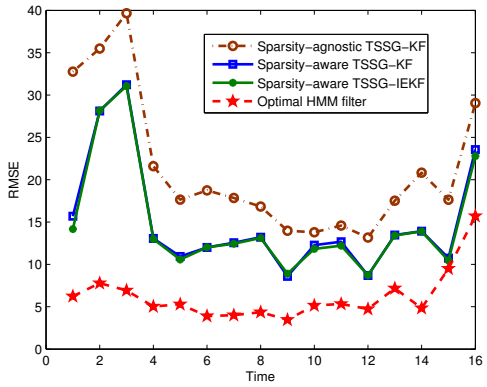

Figure 2.5: Comparison of TSSG-KF and TSSG-IEKF trackers.

Finally, Fig. 2.6 demonstrates the dynamic behavior of the sparsity-aware estimator in (2.11) with $\lambda_k = 0.9\bar{\lambda}_k$. Even though the sparsity-aware TSSG-KF performs worse than sparsity-agnostic TSSG-KF for this value of λ_k, it is chosen to demonstrate how sparsity affects the tracking process. The estimated TSSG state vectors are depicted over time, with a circle representing a nonzero TSS at the corresponding grid point. The true and estimated tracks are plotted as well. For clarity, only the projection of the target track on the y-direction is depicted. It is seen that the "cloud" of nonzero target signal strengths follows the true track. The estimated target profile is seen to be indeed spatially sparse. The size of the nonzero support indicates the uncertainty in target position estimates, which apparently does not grow over time, even when using a simple grid-induced linear KF tracker to follow the state transition pattern.

2.6.2 Multi-target case

Two targets are respectively located at the south-center and west-center of the grid at time $k = 1$. They start moving according to the same movement model used for the single-target

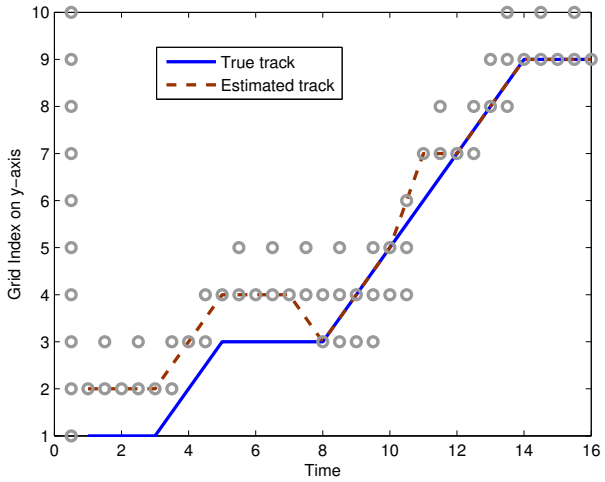

Figure 2.6: Nonzero support of estimated TSSG, true, and estimated tracks (y-direction only).

case. Fig. 2.7 plots one random realization of these target trajectories used for the ensuing multi-target test cases. Adhering to as1), these two trajectories do not overlap on the same grid point at the same time. The target signal strengths are set to be $s^{(1)} = s^{(2)} = 10$. It is assumed that the trackers know the number of targets unless otherwise stated. There are 100 sensors deployed randomly over the surveillance region to measure the total received signal strengths.

First, the position estimation method presented in Subsection 2.5.1 is tested. Fig. 2.7 depicts the position estimates as circles along with the true target trajectories, for both the sparsity-agnostic TSSG-KF and the sparsity-aware TSSG-KF trackers with $\lambda_k = 0.1\bar{\lambda}_k$. When the ℓ_1-norm sparsity-promoting regularization term is not present (cf. Fig. 2.7), position estimates are rather inaccurate and some of them fall far from either of the two targets. In contrast, the sparsity-aware TSSG-KF in Fig. 2.7 results in quite accurate position estimates. One can clearly associate each position estimate with one of the two

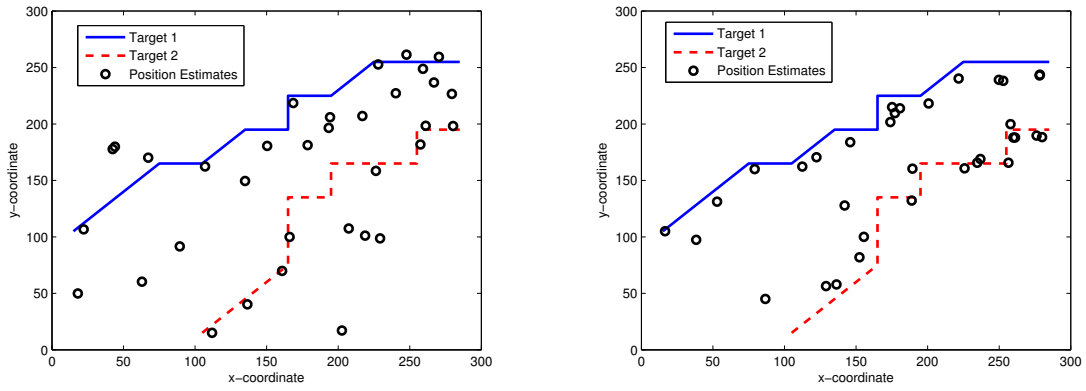

Figure 2.7: True tracks and position estimates for two targets: (left) sparsity-agnostic TSSG-KF tracker, (right) sparsity-aware TSSG-KF tracker. Circles indicate the estimated target positions.

targets, and readily visualize target tracks from the position estimates. Before the position estimates are associated with individual targets, a pertinent performance metric quantifying estimation accuracy is the so-called Wasserstein distance (WD) that measures the distance between two finite sets [26]. Let $P_k = \{\mathbf{p}_k^{(m)}\}_m$ denote the finite set of the true target positions at time k and $\hat{P}_k = \{\hat{\mathbf{p}}_k^{(n)}\}_n$ the set of position estimates, respectively. Let $d(.,.)$ stand for the Euclidean ℓ_2-norm, and $|\cdot|$ for set cardinality. The L^p WD between these two sets is defined as

$$\begin{aligned} d_p^W(P_k, \hat{P}_k) = \quad & \min_{\{C_{mn}\}} \left(\sum_{\mathbf{p}^{(m)} \in P_k} \sum_{\hat{\mathbf{p}}^{(n)} \in \hat{P}_k} C_{mn} d\left(\mathbf{p}^{(m)}, \hat{\mathbf{p}}^{(n)}\right)^p \right)^{1/p} \\ \text{subject to} \quad & \sum_{m=1}^{|P_k|} C_{mn} = \frac{1}{|\hat{P}_k|}, \ \forall n = 1, \ldots, |\hat{P}_k| \\ & \sum_{n=1}^{|\hat{P}_k|} C_{mn} = \frac{1}{|P_k|}, \ \forall m = 1, \ldots, |P_k|. \end{aligned}$$

Fig. 2.8 depicts the L^1 WD for both sparsity-aware TSSG-KF and TSSG-IEKF trackers, in comparison with the sparsity-agnostic TSSG-KF tracker. The TSSG-IEKF tracker is implemented with $\mu_k = 2$ and $\sigma_k = 2$. The WD is evaluated by averaging over $1,000$ Monte Carlo runs for each tracker. Evidently, both sparsity-aware designs are effective and improve the WD performance.

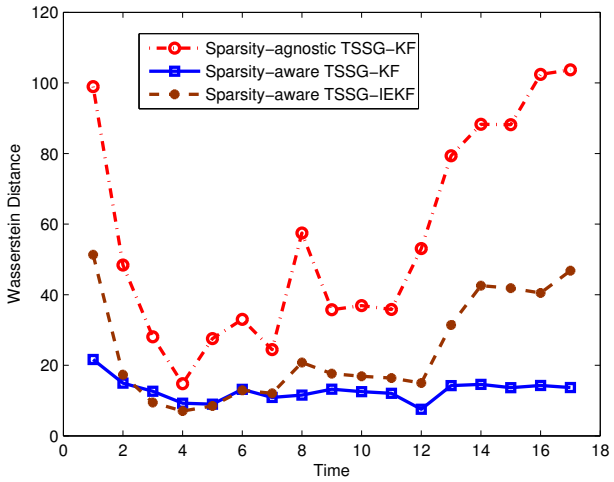

Figure 2.8: WD versus time for sparsity-agnostic and sparsity-aware TSSG-KF and TSSG-IEKF trackers.

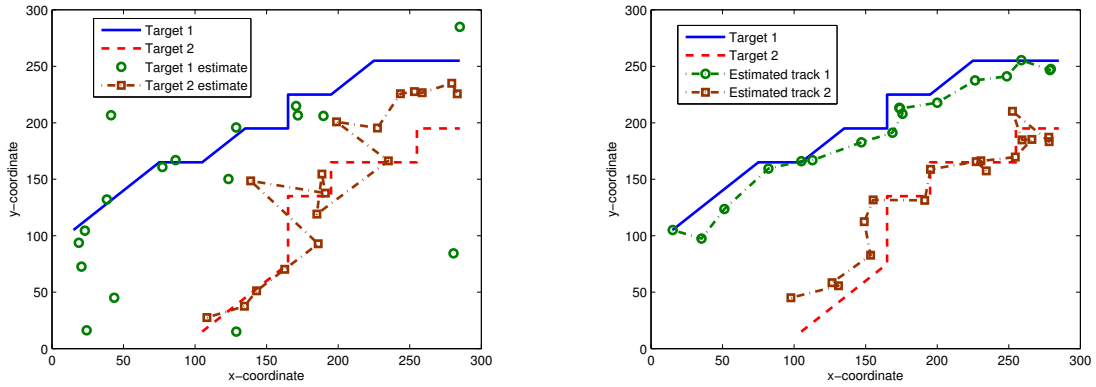

Figure 2.9: True and estimated tracks: (left) sparsity-agnostic TSSG-KF; (right) sparsity-aware TSSG-KF;

The track formation algorithm of Subsection 2.5.2 is investigated next for the same target realization. The target tracks formed using the position estimates of a single Monte

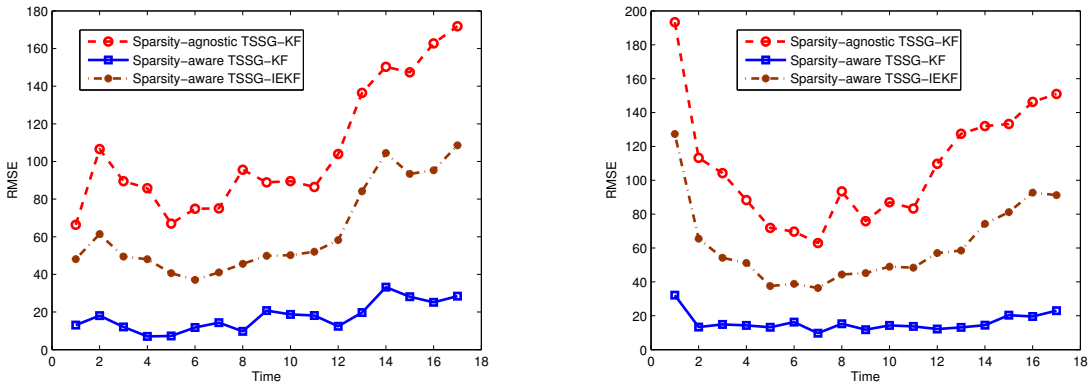

Figure 2.10: Tracking performance for multi-target case: (left) RMSE for target 1, (right) RMSE for target 2.

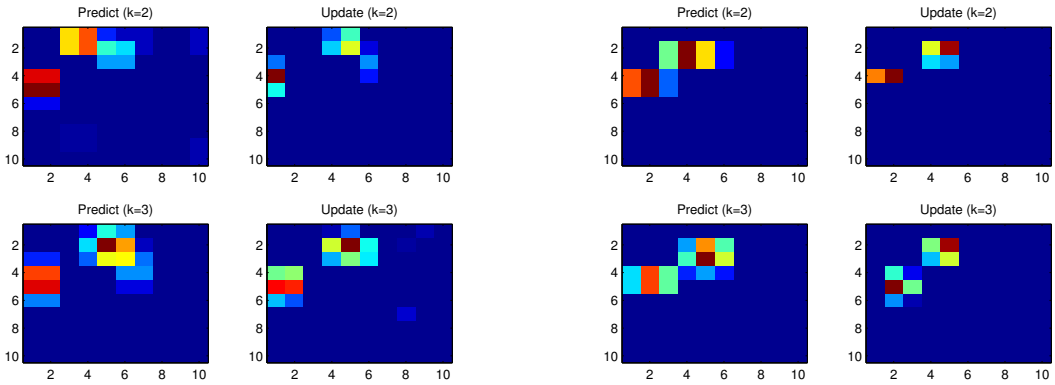

Figure 2.11: Heat map: (left) sparsity-agnostic TSSG-KF tracker, (right) sparsity-aware TSSG-KF tracker.

Carlo run are plotted in Fig. 2.9, for the sparsity-agnostic TSSG-KF tracker. The estimated track for target 1 is not even plotted because it deviates too much from the true trajectory. The estimated track for target 2 shows some erratic behavior. As will be discussed shortly,

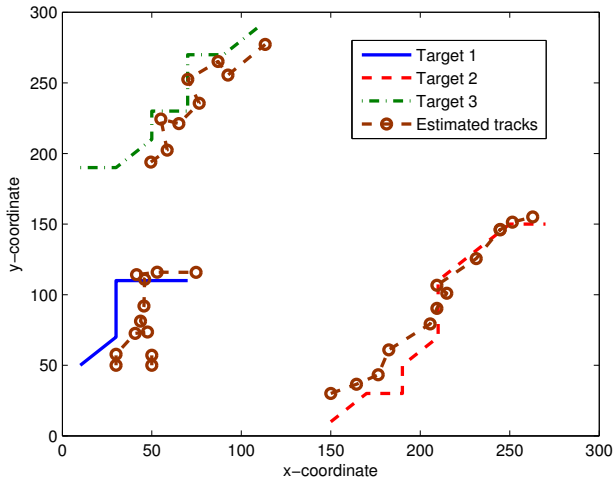

Figure 2.12: True and estimated tracks with unknown number of clusters.

the unsatisfactory performance is not due to the proposed track formation algorithm itself; rather, it is a manifestation of inaccurate clustering that results from badly shaped TSSG estimates to begin with. The accuracy of the TSS map provided by the TSSG filters is essential in ensuring good performance of position estimates and track formation algorithms. Fig. 2.9 illustrates the track estimates obtained after processing the sparsity-aware TSSG-KF output. It can be seen that both targets are closely tracked. To compare these methods quantitatively, the RMSE curves for the two targets are plotted versus time in Fig. 2.10, for $1,000$ Monte Carlo runs. It is evident that exploitation of sparsity markedly improves performance of the TSSG filters. In addition, sparsity-aware TSSG-KF seems to outperform the TSSG-IEKF for this specific setting and choice of parameters.

To further illustrate the importance of TSSG estimation for subsequently forming position and track estimates, Fig. 2.11 depicts two snapshots of the TSSG heat maps after the KF prediction and correction steps at times $k = 2$ and 3. For the sparsity-agnostic TSSG-KF tracker, the correction heat map at $k = 2$ seems to contain three clusters while

there are only two targets. In the correction heat map at $k = 3$, there is a single point in the lower right which is nonzero and far from both targets. This spurious point can have a detrimental effect during the clustering phase as it can greatly shift mean positions of the two clusters. These malign effects do not show up in the TSSG heat maps for the sparsity-aware TSSG-KF, where heat maps exhibit two compact clusters in both KF correction steps.

Lastly, simulations for an unknown number of targets are performed on a 15×15 grid with the true and estimated target tracks plotted in Fig. 2.12. In this setup, targets 1 and 2 begin their movement at time $k = 1$; at $k = 5$ target 3 is born, and at $k = 10$ target 1 disappears. The sparsity-aware TSSG-KF is utilized in both simulations. Various clustering options are available when the number of clusters is unknown [119]. Here a simple MATLAB routine called "silhouette" is used to determine the best number of natural clusters in the TSS maps. After k-means clustering is performed, silhouette returns a value between -1 and 1 for every point that has participated in the clustering phase. The value that silhouette returns measures how well every point is explained by the cluster it belongs to, compared to other clusters. A value close to 1 is desirable. Therefore, silhouette values averaged over the clustered points offer a good measure of how well clusters explain the points which belong to them. The number of clusters with the largest average silhouette value is selected as the most appropriate number of clusters. It can be seen that the three targets are accurately tracked. However, a small erroneous track emerges close to target 1 for two time periods. Unfortunately, performance of the case with unknown number of targets is not always as accurate as shown here and more than one inaccurate track may arise. On the other hand, when applied to the two-target example previously considered in the absence of target births or deaths, the algorithm with unknown number of targets is always successful in recovering accurate target tracks.

2.7 Summary

The problem of tracking multiple targets on a plane using the superposition of their received signal strengths as measurements has been investigated. A grid-based state space model was introduced to describe the dynamic behavior of target signal strengths. This model not only renders the nonlinear estimation problem linear, but also facilitates incorporation and exploitation of the grid-induced sparsity present. Two sparsity-aware Kalman trackers were developed to exploit this sparsity attribute: TSSG-KF promoting sparsity of the state estimates through ℓ_1-norm minimization, and TSSG-IEKF effecting sparsity by viewing it as an extra measurement. To address the challenge of updating the state estimation error covariances under sparsity constraints, a novel approach based on iterative extended KF and measurement augmentation was also developed to provide tractable and accurate covariance updates. Position estimation and position-to-track association issues were considered as well. The proposed trackers do not require knowing the number of targets or their signal strengths, and considerably reduce complexity when compared to the optimal hidden Markov model filter. They offer improved tracking performance at reduced sensing and computational cost, especially when compared to sparsity-agnostic trackers.

2.8 Appendix

2.8.1 State transition model

From the total probability argument, it holds that

$$p\left(x_k^{(j)} \neq 0 \,\middle|\, j \in \mathcal{G}_k^{(m)}\right) = \sum_{i=1}^{G} p\left(x_k^{(j)} \neq 0, x_{k-1}^{(i)} \neq 0, i \in \mathcal{G}_{k-1}^{(m)} \,\middle|\, j \in \mathcal{G}_k^{(m)}\right)$$

which leads to the following equality after invoking as2) in Bayes' rule[2]:

$$p\left(x_k^{(j)} \neq 0 \,\middle|\, j \in \mathcal{G}_k^{(m)}\right) = \sum_{i=1}^{G} f_k^{(ji)} p\left(x_{k-1}^{(i)} \neq 0, i \in \mathcal{G}_{k-1}^{(m)}\right) = \sum_{i \in \mathcal{G}_{k-1}^{(m)}} f_k^{(ji)} p\left(x_{k-1}^{(i)} \neq 0 \,\middle|\, i \in \mathcal{G}_{k-1}^{(m)}\right). \tag{2.34}$$

Any grid point $j = 1, \ldots, G$ with a nonzero $x_k^{(j)} \neq 0$ is associated with a single target index $m_k^{(j)} \in [1, M]$ at time k, which means $p(x_k^{(j)} \neq 0, j \in \mathcal{G}_k^{(m_k^{(j)})}) \neq 0$ for $m_k^{(j)} \in [1, M]$; and according to as1), $p(x_k^{(j)} \neq 0, j \in \mathcal{G}_k^{(m)}) = 0$, $\forall m \neq m_k^{(j)}$ or $m = m_k^{(j)} = 0$. Invoking $p(x_k^{(j)} \neq 0) = \sum_{m=0}^{M} p(x_k^{(j)} \neq 0, j \in \mathcal{G}_k^{(m)})$, and noting that $p(j \in \mathcal{G}_k^{(m_k^{(j)})}) = 1$, yields

$$p(x_k^{(j)} \neq 0) = p(x_k^{(j)} \neq 0, j \in \mathcal{G}_k^{(m_k^{(j)})}) = p(x_k^{(j)} \neq 0 | j \in \mathcal{G}_k^{(m_k^{(j)})}), \quad \forall j. \tag{2.35}$$

Similarly for a grid point i at time $(k-1)$, there exists a target index $m_{k-1}^{(i)} \in [0, M]$ such that $p(x_{k-1}^{(i)} \neq 0) = p(x_{k-1}^{(i)} \neq 0, i \in \mathcal{G}_{k-1}^{(m_{k-1}^{(i)})})$, and $p(x_{k-1}^{(i)} \neq 0, i \notin \mathcal{G}_{k-1}^{(m_{k-1}^{(i)})}) = 0$, $\forall i \in [1, G]$. Under as1) and as2), it follows from (2.34) and (2.35) that

$$\begin{aligned} x_k^{(j)} &= s^{(m_k^{(j)})} p\left(x_k^{(j)} \neq 0\right) = s^{(m_k^{(j)})} p\left(x_k^{(j)} \neq 0 \,\middle|\, j \in \mathcal{G}_k^{(m_k^{(j)})}\right) = s^{(m_k^{(j)})} \sum_{i=1}^{G} f_k^{(ji)} p\left(x_{k-1}^{(i)} \neq 0, i \in \mathcal{G}_{k-1}^{(m_k^{(j)})}\right) \\ &= \sum_{\forall i: m_{k-1}^{(i)} = m_k^{(j)}} f_k^{(ji)} s^{(m_{k-1}^{(i)})} p\left(x_{k-1}^{(i)} \neq 0, i \in \mathcal{G}_{k-1}^{(m_{k-1}^{(i)})}\right) + s^{(m_k^{(j)})} \sum_{\forall i: m_{k-1}^{(i)} \neq m_k^{(j)}} f_k^{(ji)} \underbrace{p\left(x_{k-1}^{(i)} \neq 0, i \notin \mathcal{G}_{k-1}^{(m_{k-1}^{(i)})}\right)}_{=0,\ \forall i} \end{aligned}$$

[2]It holds trivially for the dummy target $m = 0$ as well, because $p(x_k^{(j)} \neq 0 | j \in \mathcal{G}_k^{(0)}) = 0$ and $p(x_k^{(j)} \neq 0, j \in \mathcal{G}_k^{(0)}) = 0$.

$$= \sum_{\forall i: m_{k-1}^{(i)} = m_k^{(j)}} f_k^{(ji)} s^{(m_{k-1}^{(i)})} p\left(x_{k-1}^{(i)} \neq 0\right) + \sum_{\forall i: m_{k-1}^{(i)} \neq m_k^{(j)}, m_{k-1}^{(i)} = 0} f_k^{(ji)} s^{(0)} \underbrace{p\left(x_{k-1}^{(i)} \neq 0, i \in \mathcal{G}_{k-1}^{(m_{k-1}^{(i)})}\right)}_{=0,\ \forall i:\ m_{k-1}^{(i)} = 0}$$

$$= \sum_{\forall i: m_{k-1}^{(i)} = m_k^{(j)}} f_k^{(ji)} s^{(m_{k-1}^{(i)})} p\left(x_{k-1}^{(i)} \neq 0\right) + \sum_{\forall i: m_{k-1}^{(i)} \neq m_k^{(j)}, m_{k-1}^{(i)} = 0} f_k^{(ji)} s^{(m_{k-1}^{(i)})} p\left(x_{k-1}^{(i)} \neq 0 | m_{k-1}^{(i)} = 0\right)$$

$$= \sum_{i=1}^{G} f_k^{(ji)} x_{k-1}^{(i)}, \quad \forall j \in [1, G]. \tag{2.36}$$

Chapter 3

Doubly robust smoothing of dynamical processes via outlier sparsity constraints

Coping with outliers contaminating dynamical processes is of major importance in various applications because mismatches from nominal models are not uncommon in practice. In this context, the present chapter develops novel fixed-lag and fixed-interval smoothing algorithms that are robust to outliers simultaneously present in the measurements *and* in the state dynamics. Outliers are handled through auxiliary unknown variables that are jointly estimated along with the state based on the least-squares criterion that is regularized with the ℓ_1-norm of the outliers in order to effect sparsity control. The resultant iterative estimators rely on coordinate descent and the alternating direction method of multipliers, are expressed in closed form per iteration, and are provably convergent. Additional attractive features of the novel doubly robust smoother include: i) ability to handle both types of outliers; ii) universality to unknown nominal noise and outlier distributions; iii) flexibility

to encompass maximum a posteriori optimal estimators with reliable performance under nominal conditions; and iv) improved performance relative to competing alternatives at comparable complexity, as corroborated via simulated tests.

3.1 Problem statement and preliminaries

Consider the following *outlier-aware* state-space model

$$\mathbf{x}_n = \mathbf{F}_n\mathbf{x}_{n-1} + \mathbf{w}_n + \mathbf{o}_{x,n}, \quad n = 1, \ldots, N \tag{3.1a}$$

$$\mathbf{y}_n = \mathbf{H}_n\mathbf{x}_n + \mathbf{v}_n + \mathbf{o}_{y,n}, \quad n = 1, \ldots, N \tag{3.1b}$$

where $\mathbf{x}_n \in \mathbb{R}^{D_x}$ and $\mathbf{y}_n \in \mathbb{R}^{D_y}$ denote the state and measurement vectors at time n, respectively; $\mathbf{w}_n$ and $\mathbf{v}_n$ are mutually independent, zero-mean nominal noise vectors, each independent across time, and from the initial state $\mathbf{x}_0$, with respective covariance matrices $\{\mathbf{Q}_n, \mathbf{R}_n\}_{n=1}^N$; $\mathbf{x}_0$ has mean $\mathbf{m}_0$ and covariance $\mathbf{\Sigma}_0$; and $\{\mathbf{o}_{x,n}, \mathbf{o}_{y,n}\}_{n=1}^N$ represent the unknown state and measurement outlier vectors.

Given $\{\mathbf{F}_n, \mathbf{H}_n, \mathbf{Q}_n, \mathbf{R}_n, \mathbf{y}_n\}_{n=1}^N$, $\mathbf{m}_0$, and $\mathbf{\Sigma}_0$, the goal of *fixed-interval* DRS is to estimate $\{\mathbf{x}_n\}_{n=1}^N$ and $\{\mathbf{o}_{x,n}, \mathbf{o}_{y,n}\}_{n=1}^N$. Different from [5, 84, 101, 107], note that the outliers are explicitly introduced and treated as unknown variables to be estimated. This problem can be cast as one of linear regression, since $\mathbf{x}_n = \mathbf{F}_n\mathbf{x}_{n-1} + \mathbf{o}_{x,n} + \mathbf{w}_n$ can be viewed as an extra "zero measurement" $\mathbf{0} = -\mathbf{x}_n + \mathbf{F}_n\mathbf{x}_{n-1} + \mathbf{o}_{x,n} + \mathbf{w}_n$; and similarly for the initial condition as $-\mathbf{m}_0 = -\mathbf{x}_0 + \mathbf{w}_0$, where $\mathbf{w}_0$ is zero-mean with covariance $\mathbf{\Sigma}_0$. Thus, (3.1) can

be expressed in a matrix-vector form as

$$\left[\begin{array}{cccc} -\mathbf{I} & & & \\ \mathbf{F}_1 & -\mathbf{I} & & \\ & \ddots & \ddots & \\ & & \mathbf{F}_N & -\mathbf{I} \\ \hline \mathbf{0} & \mathbf{H}_1 & & \\ \vdots & & \ddots & \\ \mathbf{0} & & & \mathbf{H}_N \end{array}\right] \left[\begin{array}{c} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{array}\right] + \left[\begin{array}{c} \mathbf{0} \\ \mathbf{o}_{x,1} \\ \vdots \\ \mathbf{o}_{x,N} \\ \hline \mathbf{o}_{y,1} \\ \vdots \\ \mathbf{o}_{y,N} \end{array}\right] + \left[\begin{array}{c} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_N \\ \hline \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_N \end{array}\right] = \left[\begin{array}{c} -\mathbf{m}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \hline \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{array}\right] \tag{3.2}$$

or in a more compact form (with obvious definitions) as

$$\mathbf{A}\mathbf{x} + \mathbf{o} + \mathbf{w} = \mathbf{y} \tag{3.3}$$

where matrix $\mathbf{A}$ is tall, and vector $\mathbf{w}$ has block diagonal covariance matrix $\mathbf{Q}_w := \text{diag}(\mathbf{\Sigma}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_N, \mathbf{R}_1, \ldots, \mathbf{R}_N)$. Since both $\mathbf{x}$ and $\mathbf{o}$ are unknown, the linear system in (3.2) is clearly under-determined.

When there are no outliers (cf. $\mathbf{o} = \mathbf{0}$) and $\mathbf{A}$ is full rank, the WLS estimate [cf. (3.3)] $\hat{\mathbf{x}} := \arg\ \min_{\mathbf{x}} (\mathbf{y} - \mathbf{A}\mathbf{x})^T \mathbf{Q}_w^{-1} (\mathbf{y} - \mathbf{A}\mathbf{x})$ yields the KS. Substituting from (3.2), this estimate can also be written as [2, p. 189]

$$\widehat{\mathbf{x}}^{\text{KS}} := \arg\ \min_{\mathbf{x}}\ \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{H}_n\mathbf{x}_n\|^2_{\mathbf{R}_n^{-1}} + \frac{1}{2}\|\mathbf{x}_0 - \mathbf{m}_0\|^2_{\mathbf{\Sigma}_0^{-1}} + \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{F}_n\mathbf{x}_{n-1}\|^2_{\mathbf{Q}_n^{-1}} \tag{3.4}$$

where $\|\mathbf{x}\|^2_{\mathbf{M}} := \mathbf{x}^T\mathbf{M}\mathbf{x}$. The estimate $\widehat{\mathbf{x}}^{\text{KS}}$ is also known as the Rauch-Tung-Striebel (RTS) smoother [96]. It is both minimum mean-square error (MMSE) and maximum a posteriori (MAP) optimal if the initial state and all nominal noise vectors are Gaussian; otherwise, it is linear (L)MMSE optimal. In fact, adding to the WLS cost in (3.4) a ridge regularization term $\lambda\|\mathbf{x}\|_2^2$ to constrain the ℓ_2 norm of $\mathbf{x}$, the resultant ridge WLS, as well as the (L)MMSE and MAP, all yield a unique estimate (even for under-determined models), and can be

rendered equivalent depending on the assumptions and corresponding optimality claims one is willing to make. The exposition henceforth is centered around the (regularized) WLS approach, because it is universal with respect to (wrt) the underlying probability density functions.

With $\mathbf{o} = \mathbf{0}$ (or known for that matter), the state can be clearly estimated by solving the equations $(\mathbf{A}^T\mathbf{Q}_w^{-1}\mathbf{A})\mathbf{x} = \mathbf{A}^T\mathbf{Q}_w^{-1}(\mathbf{y}-\mathbf{o})$, where the matrix $\mathbf{A}^T\mathbf{Q}_w^{-1}\mathbf{A}$ has a *block tridiagonal* structure [cf. (3.2) and (3.3)]. This allows obtaining the solution in batch form at complexity which is linear in N [58, p. 174]. Alternatively, one can use the forward-backward algorithm in e.g., [2, p. 189] or [96] to solve (3.4) recursively. The forward direction is a KF followed by the backward run, which smooths the filtered estimates. The forward-backward algorithm also exhibits linear complexity in N. In a nutshell, both batch and recursive solvers of (3.2)-(3.4) exhibit low complexity (linear in N) when $\mathbf{o}$ is known.

If unknown outliers $\mathbf{o}$ are present in (3.3), and one chooses to ignore them and run a clairvoyant KS as if $\mathbf{o}$ were absent, the MSE performance will be poor because the (W)LS criterion is known to be severely affected by outliers [67]. This mandates dealing with the outliers in (3.3) explicitly – a challenge addressed in the next section by exploiting sparsity constraints on $\mathbf{o}$.

3.2 Robustness by controlling outlier sparsity

The under-determinacy in (3.3) when $\mathbf{o}$ is unknown, raises non-uniqueness and thus state identifiability issues. Ridge WLS, (L)MMSE, and MAP estimators cannot recover the exact $\mathbf{x}$, a fact confirmed by the nominal-noise-free setup [cf. $\mathbf{w} = \mathbf{0}$ in (3.3)], where one faces an under-determined system of linear equations generally admitting infinite solutions. Key to addressing this issue is the degree of *sparsity* (number of nonzero entries) of the vector $\mathbf{o}$ – an attribute offering the potential for solving uniquely under-determined systems of

linear equations, as established recently in the context of compressive sampling [21]. This motivates recovery of a controllably sparse estimate of $\mathbf{o}$ by effecting sparsity through an ℓ_0-(pseudo)norm regularization term. Specifically, the proposed robust smoother aims at

$$[\hat{\mathbf{x}}, \hat{\mathbf{o}}] := \arg \min_{\mathbf{x},\mathbf{o}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{o}\|^2_{\mathbf{Q}_w^{-1}} + \lambda\|\mathbf{o}\|_0 \tag{3.5}$$

where the scalar λ is used to control the degree of sparsity in $\mathbf{o}$. The level of outlier sparsity can be selected by tuning λ, and the outliers can then be estimated jointly with the state via (3.5). Unfortunately, the ℓ_0-norm renders the problem non-convex and in fact NP-hard, which suggests a convex relaxation using the closest *convex* approximation to the ℓ_0-norm, namely the ℓ_1-norm [21, 112].

Using an ℓ_1-norm regularization and defining $\mathbf{o}_x := [\mathbf{o}_{x,1}^T, \ldots, \mathbf{o}_{x,N}^T]^T$ as the state outlier vector, and $\mathbf{o}_y := [\mathbf{o}_{y,1}^T, \ldots, \mathbf{o}_{y,N}^T]^T$ as the measurement outlier vector, the novel DRS approach amounts to [cf. (3.1)-(3.5)]

$$[\widehat{\mathbf{x}}^{\mathrm{DRS}}, \widehat{\mathbf{o}}_x, \widehat{\mathbf{o}}_y] := \arg \min_{\mathbf{x},\mathbf{o}_x,\mathbf{o}_y} \left\{ \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{H}_n\mathbf{x}_n - \mathbf{o}_{y,n}\|^2_{\mathbf{R}_n^{-1}} + \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{x}_n - \mathbf{F}_n\mathbf{x}_{n-1} - \mathbf{o}_{x,n}\|^2_{\mathbf{Q}_n^{-1}} \right.$$
$$\left. + \frac{1}{2}\|\mathbf{x}_0 - \mathbf{m}_0\|^2_{\mathbf{\Sigma}_0^{-1}} + \sum_{n=1}^{N} [\lambda_x\|\mathbf{o}_{x,n}\|_1 + \lambda_y\|\mathbf{o}_{y,n}\|_1] \right\} \tag{3.6}$$

where λ_x and λ_y are introduced in (3.6) to allow individual control of sparsity levels in $\mathbf{o}_{x,n}$ and $\mathbf{o}_{y,n}$. Viewing the cost in (3.6) as a Lagrangian function, allows casting this *unconstrained* minimization problem as a constrained one. Indeed, sufficiency of the Lagrange multiplier theory implies that [12, Sec. 3.3.4]: using the solution $\widehat{\mathbf{o}}_x, \widehat{\mathbf{o}}_y$ of (3.6) for given multipliers λ_x, $\lambda_y \geq 0$ and letting $\tau_x := \|\widehat{\mathbf{o}}_x\|_1$, $\tau_y := \|\widehat{\mathbf{o}}_y\|_1$, the equivalent *constrained* minimization problem entails the WLS cost (quadratic terms in (3.6)) subject to the constraints $\|\mathbf{o}_x\|_1 \leq \tau_x$, and $\|\mathbf{o}_y\|_1 \leq \tau_y$. Note however, that $\lambda_x(\lambda_y)$ in (3.6), and likewise $\tau_x(\tau_y)$ in its constrained equivalent, are tuning parameters and not optimization variables.

The DRS state estimate in (3.6) can cope with outliers *jointly* present in the state and

in the measurements. In addition, it is *universal* because it does not require knowing the distribution of the nominal noise or the outlier vectors. (The choice of λ_x and λ_y discussed in the next section will not follow from the distribution of a contaminating model but will be data driven.) Different from [4] and [114] which enforce sparsity in the state, DRS controls sparsity in the outliers to effect robustness. At this point, it is worth recalling that $\mathbf{o}$ in smoothing dynamical processes is indeed sparse, since it models abrupt changes (target maneuvers) in the state which cannot be too many in the analysis window, and glint noise giving rise to large-magnitude observations which occur rarely too. Having explained why it is meaningful to expect only few nonzero entries in $\mathbf{o}$, it is also useful to clarify that this is not necessary. (Simulated tests in Section 3.7 will allow for outlier contamination as high as 80%.) Although smoothing performance degrades as the number of nonzero entries in $\mathbf{o}$ increases, all the proposed approach needs is a handle on the percentage of outliers without requiring this percentage to be necessarily low.

The WLS cost can be also replaced by other functions (e.g., the ℓ_1-norm of the error), and alternative regularization terms (e.g., the ℓ_2-norm of the outliers) can be employed instead of, or, in addition to the ℓ_1-norm [56]. Non-convex costs and regularizers are also possible, but they are not recommended as stand-alone solvers of (3.6) because they cannot guarantee convergence to the global optimum. In contrast, it will be seen in Sections 3.4 and 3.5 that (3.6) and variants involving ℓ_1 and ℓ_2 norms can afford not only globally convergent but also computationally efficient solvers.

Having clarified that the ensuing developments will rely on (3.6), which is meaningful regardless of the $\{\mathbf{w}, \mathbf{v}, \mathbf{o}\}$ distributions, it is natural to ask the following question: Under what assumptions on these distributions can one claim MAP optimality of the resultant state and outlier estimators? The ensuing proposition (proved in Appendix 3.9.1) asserts that this is possible if the nominal noise vectors are Gaussian and the additive outliers are known to be Laplacian distributed.

Proposition 4. *Suppose* $\mathbf{w}_n$ *and* $\mathbf{v}_n$ *are Gaussian distributed, mutually independent, and independent from* $\mathbf{o}_{x,n}$ *and* $\mathbf{o}_{y,n}$*, respectively. Furthermore, assume* $\mathbf{o}_{x,n}$ *has Laplacian distributed entries* $o_{x,n,d}$ *that are independent from past states, past state outliers, measurement outliers, and across different dimensions; that is,* $o_{x,n,d}$ *and* $o_{x,n,d'}$ *are independent for* $d \neq d'$*. Similarly,* $\mathbf{o}_{y,n}$ *has Laplacian distributed entries* $o_{y,n,d}$ *that are independent from past states, past measurement outliers, state outliers, and across different dimensions; then the estimators obtained as in (3.6) are MAP optimal.*

Albeit simple to prove, the usefulness of Proposition 4 is twofold: (a) it allows for a side-by-side comparison with the MAP optimality offered by the clairvoyant KS in (3.4); and (b) it positions the proposed approach in the context of related MAP-optimal schemes adopting ℓ_1-error based smoothers; see e.g., [5] and references therein. Specifically, different from the multivariate Laplacian in Proposition 4 described by the two scalar λ_x and λ_y parameters, the ℓ_1-Laplacian model in [5] entails a $D_y \times D_y$ matrix of parameters that are assumed known.

Next, robustness of the estimators (3.6) is established. Specifically, the ensuing proposition proved in Appendix 3.9.2, shows that DRS subsumes Huber's M-estimator as a special case.

Proposition 5. *When* $\{\mathbf{Q}_n, \mathbf{R}_n\}_{n=1}^N$ *and* $\mathbf{\Sigma}_0$ *are all identity matrices, the DRS in (3.6) boils down to solving the following Huber M-estimator problem*

$$\widehat{\mathbf{x}} := \arg\min_{\mathbf{x}} \left\{ \sum_{d=1}^{D_y} \sum_{n=1}^{N} \rho_{\lambda_y}(y_{n,d} - \mathbf{h}_{n,d}^T \mathbf{x}_n) + \sum_{d'=1}^{D_x} \left[(x_{0,d'} - m_{0,d'})^2 + \sum_{n=1}^{N} \rho_{\lambda_x}(x_{n,d'} - \mathbf{f}_{n,d'}^T \mathbf{x}_{n-1}) \right] \right\} \tag{3.7}$$

where $\mathbf{y}_n := [y_{n,1}, y_{n,2}, \ldots, y_{n,D_y}]^T$, $\mathbf{x}_n := [x_{n,1}, x_{n,2}, \ldots, x_{n,D_x}]^T$, $\mathbf{x} := [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_N^T]^T$,

$\mathbf{H}_n := [\mathbf{h}_{n,1}, \ldots, \mathbf{h}_{n,D_y}]^T$, $\mathbf{F}_n := [\mathbf{f}_{n,1}, \ldots, \mathbf{f}_{n,D_x}]^T$, *and* ρ_λ *denotes the Huber cost [67]*

$$\rho_\lambda(r) := \begin{cases} \frac{1}{2}r^2, & \text{if } |r| \leq \lambda \\ \lambda|r| - \frac{\lambda^2}{2}, & \text{otherwise.} \end{cases}$$

Proposition 5 generalizes to dynamical systems the link between (3.6) and (3.7) established in [53] for linear regression models. As a result, DRS also inherits the robustness attributes associated with the Huber M-estimators. Figure 3.1 depicts Huber's cost along with the quadratic one. For small residuals (i.e., $|r| \leq \lambda$), $\rho_\lambda(r)$ coincides with the quadratic one. But for $|r| > \lambda$, Huber's cost grows only linearly with r, which allows for down-weighting large errors. Therefore, outliers which are responsible for large errors will be weighted less in the overall objective function. Clearly, for large values of λ, Huber's cost coincides with the quadratic one. As a consequence, a large number of outliers in the observations and state is effected through small λ_y and λ_x, respectively. Finally, it should be mentioned that the Huber function is not the only one enabling robustness. A gamut of related robust costs can be found in e.g., [62, Appendix A6.8] with different properties. The most convincing reason for exploiting sparsity constraints under the ℓ_1-norm of the outlier vectors is to leverage recent advances on compressive sampling to develop the computationally efficient and globally convergent solvers presented in Section 3.4.

While DRS inherits the robustness features of Huber's M-estimator, it enjoys several advantages over it, as detailed in the following two remarks.

Remark 4. As mentioned earlier, the universality of DRS pertains also to the regularization term. If outliers are present in *all* entries of $\mathbf{x}_n$ or $\mathbf{y}_n$, this form of *group sparsity* can be effected by replacing $\|\mathbf{o}_{y,n}\|_1$ and $\|\mathbf{o}_{x,n}\|_1$ in (3.6) with $\|\mathbf{o}_{y,n}\|_2$ and $\|\mathbf{o}_{x,n}\|_2$, respectively. Using the latter regularization, either $\mathbf{o}_{y,n} = \mathbf{0}_{D_y}$ ($\mathbf{o}_{x,n} = \mathbf{0}_{D_x}$), or *all* the entries of $\mathbf{o}_{y,n}$ ($\mathbf{o}_{x,n}$) are nonzero, signifying the presence of outliers in *all* measurement (state) variables at time n. The cost function resulting from ℓ_2-norm regularization is still convex [120], and

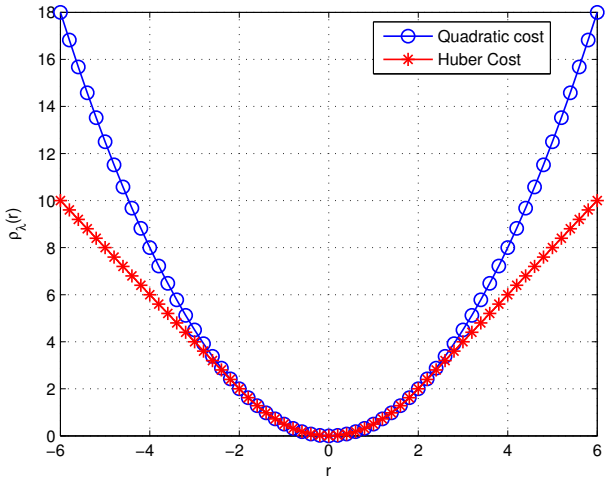

Figure 3.1: Quadratic cost versus Huber cost ($\lambda = 2$).

its minimization can be carried out using solvers similar to those of (3.6) to be presented in Sections 3.4 and 3.5. For this reason, only ℓ_1-norm regularization will be considered henceforth.

Remark 5. In addition to universal robustness, the novel approach to DRS is also flexible in three counts. First, Huber estimators fix λ_x or λ_y *a fortiori* based on knowledge of the nominal distribution and the contamination model, e.g., for the ϵ-contaminated class with Gaussian nominal, it follows that $\lambda_x = \lambda_y = 1.345$ [67]. In contrast, DRS does not assume any specific model for the outliers' distribution. From this viewpoint, M-estimators are subsumed by the present formulation as special cases corresponding to specific values of λ_x and λ_y. In addition, DRS can accommodate colored noise [cf. (3.6)], which is formidable for the robust estimators of [84] and [107] because pre-whitening in (3.7) with $\mathbf{Q}_n^{-0.5}$ and $\mathbf{R}_n^{-0.5}$ spreads the outliers to non-contaminated measurements. Finally, DRS not only allows one to apply KS on outlier-free data but also reveals the outliers – a feature not available to Huber-based approaches, which only implicitly incorporate the outliers.

The next section presents systematic means of adjusting λ_x and λ_y to accommodate fully nominal settings (i.e., no outliers), fully contaminated scenarios, and all cases in between, even when the degree of contamination is unknown.

3.3 Selecting λ_x and λ_y

Parameters λ_x and λ_y control the level of sparsity in the estimated outlier vectors, and their judicious selection is crucial for the successful operation of DRS. Too large a value for these parameters reverts DRS back to the KS, which is non-robust. On the other hand, very small values give rise to many spurious state and measurement outliers, thus degrading DRS performance. Standard cross-validation techniques [99], are not effective when outliers are present [74]. Toward choosing proper values of λ_x and λ_y, the next proposition provides computable bounds so that if $\lambda_y \geq \bar{\lambda}_y$ and $\lambda_x \geq \bar{\lambda}_x$, then DRS coincides with KS. (See Appendix 3.9.3 for the proof.)

Proposition 6. *The DRS estimate in (3.6) coincides with KS estimate* $\widehat{\mathbf{x}}^{\mathrm{KS}}$ *if*

$$\lambda_y \geq \bar{\lambda}_y := \max_{1\leq n\leq N} \left\|\mathbf{R}_n^{-1}(\mathbf{y}_n - \mathbf{H}_n\widehat{\mathbf{x}}_n^{\mathrm{KS}})\right\|_\infty \tag{3.8a}$$

$$\lambda_x \geq \bar{\lambda}_x := \max_{1\leq n\leq N} \left\|\mathbf{Q}_n^{-1}(\widehat{\mathbf{x}}_n^{\mathrm{KS}} - \mathbf{F}_n\widehat{\mathbf{x}}_{n-1}^{\mathrm{KS}})\right\|_\infty . \tag{3.8b}$$

Having established the upper bounds in (3.8), desirable values for λ_x and λ_y will be points in the rectangle $[0, \bar{\lambda}_x] \times [0, \bar{\lambda}_y]$. Consider a two-dimensional grid on this rectangle and a properly chosen cost generated by each grid point. Depending on the information available to the designer, the "best" λ_x and λ_y will be those values minimizing either one of two costs presented in the ensuing subsections.

3.3.1 Known percentage of outliers

Here the percentage of (non-)zero entries of the outlier vectors is assumed (at least approximately) known; denote them as $\pi_{o,x}$ and $\pi_{o,y}$. Consider the 2-D grid on $[0, \bar{\lambda}_x] \times [0, \bar{\lambda}_y]$, comprising I_x points along the λ_x axis and I_y points along the λ_y axis. Let (i_x, i_y), with $1 \leq i_x \leq I_x$ and $1 \leq i_y \leq I_y$ be the grid point corresponding to values $\lambda_x(i_x)$ and $\lambda_y(i_y)$. For the given (i_x, i_y), solve (3.6) with $\lambda_x = \lambda_x(i_x)$ and $\lambda_y = \lambda_y(i_y)$, to obtain $\widehat{\mathbf{x}}(i_x, i_y), \widehat{\mathbf{o}}_x(i_x, i_y)$ and $\widehat{\mathbf{o}}_y(i_x, i_y)$. With $\mathrm{supp}(\mathbf{x})$ representing the non-zero entries of $\mathbf{x}$, and $|\mathbf{x}|$ the number of entries of $\mathbf{x}$, the "best" $\lambda_x(i_x^*)$ and $\lambda_y(i_y^*)$ are found as those with

$$[i_x^*, i_y^*] \;:=\; \arg\min_{i_x, i_y} \left\{ \left| \pi_{x,o} - \frac{|\mathrm{supp}(\widehat{\mathbf{o}}_x(i_x, i_y))|}{|\widehat{\mathbf{o}}_x(i_x, i_y)|} \right| + \left| \pi_{y,o} - \frac{|\mathrm{supp}(\widehat{\mathbf{o}}_y(i_x, i_y))|}{|\widehat{\mathbf{o}}_y(i_x, i_y)|} \right| \right\} . \quad (3.9)$$

Finding $\lambda_x(i_x^*)$ and $\lambda_y(i_y^*)$ as in (3.9), appears to require solving (3.6) for all pairs (i_x, i_y) of the two-dimensional grid. The associated computational cost can be viewed as the "price paid" for the universality attribute of DRS elaborated in Section 3.2. Instead of two, (λ_x, λ_y), recall that the number of parameters (and thus dimensionality of the search space had those been unknown) in [5] is $\mathcal{O}(D_y^2)$. Of course, this is not an issue in [5] where these parameters are assumed known.

Fortunately, the special structure of the optimization problem in (3.6) allows for solvers at complexity lower than running $I_x I_y$ robust smoothers, one per (λ_x, λ_y) point on the grid. Indeed, (3.6) can be formulated as a quadratic program (QP), and its form can leverage recent advances in computing the so-termed least-absolute shrinkage and selection operator (Lasso), originally developed for static linear regressions; see e.g., [63]. As will be detailed in Section 3.4, Lasso can be also exploited for the DRS dynamical model considered here. General-purpose QP solvers incur polynomial complexity up to $\mathcal{O}(D^{3.5})$ per iteration, where D is the number of optimization variables involved [19]; here, $D = N(2D_x + D_y + 1)$. The reduction to $\mathcal{O}(D)$ per iteration afforded by Lasso-based solvers becomes possible by starting from $\lambda := (\bar{\lambda}_x, \bar{\lambda}_y)$ (sparsest initialization) and solving successively over decreasing

λ-points on the grid, using coordinate descent iterations. Qualitatively speaking, about one nonzero entry of $\mathbf{o}$ emerges per λ-point on the grid, and its value is used to initialize the iteration for the next point on the grid (warm start) [51, 116]. Especially for large problem dimensions ($D \gg$), it has been demonstrated that such Lasso solutions for the entire so-termed *regularization path* (corresponding to all λ-points on the grid), can be more computationally efficient than solving Lasso even for a single fixed point λ on the grid; see also [52].

3.3.2 Known covariance of nominal noise vectors

The key observation here is that if the estimates $\widehat{\mathbf{o}}_{x,n}(i_x, i_y)$ and $\widehat{\mathbf{o}}_{y,n}(i_x, i_y)$ are accurate, then $\widehat{\mathbf{x}}_n(i_x, i_y) - \mathbf{F}_n\widehat{\mathbf{x}}_{n-1}(i_x, i_y) - \widehat{\mathbf{o}}_{x,n}(i_x, i_y)$ should have the same statistics as $\mathbf{w}_n$; and likewise, the statistics of $\mathbf{y}_n - \mathbf{H}_n\widehat{\mathbf{x}}_n(i_x, i_y) - \widehat{\mathbf{o}}_{y,n}(i_x, i_y)$ should coincide with those of $\mathbf{v}_n$. Focusing for instance on second-order statistics, if these estimated residuals are pre-whitened (by left-multiplication with $\mathbf{Q}_n^{-0.5}$ and $\mathbf{R}_n^{-0.5}$), they should have zero mean and unit variance. Thus, upon pre-whitening and averaging, their sample variance should approach 1. As a consequence, the "best" $\lambda_x(i_x^*)$ and $\lambda_y(i_y^*)$ are found as those with

$$\begin{aligned} [i_x^*, i_y^*] &:= \arg\min_{i_x, i_y} \left|1 - \hat{\sigma}_e^2(i_x, i_y)\right| \qquad (3.10) \\ \hat{\sigma}_e^2(i_x, i_y) &:= \frac{\|\widehat{\mathbf{w}}_0(i_x, i_y)\|^2_{\mathbf{\Sigma}_0^{-1}} + \sum_{n=1}^{N}\left[\|\widehat{\mathbf{v}}_n(i_x, i_y)\|^2_{\mathbf{R}_n^{-1}} + \|\widehat{\mathbf{w}}_n(i_x, i_y)\|^2_{\mathbf{Q}_n^{-1}}\right]}{ND_y + (N+1)D_x} \end{aligned}$$

where

$$\begin{aligned} \widehat{\mathbf{v}}_n(i_x, i_y) &:= \mathbf{y}_n - \mathbf{H}_n\widehat{\mathbf{x}}_n(i_x, i_y) - \widehat{\mathbf{o}}_{y,n}(i_x, i_y) \\ \widehat{\mathbf{w}}_n(i_x, i_y) &:= \widehat{\mathbf{x}}_n(i_x, i_y) - \mathbf{F}_n\widehat{\mathbf{x}}_{n-1}(i_x, i_y) - \widehat{\mathbf{o}}_{x,n}(i_x, i_y) \\ \widehat{\mathbf{w}}_0(i_x, i_y) &:= \widehat{\mathbf{x}}_0(i_x, i_y) - \mathbf{m}_0. \end{aligned}$$

The number of grid points I_x and I_y should be chosen large enough to ensure that a point in the vicinity of the global minimum of (3.10) is obtained. The grid need not be uniform. Indeed, simulations confirm that the search is more efficient if grid points are chosen on the log scale; see also [52]. This parameter tuning method, will be henceforth referred to as absolute variance deviation (AVD). Since DRS in (3.6) requires knowledge of nominal noise covariances, the AVD scheme needs no additional assumption; and similar to the method of the previous subsection, it can capitalize on Lasso coordinate descent based schemes to lower the computational complexity of solving (3.6) per grid point, as detailed next.

3.4 DRS via coordinate descent

While general purpose QP solvers can be utilized to solve (3.6) with polynomial complexity in N, their complexity can still be too high when N is large. A reduced-complexity alternative is developed in this section to solve (3.6) using block coordinate descent iterations. Letting $C(\mathbf{x}, \mathbf{o}_x, \mathbf{o}_y)$ denote the cost in (3.6) and j indexing coordinate descent iterations, the following sub-problems are solved per iteration j and coordinate dimension d

$$\mathbf{x}^{(j)} = \arg\min_{\mathbf{x}} C(\mathbf{x}, \mathbf{o}_x^{(j-1)}, \mathbf{o}_y^{(j-1)}) \tag{3.11a}$$

$$o_{x,n,d}^{(j)} = \arg\min_{o_{x,n,d}} C(\mathbf{x}^{(j)}, \mathbf{o}_{x,1}^{(j)}, \ldots, \mathbf{o}_{x,n-1}^{(j)}, \mathbf{o}_{x,n,1:d-1}^{(j)}, o_{x,n,d}, \mathbf{o}_{x,n,d+1:D_x}^{(j-1)}, \mathbf{o}_{x,n+1}^{(j-1)}, \ldots, \mathbf{o}_{x,N}^{(j-1)}, \mathbf{o}_y^{(j-1)}) \tag{3.11b}$$

$$o_{y,n,d}^{(j)} = \arg\min_{o_{y,n,d}} C(\mathbf{x}^{(j)}, \mathbf{o}_x^{(j)}, \mathbf{o}_{y,1}^{(j)}, \ldots, \mathbf{o}_{y,n-1}^{(j)}, \mathbf{o}_{y,n,1:d-1}^{(j)}, o_{y,n,d}, \mathbf{o}_{y,n,d+1:D_y}^{(j-1)}, \mathbf{o}_{y,n+1}^{(j-1)}, \ldots, \mathbf{o}_{y,N}^{(j-1)}) \tag{3.11c}$$

where (3.11b) is solved for $n = 1, \ldots, N$ and $d = 1, \ldots, D_x$, while (3.11c) is solved for $n = 1, \ldots, N$ and $d = 1, \ldots, D_y$. The initial conditions are $\mathbf{o}_x^{(0)} = \mathbf{0}_{ND_x}$ and $\mathbf{o}_y^{(0)} = \mathbf{0}_{ND_y}$.

The optimization in (3.11a) can be explicitly written as

$$\mathbf{x}^{(j)} := \arg\min_{\mathbf{x}} \left\{ \frac{1}{2}\sum_{n=1}^{N} \left\| \mathbf{y}_n - \mathbf{H}_n\mathbf{x}_n - \mathbf{o}_{y,n}^{(j-1)} \right\|_{\mathbf{R}_n^{-1}}^2 + \frac{1}{2}\sum_{n=1}^{N} \left\| \mathbf{x}_n - \mathbf{F}_n\mathbf{x}_{n-1} - \mathbf{o}_{x,n}^{(j-1)} \right\|_{\mathbf{Q}_n^{-1}}^2 + \frac{1}{2}\|\mathbf{x}_0 - \mathbf{m}_0\|_{\mathbf{\Sigma}_0^{-1}}^2 \right\}. \quad (3.12)$$

Solving (3.12) is equivalent to finding the KS estimate for a system with outlier-compensated measurements $\mathbf{y}_n - \mathbf{o}_{y,n}^{(j-1)}$, and outlier-compensated state $\mathbf{x}_n - \mathbf{o}_{x,n}^{(j-1)}$. Therefore, either the batch or the forward-backward recursive algorithms reviewed in Section 3.1 can be adopted to solve (3.12) with linear complexity in N.

Focusing on (3.11b), one should solve

$$o_{x,n,d}^{(j)} := \arg\min_{o_{x,n,d}} \frac{1}{2} \left\| \mathbf{x}_n^{(j)} - \mathbf{F}_n\mathbf{x}_{n-1}^{(j)} - \begin{bmatrix} \mathbf{o}_{x,n,1:d-1}^{(j)} \\ o_{x,n,d} \\ \mathbf{o}_{x,n,d+1:D_x}^{(j-1)} \end{bmatrix} \right\|_{\mathbf{Q}_n^{-1}}^2 + \lambda_x |o_{x,n,d}| \quad (3.13)$$

for every $d = 1, \ldots, D_x$ and $n = 1, \ldots, N$. The scalar problem (3.13) is solved using the Lasso, which can afford a closed-form solution [63]. Indeed, (3.13) can be equivalently expressed as

$$o_{x,n,d}^{(j)} := \arg\min_{o_{x,n,d}} \frac{1}{2}\left(o_{x,n,d} - \gamma_{x,n,d}^{(j)}\right)^2 + \lambda_{x,n,d}|o_{x,n,d}| \quad (3.14)$$

where

$$\gamma_{x,n,d}^{(j)} := \frac{1}{q_{n,d,d}}\left[\alpha_{x,n,d}^{(j)} - \sum_{k=1}^{d-1} q_{n,k,d}o_{x,n,k}^{(j)} - \sum_{k=d+1}^{D_x} q_{n,k,d}o_{x,n,k}^{(j-1)}\right]$$

$$\boldsymbol{\alpha}_{x,n}^{(j)} := \mathbf{Q}_n^{-1}\left(\mathbf{x}_n^{(j)} - \mathbf{F}_n\mathbf{x}_{n-1}^{(j)}\right), \qquad \lambda_{x,n,d} := \lambda_x/q_{n,d,d}$$

and $\mathbf{Q}_n^{-1}$ has entries $[\mathbf{Q}_n^{-1}]_{k,k'} := q_{n,k,k'}$. The solution to (3.14) is given by

$$o_{x,n,d}^{(j)} = \left[\left|\gamma_{x,n,d}^{(j)}\right| - \lambda_{x,n,d}\right]^+ \text{ sign } \left(\gamma_{x,n,d}^{(j)}\right)$$

where $[x]^+ := \max(x, 0)$ and $\text{sign}(\cdot)$ denotes the sign operator.

A similar closed-form solution becomes available for (3.11c), since

$$o_{y,n,d}^{(j)} := \arg\min_{o_{y,n,d}} \frac{1}{2} \left\| \mathbf{y}_n - \mathbf{H}_n \mathbf{x}_n^{(j)} - \begin{bmatrix} \mathbf{o}_{y,n,1:d-1}^{(j)} \\ o_{y,n,d} \\ \mathbf{o}_{y,n,d+1:D_y}^{(j-1)} \end{bmatrix} \right\|_{\mathbf{R}_n^{-1}}^2 + \lambda_y |o_{y,n,d}|. \tag{3.15}$$

for every $d = 1, \ldots, D_y$ and $n = 1, \ldots, N$.

Problem (3.15) can be alternatively written as

$$o_{y,n,d}^{(j)} := \arg\min_{o_{y,n,d}} \frac{1}{2} \left(o_{y,n,d} - \gamma_{y,n,d}^{(j)} \right)^2 + \lambda_{y,n,d} |o_{y,n,d}| \tag{3.16}$$

where

$$\begin{aligned} \gamma_{y,n,d}^{(j)} &:= \frac{1}{r_{n,d,d}} [\alpha_{y,n,d}^{(j)} - \sum_{k=1}^{d-1} r_{n,k,d} o_{y,n,k}^{(j)} - \sum_{k=d+1}^{D_y} r_{n,k,d} o_{y,n,k}^{(j-1)}] \\ \boldsymbol{\alpha}_{y,n}^{(j)} &:= \mathbf{R}_n^{-1} \left(\mathbf{y}_n - \mathbf{H}_n \mathbf{x}_n^{(j)} \right), \qquad \lambda_{y,n,d} := \lambda_y / r_{n,d,d} \end{aligned}$$

and $\mathbf{R}_n^{-1}$ has entries $[\mathbf{R}_n^{-1}]_{k,k'} := r_{n,k,k'}$. The solution to (3.14) is given by

$$o_{y,n,d}^{(j)} = \left[\left| \gamma_{y,n,d}^{(j)} \right| - \lambda_{y,n,d} \right]^+ \text{ sign } \left(\gamma_{y,n,d}^{(j)} \right).$$

Global convergence of the (3.12)-(3.15) iterates is guaranteed from the results in [113], as summarized next.

Proposition 7. *For any initial values $\mathbf{x}^{(0)}, \mathbf{o}_x^{(0)}, \mathbf{o}_y^{(0)}$, the iterates in (3.12), (3.13) and (3.15) are all convergent. Furthermore, every limit point of the sequences $\mathbf{x}^{(j)}$, $\mathbf{o}_x^{(j)}$, $\mathbf{o}_y^{(j)}$ solves (3.6).*

Note that (3.12) contains the bulk of computation per iteration j, and its complexity is equivalent to that of KS, which is linear in N. This should be contrasted with the

general purpose convex solvers whose complexity is polynomial in N (worst-case of order $\mathcal{O}(N^{3.5})$; see e.g., [19]). As mentioned earlier, the complexity reduction is due to the unique properties of Lasso-related problems, namely variable separability, closed-form thresholding per variable, and warm starts. Coordinate descent solvers capitalize on these properties, and have been documented to outperform competing alternatives, including off-the-shelf QP solvers [51, 52, 116].

Remark 6. This section's efficient solvers of the ℓ_1-norm based convex cost in (3.6) will converge to estimates generally not coinciding with the global optimum of the ultimate ℓ_0-norm based sparsity-promoting cost in (3.5). This motivates *concave* regularization terms, which offer improved approximations of the ℓ_0-norm relative to that offered by the ℓ_1-norm [74]. One such alternative leads to solving

$$[\hat{\mathbf{x}},\hat{\mathbf{o}}] := \arg\min_{\mathbf{x},\mathbf{o}} \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{o}\|^2_{\mathbf{Q}_w^{-1}}+\lambda_x\sum_{n=1}^{N}\sum_{d=1}^{D_x}\log(|o_{x,n,d}|+\delta_x)+\lambda_y\sum_{n=1}^{N}\sum_{d=1}^{D_y}\log(|o_{y,n,d}|+\delta_y) \tag{3.17}$$

where δ_x (δ_y) are small positive constants to ensure that the argument of the logarithm stays away from zero. Since the cost in (3.17) is non-convex, it is recommended to initialize its iterative minimization with the efficient convex solver of (3.6). Starting with such an initialization $(\mathbf{x}^{(0)},\mathbf{o}^{(0)})$, the logarithm can be successively linearized around the l-th iterate using $\log(t+\delta)\approx\log(t^{(l)}+\delta)+(t-t^{(l)})/(t^{(l)}+\delta)$ to arrive at a convex cost, which can be readily optimized to obtain the estimates at iteration $(l+1)$. Specifically, at iteration l one solves

$$[\mathbf{x}^{(l)},\mathbf{o}^{(l)}] := \arg\min_{\mathbf{x},\mathbf{o}} \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}-\mathbf{o}\|^2_{\mathbf{Q}_w^{-1}}+\lambda_x\sum_{n=1}^{N}\sum_{d=1}^{D_x}w^{(l)}_{x,n,d}|o_{x,n,d}|+\lambda_y\sum_{n=1}^{N}\sum_{d=1}^{D_y}w^{(l)}_{y,n,d}|o_{y,n,d}| \tag{3.18}$$

where

$$w^{(l)}_{x,n,d} := \left(|o^{(l-1)}_{x,n,d}|+\delta_x\right)^{-1}, \quad w^{(l)}_{y,n,d} := \left(|o^{(l-1)}_{y,n,d}|+\delta_y\right)^{-1}.$$

Note that (3.18) is similar to the DRS one in (3.6) except that the entries of vector $\mathbf{o}$ in the regularization are weighted non-uniformly. Being convex, (3.18) can be solved as easily as (3.6). With reliable initialization offered by the solution of (3.6), one reason behind the enhancement offered by (3.18) is the bias correction to Lasso, which is known to yield reliable estimates of the $\mathbf{o}$ support but biased estimates of its nonzero entries [63]. Besides (3.18), alternative means to mitigate such bias effects is to retain only the outlier-free measurements, after identifying them through the zero entries of $\mathbf{o}$, and use them to re-run the clairvoyant KS which is unbiased. The improvement offered by these refined estimates and those obtained by solving (3.18) will be corroborated via simulated tests.

3.5 Fixed-lag DRS for online operation

The major limitation of *fixed-interval* smoothing is that the whole batch $\{\mathbf{y}_n\}_{n=1}^N$ has to be available prior to estimating $\{\mathbf{x}_n\}_{n=1}^N$. This is useful for applications such as processing electroencephalograms [108], but not for target tracking. In many tracking applications, state smoothing has to be performed online and stringent delay ("lag") constraints are imposed between the smoothed state instant and the time state estimates are formed. Online smoothing is also important in applications where the state is affected by abrupt changes since these events may be the manifestation of, e.g., system failures [93].

When the outliers are absent in (3.1), optimal fixed-lag KS can be regarded as a special case of fixed-interval KS [2, p. 176]. The goal here is to estimate $\mathbf{x}_n$, relying upon observations up to time $n+\ell$, where ℓ denotes the estimation lag. Supposing that a KF has been run up to time n to yield the state and covariance estimates $\mathbf{x}_{n|n}$ and $\mathbf{\Sigma}_{n|n}$, fixed-lag KS can be accomplished using

$$\widehat{\mathbf{x}}_{n:n+\ell}^{\mathrm{KS}} = \arg\min_{\mathbf{x}} \quad \left\{ \frac{1}{2} \sum_{n'=n+1}^{n+\ell} \|\mathbf{y}_{n'} - \mathbf{H}_{n'}\mathbf{x}_{n'}\|_{\mathbf{R}_{n'}^{-1}}^2 + \frac{1}{2}\|\mathbf{x}_n - \mathbf{x}_{n|n}\|_{\mathbf{\Sigma}_{n|n}^{-1}}^2 \right.$$

$$+\frac{1}{2}\sum_{n'=n+1}^{n+\ell}\|\mathbf{x}_{n'}-\mathbf{F}_{n'}\mathbf{x}_{n'-1}\|^2_{\mathbf{Q}_{n'}^{-1}}\Bigg\} \tag{3.19}$$

where $\widehat{\mathbf{x}}^{\mathrm{KS}}_{n:n+\ell} := [(\widehat{\mathbf{x}}^{\mathrm{KS}}_{n})^T,\ldots,(\widehat{\mathbf{x}}^{\mathrm{KS}}_{n+\ell})^T]^T$. Observe that fixed-lag KS in (3.19) is a special case of fixed-interval KS, when the initial condition on the state, namely $\mathbf{x}_{n|n}$ and $\mathbf{\Sigma}_{n|n}$, is given by the KF, and the state is smoothed over the interval $[n, n+\ell]$. Thus, the solution of (3.19) can be found with either one of the two algorithms of Section 3.1. However, since (3.19) does not account for outliers, the resulting estimator is not robust. To address this issue, a fixed-lag DRS is developed next.

3.5.1 Fixed-lag DRS

In the previous section, the fixed-lag KS was regarded as a special case of the fixed-interval KS with properly chosen smoothing interval and initial conditions. Furthermore, in Section 3.2 a fixed-interval DRS was developed, which is extended here to robustify the fixed-lag KS in (3.19). Mimicking the steps followed in Section 3.2 to robustify (3.19) is challenged by the fact that the initial conditions $\mathbf{x}_{n|n}$ and $\mathbf{\Sigma}_{n|n}$ are evaluated by the clairvoyant (and thus non-robust) KF. To overcome this obstacle, the fixed-lag KS will be recast in a form entailing the interval $[n-w, n+\ell]$; that is,

$$\check{\mathbf{x}}^{\mathrm{KS}}_{n-w:n+\ell}=\arg\min_{\mathbf{x}} \quad \Bigg\{\frac{1}{2}\sum_{n'=n-w+1}^{n+\ell}\|\mathbf{y}_{n'}-\mathbf{H}_{n'}\mathbf{x}_{n'}\|^2_{\mathbf{R}_{n'}^{-1}}+\frac{1}{2}\|\mathbf{x}_{n-w}-\mathbf{x}_{n-w|n-w}\|^2_{\mathbf{\Sigma}^{-1}_{n-w|n-w}}$$
$$+\frac{1}{2}\sum_{n'=n-w+1}^{n+\ell}\|\mathbf{x}_{n'}-\mathbf{F}_{n'}\mathbf{x}_{n'-1}\|^2_{\mathbf{Q}_{n'}^{-1}}\Bigg\}. \tag{3.20}$$

The formulation in (3.20) is equivalent to that of (3.19) in the sense that $\check{\mathbf{x}}^{\mathrm{KS}}_{n}=\widehat{\mathbf{x}}^{\mathrm{KS}}_{n}$. It also suggests that the fixed-lag KS estimate at time n and lag ℓ can also be obtained by initializing its recursions with the KF estimates $\mathbf{x}_{n-w|n-w}$ and $\mathbf{\Sigma}_{n-w|n-w}$ for arbitrary w. The fixed-lag DRS is obtained by robustifying the fixed-lag KS (3.20) in a fashion similar

to that used in Section 3.2; that is,

$$[\check{\mathbf{x}}^{\mathrm{DRS}}_{n-w:n+\ell}, \check{\mathbf{o}}_{x,n-w:n+\ell}, \check{\mathbf{o}}_{y,n-w:n+\ell}] = \\ \arg\min_{\mathbf{x},\mathbf{o}_x,\mathbf{o}_y} \left\{ \frac{1}{2} \sum_{n'=n-w+1}^{n+\ell} \|\mathbf{y}_{n'} - \mathbf{H}_{n'}\mathbf{x}_{n'} - \mathbf{o}_{y,n'}\|^2_{\mathbf{R}^{-1}_{n'}} + \frac{1}{2}\|\mathbf{x}_{n-w} - \mathbf{x}_{n-w|n-w}\|^2_{\mathbf{\Sigma}^{-1}_{n-w|n-w}} \right. \\ \left. + \frac{1}{2} \sum_{n'=n-w+1}^{n+\ell} \|\mathbf{x}_{n'} - \mathbf{F}_{n'}\mathbf{x}_{n'-1} - \mathbf{o}_{x,n'}\|^2_{\mathbf{Q}^{-1}_{n'}} + \sum_{n'=n-w+1}^{n+\ell} [\lambda_y \|\mathbf{o}_{y,n'}\|_1 + \lambda_x \|\mathbf{o}_{x,n'}\|_1] \right\}. \tag{3.21}$$

Observe that eventual errors in $\mathbf{x}_{n-w|n-w}$ and $\mathbf{\Sigma}_{n-w|n-w}$ due to the non-robust KF do not severely affect the estimates at time n provided that w is sufficiently large. Certainly, the larger the w, the larger the number of nuisance variables involved.

The major limitation of the fixed-lag DRS in (3.21) is that a convex optimization problem has to be solved at each time n to obtain $\check{\mathbf{x}}^{\mathrm{DRS}}_n$. As a consequence, the associated computational burden to solve the fixed-lag DRS in (3.21) is not comparable with that of the standard fixed-lag KS. This motivates approximating the fixed-lag approach in (3.21) to enable online DRS at complexity comparable to that of standard fixed-lag KS and state-of-the-art non-linear smoothers.

3.5.2 Online fixed-lag DRS

The coordinate descent-based fixed-interval algorithm in Section 3.4 is properly modified in this section in order to solve the fixed-lag DRS problem formulated in (3.21). Despite the fact that convergence to a solution of (3.21) is provably guaranteed asymptotically (i.e., for infinite iterations), satisfactory estimates can be obtained with only a few coordinate descent iterations.

Suppose that between two consecutive observations (say $n+\ell$ and $n+\ell+1$), the affordable delay allows for J coordinate descent iterations to be implemented. Furthermore,

for a limited number of iterations, initializing with $\mathbf{o}_{x,n-w:n+\ell}^{(0)}$ and $\mathbf{o}_{y,n-w:n+\ell}^{(0)}$ close to their global optimum values provides a "warm start-up" considerably improving the performance. Observe that for estimating the state at time n, fixed-lag DRS entails smoothing over the interval $[n-w, n+\ell]$, and after J coordinate descent iterations, the variables $\mathbf{x}_{n-w:n+\ell}^{(J)}$, $\mathbf{o}_{x,n-w:n+\ell}^{(J)}$, $\mathbf{o}_{y,n-w:n+\ell}^{(J)}$ become available. Since fixed-lag DRS at time $n+1$ entails smoothing over the interval $[n-w+1, n+\ell+1]$, the variables $\mathbf{o}_{x,n-w+1:n+\ell}^{(0)}$, $\mathbf{o}_{y,n-w+1:n+\ell}^{(0)}$ can be initialized to $\mathbf{o}_{x,n-w:n+\ell}^{(J)}$ and $\mathbf{o}_{y,n-w:n+\ell}^{(J)}$ obtained from the previous J iterations, which provides the aforementioned warm start-up. Granted, when the window w is smaller, the effect of the non-robust initialization is more pronounced. Even though no analytical results are claimed on the performance as a function of w, the simulated RMSE comparisons in Section 3.7 with competing alternatives of comparable complexity, speak for the merits of this section's online algorithm.

The novel fixed-lag DRS scheme amounts to sequentially running J KS's and combining their outputs. Interestingly, several non-linear smoothers including those based on SMC and IMM approaches also combine the outputs of several fixed-lag KS's, which allows for a fair comparison of these techniques.

3.6 Generalized linear state-space model

Consider the more general linear state-space model, given by [cf. (3.1a)]

$$\mathbf{x}_n = \mathbf{F}_n \mathbf{x}_{n-1} + \mathbf{G}_n \mathbf{w}_n + \mathbf{o}_{x,n},\ \forall n = 1, \ldots, N. \tag{3.22}$$

where $\{\mathbf{G}_n\}_{n=1}^N$ are known matrices. If matrix $\mathbf{G}_n$ is tall, $\mathbf{G}_n \mathbf{G}_n^T$ is rank deficient, which prevents one from writing the WLS state error as in (3.4). Instead, KS can be formulated as a constrained optimization problem, and likewise for the corresponding fixed-interval and

fixed-lag DRS. Specifically, the novel fixed-interval DRS can be obtained as

$$\begin{aligned}[\widehat{\mathbf{x}}^{\mathrm{DRS}}, \widehat{\bar{\mathbf{w}}}, \widehat{\mathbf{o}}_x, \widehat{\mathbf{o}}_y] \;\; := \;\; & \arg \min_{\mathbf{x}, \bar{\mathbf{w}}, \mathbf{o}_x, \mathbf{o}_y} C^{\mathrm{DRS}}_{\lambda_x, \lambda_y}(\mathbf{x}, \bar{\mathbf{w}}, \mathbf{o}_x, \mathbf{o}_y) \\ & \text{subject to} \quad \mathbf{x}_n = \mathbf{F}_n \mathbf{x}_{n-1} + \mathbf{G}_n \mathbf{w}_n + \mathbf{o}_{x,n}, \; \forall \; n = 1, \ldots, N \end{aligned} \tag{3.23}$$

where $\bar{\mathbf{w}} := [\mathbf{w}_1^T, \mathbf{w}_2^T, \ldots, \mathbf{w}_N^T]^T$ and

$$\begin{aligned} C^{\mathrm{DRS}}_{\lambda_x, \lambda_y}(\mathbf{x}, \bar{\mathbf{w}}, \mathbf{o}_x, \mathbf{o}_y) \;\; = \;\; & \frac{1}{2} \sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{H}_n \mathbf{x}_n - \mathbf{o}_{y,n}\|^2_{\mathbf{R}_n^{-1}} + \frac{1}{2} \|\mathbf{x}_0 - \mathbf{m}_0\|^2_{\mathbf{\Sigma}_0^{-1}} + \frac{1}{2} \sum_{n=1}^{N} \|\mathbf{w}_n\|^2_{\mathbf{Q}_n^{-1}} \\ & + \sum_{n=1}^{N} [\lambda_x \|\mathbf{o}_{x,n}\|_1 + \lambda_y \|\mathbf{o}_{y,n}\|_1]. \end{aligned}$$

Due to the constrained nature of the problem in (3.23), coordinate descent iterations can not be directly applied. However, it is possible to develop iterations based on the alternating direction method of multipliers (AD-MoM) [13]. These iterations are simple if one introduces the auxiliary variables $\mathbf{a}_n = \mathbf{o}_{x,n}$ and $\mathbf{b}_n = \mathbf{o}_{y,n}$ which imply additional constraints. Then, the augmented Lagrangian can be written as

$$\begin{aligned} \mathcal{L}_\kappa \;\; = \;\; & \frac{1}{2} \sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{H}_n \mathbf{x}_n - \mathbf{o}_{y,n}\|^2_{\mathbf{R}_n^{-1}} + \frac{1}{2} \|\mathbf{x}_0 - \mathbf{m}_0\|^2_{\mathbf{\Sigma}_0^{-1}} \\ & + \frac{1}{2} \sum_{n=1}^{N} \|\mathbf{w}_n\|^2_{\mathbf{Q}_n^{-1}} + \sum_{n=1}^{N} [\lambda_y \|\mathbf{b}_n\|_1 + \lambda_x \|\mathbf{a}_n\|_1] \\ & + \sum_{n=1}^{N} \left[\boldsymbol{\chi}_n^T (\mathbf{x}_n - \mathbf{F}_n \mathbf{x}_{n-1} - \mathbf{G}_n \mathbf{w}_n - \mathbf{o}_{x,n}) + \frac{\kappa}{2} \|\mathbf{x}_n - \mathbf{F}_n \mathbf{x}_{n-1} - \mathbf{G}_n \mathbf{w}_n - \mathbf{o}_{x,n}\|_2^2 \right] \\ & + \sum_{n=1}^{N} \left[\boldsymbol{\mu}_n^T (\mathbf{o}_{y,n} - \mathbf{b}_n) + \frac{\kappa}{2} \|\mathbf{o}_{y,n} - \mathbf{b}_n\|_2^2 \right] + \sum_{n=1}^{N} \left[\boldsymbol{\nu}_n^T (\mathbf{o}_{x,n} - \mathbf{a}_n) + \frac{\kappa}{2} \|\mathbf{o}_{x,n} - \mathbf{a}_n\|_2^2 \right]. \end{aligned} \tag{3.24}$$

where $\{\boldsymbol{\chi}_n, \boldsymbol{\mu}_n, \boldsymbol{\nu}_n\}_{n=1}^N$ denote the Lagrange multipliers and κ is a positive constant. Setting the derivatives of $\mathcal{L}_\kappa$ with respect to $\mathbf{x}_n$ equal to zero, yields the following AD-MoM iteration

[cf. (3.24)]

$$\begin{aligned}\mathbf{x}^{(j)} &= \arg\min_{\mathbf{x}} \left\{ \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{H}_n\mathbf{x}_n - \mathbf{o}_{y,n}^{(j-1)}\|^2_{\mathbf{R}_n^{-1}} + \frac{1}{2}\|\mathbf{x}_0 - \mathbf{m}_0\|^2_{\boldsymbol{\Sigma}_0^{-1}} \right. \\ &\qquad \left. + \sum_{n=1}^{N} \frac{\kappa}{2}\|\mathbf{x}_n - \mathbf{F}_n\mathbf{x}_{n-1} - \mathbf{G}_n\mathbf{w}_n^{(j-1)} - \mathbf{o}_{x,n}^{(j-1)} + \boldsymbol{\chi}_n^{(j-1)^T}/\kappa\|_2^2] \right\}.\end{aligned}$$

Clearly, this problem is equivalent to (3.4), which can be solved in a batch or recursive form at complexity that is linear in N. Likewise, the remaining variables are updated as follows:

$$\begin{aligned}
\mathbf{w}_n^{(j)} &= (\mathbf{Q}_n^{-1} + \kappa\mathbf{G}_n^T\mathbf{G}_n)^{-1}\mathbf{G}_n^T(\boldsymbol{\chi}_n^{(j-1)} + \kappa\mathbf{x}_n^{(j)} - \kappa\mathbf{F}_n\mathbf{x}_{n-1}^{(j)} - \kappa\mathbf{o}_{x,n}^{(j-1)}),\ n = 1,\ldots,N \\
\mathbf{o}_{y,n}^{(j)} &= (\mathbf{R}_n^{-1} + \kappa\mathbf{I}_{D_y})^{-1}\big(\mathbf{R}_n^{-1}(\mathbf{y}_n - \mathbf{H}_n\mathbf{x}_n^{(j)}) - \boldsymbol{\mu}_n^{(j-1)^T} + \kappa\mathbf{b}_n^{(j-1)}\big),\ n = 1,\ldots,N \\
\mathbf{o}_{x,n}^{(j)} &= \frac{1}{2}\big(\boldsymbol{\chi}_n^{(j-1)}/\kappa + \mathbf{x}_n^{(j)} - \mathbf{F}_n\mathbf{x}_{n-1}^{(j)} - \mathbf{G}_n\mathbf{w}_n^{(j)} + \mathbf{a}_n^{(j-1)} - \boldsymbol{\nu}_n^{(j-1)}/\kappa\big),\ n = 1,\ldots,N \\
b_{n,d}^{(j)} &= \frac{1}{\kappa}\max\big(|\kappa o_{y,n,d}^{(j)} + \mu_{n,d}^{(j-1)}| - \lambda_y, 0\big)\mathrm{sign}\big(\kappa o_{y,n,d}^{(j)} + \mu_{n,d}^{(j-1)}\big),\ n = 1,\ldots,N,\ d = 1,\ldots,D_y \\
a_{n,d}^{(j)} &= \frac{1}{\kappa}\max\big(|\kappa o_{x,n,d}^{(j)} + \nu_{n,d}^{(j-1)}| - \lambda_x, 0\big)\mathrm{sign}\big(\kappa o_{x,n,d}^{(j)} + \nu_{n,d}^{(j-1)}\big),\ n = 1,\ldots,N,\ d = 1,\ldots,D_x \\
\boldsymbol{\chi}_n^{(j)} &= \boldsymbol{\chi}_n^{(j-1)} + \kappa(\mathbf{x}_n^{(j)} - \mathbf{F}_n\mathbf{x}_{n-1}^{(j)} - \mathbf{G}_n\mathbf{w}_n^{(j)} - \mathbf{o}_{x,n}^{(j)}),\quad n = 1,\ldots,N \\
\boldsymbol{\mu}_n^{(j)} &= \boldsymbol{\mu}_n^{(j-1)} + \kappa(\mathbf{o}_{y,n}^{(j)} - \mathbf{b}_n^{(j)}),\quad n = 1,\ldots,N \\
\boldsymbol{\nu}_n^{(j)} &= \boldsymbol{\nu}_n^{(j-1)} + \kappa(\mathbf{o}_{x,n}^{(j)} - \mathbf{a}_n^{(j)}),\quad n = 1,\ldots,N.
\end{aligned}$$

Invoking the results in [13, p. 256], guarantees global convergence of these iterations as asserted next.

Proposition 8. *For any $\kappa > 0$ and arbitrary initial values $\mathbf{w}^{(0)}, \mathbf{o}_y^{(0)}, \mathbf{o}_x^{(0)}, \mathbf{b}^{(0)}, \mathbf{a}^{(0)}, \boldsymbol{\chi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\nu}^{(0)}$, the AD-MoM iterates are all convergent. Furthermore, every limit point of the sequences $\mathbf{x}^{(j)}$, $\mathbf{w}^{(j)}$, $\mathbf{o}_y^{(j)}$, $\mathbf{o}_x^{(j)}$ is a solution of the problem in (3.23).*

Although global convergence of the coordinate descent and AD-MoM iterates is ensured by Propositions 7 and 8, respectively, no analytical results are available in optimization theory regarding their rate of convergence – a challenging subject going well beyond the scope of the present work.

For AD-MoM iterations too, the bulk of computations is in the order of a KS, which grows linearly in N. The rest involves closed-form evaluations. In a nutshell, for the general linear model in (3.22) the AD-MoM iterations replace those of the coordinate descent algorithm with the same order of computational complexity. Some of the simulations in the ensuing section will test this AD-MoM based fixed-interval DRS approach, which also has a fixed-lag counterpart tailored for online operation under the general linear state-space model. Its derivation follows closely that of the coordinate descent for fixed-interval DRS, and is omitted for brevity.

3.7 Simulated tests: Maneuvering target tracking with glint

In this section, the developed robust smoothers are simulated for maneuvering target tracking in the presence of glint noise. First, DRS performance is tested on a sample target trajectory, and then sample averaged performance metrics for DRS are compared against the main competing alternatives.

3.7.1 DRS on a sample trajectory

The model in (3.22) is simulated with $\mathbf{x}_n := [p_n^x, s_n^x, p_n^y, s_n^y]^T$, where p_n^x and p_n^y denote the target position in the x and y coordinates, respectively; and correspondingly s_n^x and s_n^y denote the target velocity in the x and y directions; thus, $D_x = 4$. The matrices in (3.1b) and (3.22) are invariant $\forall\ n$

$$\mathbf{F}_n := \begin{pmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{G}_n := \begin{pmatrix} \frac{\tau^2}{2} & 0 \\ \tau & 0 \\ 0 & \frac{\tau^2}{2} \\ 0 & \tau \end{pmatrix}, \quad \mathbf{H}_n := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.25}$$

and τ denotes the sampling period. Since $\mathbf{G}_n$ is tall, this so-termed discrete white noise acceleration (DWNA) model [7, p. 273], can only be handled by the generalized linear state-space model of Section 3.6. The form of $\mathbf{H}_n$ in (3.25), shows that $\mathbf{y}_n$ comprises noisy position measurements, and $D_y = 2$.

A total of $N = 100$ observations are collected, $\tau = 1$, $\mathbf{R}_n = 150^2\mathbf{I}_2$, $\mathbf{Q}_n = 0.5\mathbf{I}_2$, $\mathbf{m}_0 = \mathbf{0}_4$, and $\mathbf{\Sigma}_0 = \text{diag}(50, 5, 50, 5)$. The target trajectory starts from position $[0, 0]$, and evolves according to the DWNA model with the specified parameters from time $n = 1$ to $n = 29$. At times $n = 30$ and 31 the target turns right and follows again the DWNA model from $n = 32$ to 59. At time $n = 60$ and 61 the target turns left and then proceeds with the DWNA model until $n = N$. The true target trajectory is depicted in Fig. 3.2 with solid line. The circles represent the acquired position measurements. Three outliers (not depicted in the figure) yield erroneous position reports, at $n = 15$, 50, and 80.

Figure 3.3 depicts the clairvoyant fixed-interval KS estimate. Observe that KS is not robust to outliers in the observations and state. The fixed-interval DRS estimates shown in Fig. 3.4 for $\lambda_y = 0.01$, and $\lambda_x = 0.05$ (which approximately correspond to 10% of the critical $\bar{\lambda}_y$ and $\bar{\lambda}_x$ in Proposition 6), demonstrate that DRS can effectively cope with outliers, and has merits over the non-robust KS even when (λ_x, λ_y) are not systematically estimated as in (3.9) or (3.10). Figures 3.5 and 3.6 depict estimates of the fixed-lag KS in (3.19), and DRS in (3.21) for lag $\ell = 10$, respectively. Again, the KS estimates are strongly affected by outliers. On the other hand, the fixed-lag DRS estimates in Fig. 3.6, for $w = \ell$, $\lambda_y = 0.01$, and $\lambda_x = 0.05$, are only minimally affected by outliers.

3.7.2 Online fixed-lag DRS versus Rao-Blackwellized SMC

The root mean-square error (RMSE) of the position estimates is used here to quantify the performance improvement of DRS relative to KS. The true target trajectory coincides with that of Fig. 3.2 (solid line), when $M = 100$ noise and outlier realizations are present. With

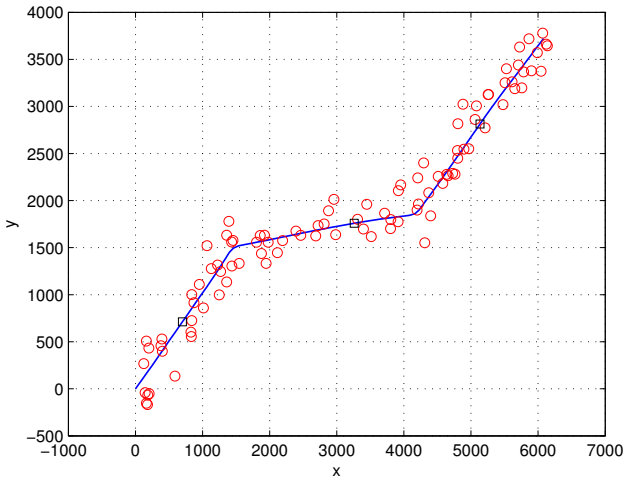

Figure 3.2: True target trajectory (solid line); Observed positions (circles). The squares indicate the trajectory instants where outliers occur ($n = 15, 50$, and 80). Outlier-corrupted measurement values are $\mathbf{y}_{15} = [-5560, 18440]^T$, $\mathbf{y}_{50} = [3880, 14440]^T$, and $\mathbf{y}_{80} = [6440, -14800]^T$.

probability $\pi = 0.97$, the model in (3.1b) was in effect with $\mathbf{o}_y = \mathbf{0}$, $\mathbf{H}_n$ in (3.25), and $\mathbf{R}_n = 150^2\mathbf{I}_2$. With probability $1-\pi = 0.03$, outliers in the observations occur, and in this case the position reports are $[y_{n,1}, y_{n,2}] \sim \mathcal{U}([-10000, 10000]^2)$. Figure 3.7 depicts the RMSE of the position estimates, $\text{RMSE}_n = \sqrt{\frac{1}{M}\sum_{m=1}^{M}[(p_n^x - \widehat{p}_n^{x(m)})^2 + (p_n^y - \widehat{p}_n^{y(m)})^2]}$, where $[\widehat{p}_n^{x(m)}, \widehat{p}_n^{y(m)}]$ is the estimated position at time n for the mth noise and outlier realization, for the fixed-interval KS and DRS with $\lambda_y = 0.01$, and $\lambda_x = 0.05$. Clearly, DRS exhibits lower RMSE than the clairvoyant KS.

Figure 3.8 depicts the instantaneous RMSE for fixed-lag KS and DRS for $\ell = 10$, $w = 10$, $\lambda_y = 0.01$, and $\lambda_x = 0.05$, along with the Rao-Blackwellized (RB) SMC smoother relying on 50 particles, and the online fixed-lag DRS with 50 AD-MoM iterations and constant $\kappa = 0.05$ (referred in the figure as O-DRS). For the RB-SMC smoother, a conditionally

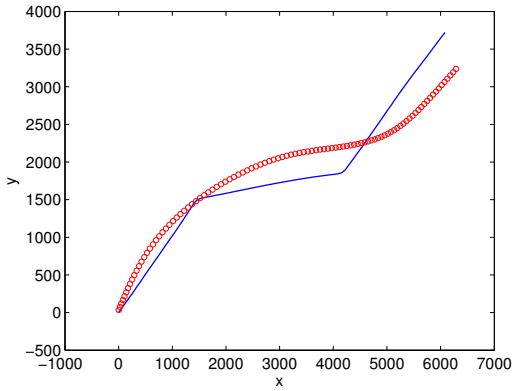

Figure 3.3: True target trajectory (solid line) and estimated trajectory (circles) using fixed-interval KS.

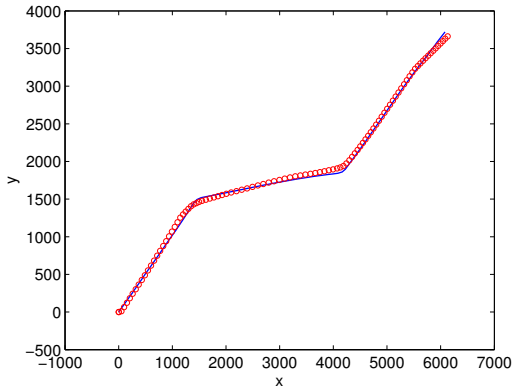

Figure 3.4: True target trajectory (solid line) and estimated trajectory (circles) using fixed-interval DRS ($\lambda_y = 0.01, \lambda_x = 0.05$).

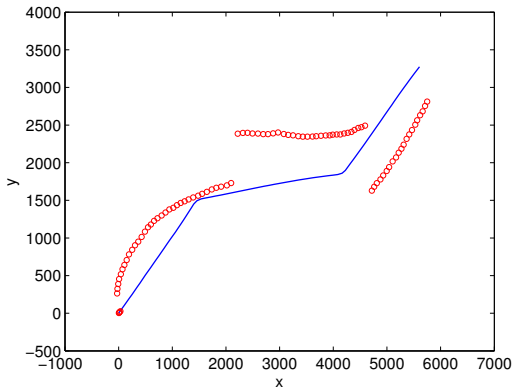

Figure 3.5: True target trajectory (solid line) and estimated trajectory (circles) using fixed-lag KS.

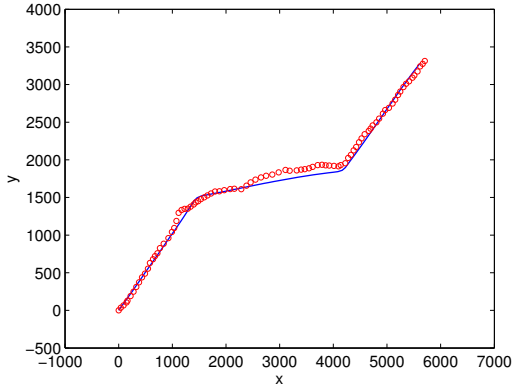

Figure 3.6: True target trajectory (solid line) and estimated trajectory (circle) using fixed-lag DRS ($\lambda_y = 0.01, \lambda_x = 0.05$).

linear, Gaussian model is adopted. Specifically, under nominal conditions, the model is that in (3.1) with $\mathbf{F}_n$, $\mathbf{G}_n$, and $\mathbf{H}_n$ as in (3.25), $\mathbf{o} = \mathbf{0}$, $\mathbf{Q}_n = 0.5\mathbf{I}_2$, and $\mathbf{R}_n = 150^2\mathbf{I}_2$. It

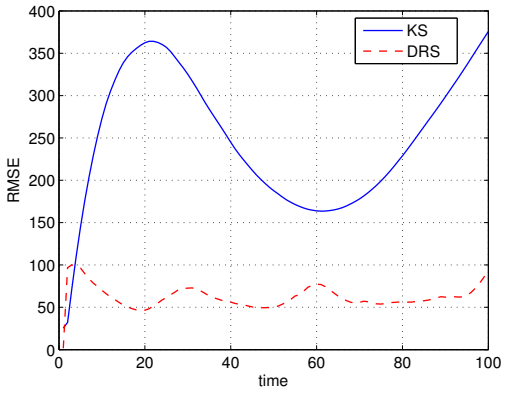

Figure 3.7: RMSE analysis of the fixed-interval KS versus DRS ($\lambda_y = 0.01, \lambda_x = 0.05$).

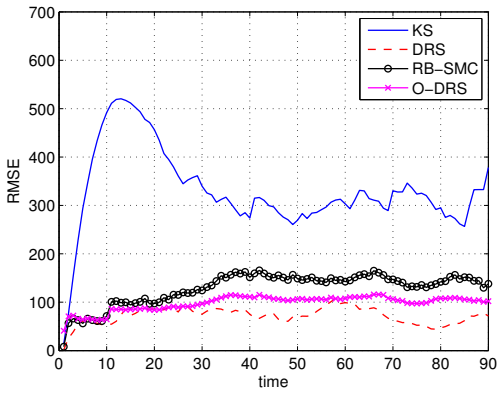

Figure 3.8: RMSE analysis of the fixed-lag KS versus DRS ($\lambda_y = 0.01, \lambda_x = 0.05$), online DRS ($\kappa > 0$, $\lambda_y = 0.01, \lambda_x = 0.05$, $J = 50$ AD-MoM iterations), and Rao-Blackwellized SMC smoother (50 particles).

is assumed that the nominal conditions are in effect with probability $\pi = 0.97$. Outliers in the measurements occur with probability $1 - \pi = 0.03$, and in this case $\mathbf{R}_n = 15000^2\mathbf{I}_2$, which allows for down-weighting the respective measurements. With the same probability, state outliers emerge too, and in this case $\mathbf{Q}_n = 500\mathbf{I}_2$. Clearly, conditioned on the outlier realizations, the dynamical process is linear and Gaussian. This allows for drawing particles for the state/measurement outliers, and using them for estimating the state via fixed-lag KS; see also [37] for details on the fixed-lag RB-SMC. In addition to fixed-lag KS, the novel O-DRS approach outperforms RB-SMC for the same computational burden.

3.7.3 Comparison with RANSAC and Huber M-estimates

DRS is compared here against state-of-the-art robust smoothers, namely the Huber based scheme, and a combined RANSAC followed by Huber scheme. In the latter, smoothing is

cast as the linear regression problem in (3.3) to which RANSAC can be applied readily [62]. RANSAC relies on random draws (here 100 or 1,000) to find the "best" possible subset of rows corresponding to the nominal model [49, 62]. To ensure that the remaining outliers do not degrade performance, the nominal rows of (3.3) found by RANSAC are pre-whitened, and subsequently plugged into Huber's cost in (3.7). The Huber parameters are set to $\lambda_x = \lambda_y = 1.345$ as suggested by [53, 67] for standardized Gaussian nominal noise (this requires pre-whitening the nominal noise). The Huber estimate is found by solving (3.7) using the iteratively re-weighted least-squares (IRLS) algorithm in [67], which unlike the Lasso-based solver pursued here, guarantees only local convergence. The fixed-interval DRS in (3.6) is also employed with λ_x and λ_y found using either of the two data-driven criteria suggested in Section 3.3. To further improve DRS, one iteration of the refined estimate in Remark 6 is also implemented. The model simulated here obeys (3.25), but with $\mathbf{G}_n = \mathbf{I}_4$, $\mathbf{w}_n \sim \mathcal{N}(\mathbf{0}_4, \mathbf{Q}_n)$, $\mathbf{Q}_n = \text{diag}(1, 0.001, 1, 0.001)$, and $\mathbf{R}_n = 5\mathbf{I}_2$. State and measurement outliers are generated as independent Laplacian with variances 200 and 20,000, respectively. The RMSE for both position and velocity estimates time-averaged over 100 Monte Carlo runs is plotted versus the percentage of outlier contamination.

Fig. 3.9 plots the RMSE versus percentage of outliers for the combined RANSAC-Huber robust smoother as well as DRS, when outliers appear only in the measurements. The numerical suffix for DRS denotes the grid size used for the AVD [cf. (3.10)], while the one for RANSAC stands for the number of RANSAC's random draws. In terms of complexity, DRS-100 (with $I_x = I_y = 10$ grid points equispaced in log scale as suggested in [51]) lies in-between RANSAC-100 and RANSAC-1000. Note that up to 50% outliers all three methods perform similarly. When the outlier contamination percentage exceeds 50%, RANSAC-100 performs poorly, while RANSAC-1000 and DRS-100 exhibit graceful performance degradation. Due to its lower-complexity, DRS-100 offers a better alternative than RANSAC-1000.

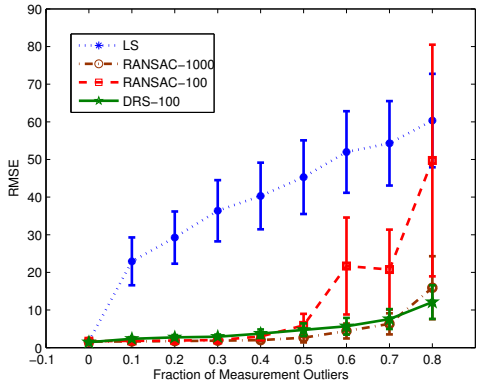

Figure 3.9: Mean RMSE ± std. deviation for estimates formed by RANSAC followed by Huber robustification versus DRS: Measurement outliers only.

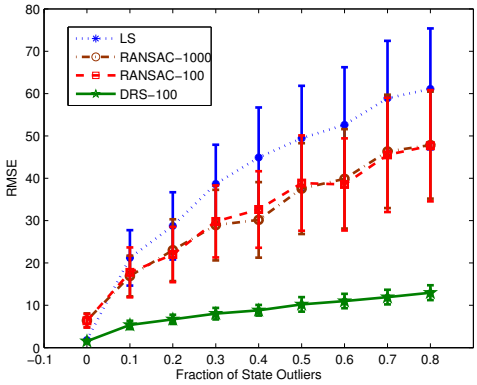

Figure 3.10: Mean RMSE ± std. deviation for estimates formed by RANSAC followed by Huber robustification versus DRS: State outliers only.

Fig. 3.10 compares RMSE performance when outliers are only present in the state. One observes that DRS-100 considerably outperforms both versions of RANSAC for all percentages of outlier contamination. The improvement in going from RANSAC-100 to RANSAC-1000 is not noticeable.

Fig. 3.11 plots the RMSE resulting from DRS-100, batch KS, Huber-only, and the combined RANSAC-Huber scheme when outliers are simultaneously present in the state and measurements. The AVD criterion is used for DRS. It can be seen that RANSAC-Huber combination performs poorly for outliers present in the state and measurement. This happens because RANSAC removes certain rows of the regression matrix, namely those contaminated by outliers, which renders the remaining sub-matrix ill-conditioned. Huber-only performs close but worse than KS – a manifestation of the fact that Huber's estimate are found for independent nominal noise. Even though the noise here is independent, it is not standard Gaussian and the subsequent pre-whitening, which is a mere scaling in this

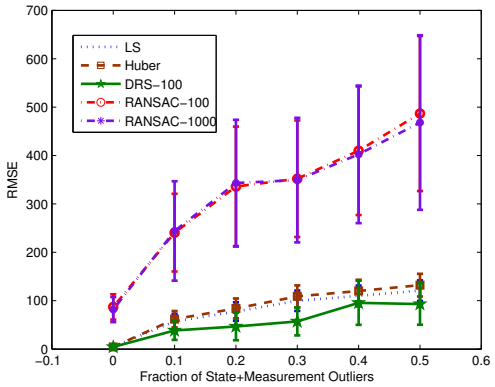

Figure 3.11: Outliers present in state and measurements.

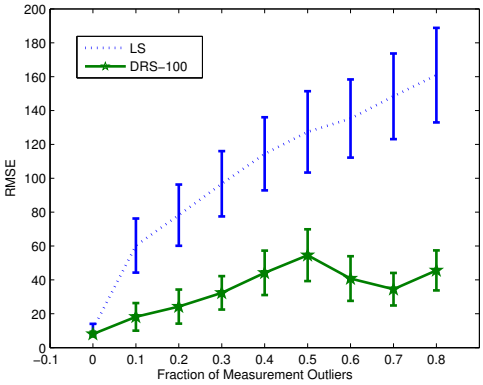

Figure 3.12: DRS versus LS with known percentage of outliers.

case, adversely affects the Huber-based estimate. Indeed, neither of the mentioned robust methods performs well when outliers are present both in the state and measurement, and surprisingly even the clairvoyant KS outperforms them. However, DRS-100 outperforms KS in terms of RMSE, and speaks for the importance of the universality property of the novel estimator.

The DRS improvement over KS is more pronounced if the percentage of outliers is known, case where (3.9) is used instead of AVD. The result is plotted in Fig. 3.12, where DRS significantly outperforms KS. Here, the percentage of state outliers is fixed at 10%, while that of measurement outliers is variable.

At last, different DRS renditions are compared against each other and with the robust smoother of [5]. For a fair comparison with [5], the setup includes outliers only in the measurements and smoothed estimates for both approaches are formed using the general-purpose optimization software SeDuMi [103]. Each randomly occurring outlying-measurement is drawn from a zero-mean uniform distribution with variance $20,000$ independently from the nominal random variables. Note that the outlying measurements here are

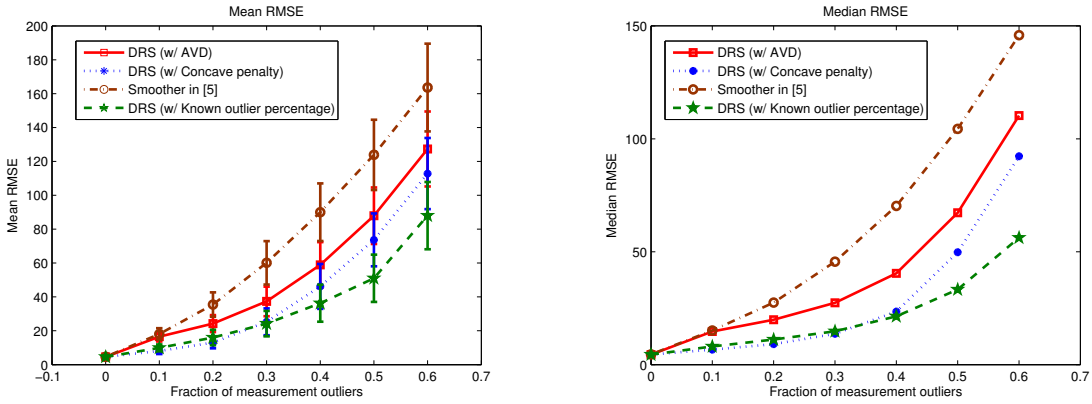

Figure 3.13: Comparison among different DRS renditions with the smoother in [5]: (left) Mean RMSE $\pm$ std. deviation; (right) Median RMSE.

generated not in accordance with the model in [5] in order to illustrate the universality attribute of the proposed DRS. Nominal model parameters commonly known to both DRS and the smoother [5], are chosen as before with the only difference that $\mathbf{Q}_n = \text{diag}(100, 1, 100, 1)$. Fig. 3.13 depicts the mean and median RMSE computed over $1,000$ Monte Carlo runs as a function of the percentage of outliers. It can be seen that DRS with AVD outperforms the smoother in [5], especially as the fraction of outlier contamination exceeds 10%. Similar to the smoother in [5], DRS with AVD utilizes only nominal parameter knowledge to form its estimate. The curve that utilizes the concave penalty pertains to the refinement outlined in Remark 6. A clear gain is observed with this refinement as a result of de-biasing the DRS estimates. DRS with known percentage of outliers is also plotted and outperforms all other alternatives. This is due to the extra knowledge on outlier sparsity that this smoother benefits from.

3.8 Summary

Robust smoothers were developed for dynamical processes contaminated with outliers in the observations and/or state. The novel fixed-interval DRS can be viewed as an ℓ_1-norm regularized version of the WLS-based clairvoyant KS algorithm. This form of regularization controls the sparsity of outliers, which are explicitly introduced as auxiliary variables. Two data-driven methods were also devised to select the associated regularization parameters. Block coordinate descent-based iterations were developed to solve the underlying convex optimization problem in an efficient manner. To enable real-time smoothing for delay-constrained applications such as target tracking, an online fixed-lag DRS was also developed. At last, the novel approach was broadened to include generalized linear state-space models. Numerical tests demonstrated that the proposed algorithms can jointly cope with state and measurement outliers, and outperform state-of-the-art methods at comparable computational burden.

3.9 Appendices

3.9.1 Proof of Proposition 4 (MAP optimality of DRS in (3.6))

Successive application of Bayes' rule, as well as the assumptions on independence and the corresponding distributions of the nominal noise and outlier vectors yield

$$\begin{aligned} p(\mathbf{x},\mathbf{o}_x,\mathbf{o}_y|\mathbf{y}_{1:N}) &= \frac{p(\mathbf{y}_{1:N},\mathbf{x},\mathbf{o}_x,\mathbf{o}_y)}{p(\mathbf{y}_{1:N})} \\ &\propto p(\mathbf{x}_0)\prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{x}_{0:n},\mathbf{y}_{1:n-1},\mathbf{o}_{x,1:n},\mathbf{o}_{y,1:n}) \\ &\quad \times p(\mathbf{x}_n|\mathbf{x}_{0:n-1},\mathbf{y}_{1:n-1},\mathbf{o}_{x,1:n},\mathbf{o}_{y,1:n}) \\ &\quad \times p(\mathbf{o}_{y,n}|\mathbf{x}_{0:n-1},\mathbf{y}_{1:n-1},\mathbf{o}_{x,1:n},\mathbf{o}_{y,1:n-1}) \\ &\quad \times p(\mathbf{o}_{x,n}|\mathbf{x}_{0:n-1},\mathbf{y}_{1:n-1},\mathbf{o}_{x,1:n-1},\mathbf{o}_{y,1:n-1}) \\ &= p(\mathbf{x}_0)\prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{x}_n,\mathbf{o}_{y,n})p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathbf{o}_{x,n})p(\mathbf{o}_{y,n})p(\mathbf{o}_{x,n}) \\ &= \mathcal{N}(\mathbf{x}_0;\mathbf{m}_0,\mathbf{\Sigma}_0)\prod_{n=1}^{N}\mathcal{N}(\mathbf{y}_n;\mathbf{H}_n\mathbf{x}_n+\mathbf{o}_{y,n},\mathbf{R}_n) \\ &\quad \times \mathcal{N}(\mathbf{x}_n;\mathbf{F}_n\mathbf{x}_{n-1}+\mathbf{o}_{x,n},\mathbf{Q}_n)\mathcal{L}(\mathbf{o}_{x,n};\lambda_x)\mathcal{L}(\mathbf{o}_{y,n};\lambda_y) \end{aligned} \tag{3.26}$$

where $\mathcal{N}(\mathbf{x};\mathbf{m}_0,\mathbf{\Sigma}_0)$ represents a Gaussian distribution with mean $\mathbf{m}_0$ and covariance $\mathbf{\Sigma}_0$, while $\mathcal{L}(\mathbf{o};\lambda) := \prod_d(\lambda/2)\exp(-\lambda|o_d|) \propto \exp(-\lambda\|\mathbf{o}\|_1)$ represents the joint Laplacian distribution for a vector with independent entries. Maximizing (3.26) amounts to minimizing the negative of the exponent, which leads to the DRS criterion in (3.6).

3.9.2 Proof of Proposition 5 (Equivalence of (3.6) with (3.7))

Consider minimizing the cost in (3.6) over $\mathbf{o}_y$ and $\mathbf{o}_x$, with $\mathbf{x}$ fixed. Given $\mathbf{x}$ and with $\{\mathbf{Q}_n,\mathbf{R}_n\}_{n=1}^N$ and $\mathbf{\Sigma}_0$ given by identity matrices, the criterion in (3.6) is separable over each

scalar entry of $\mathbf{o}_y$ and $\mathbf{o}_x$. Hence, it suffices to find

$$\widehat{o}_{y,n,d} = \arg\min_{o_{y,n,d}} \left\{ \frac{1}{2}(y_{n,d} - \mathbf{h}_{n,d}^T \mathbf{x}_n - o_{y,n,d})^2 + \lambda_y |o_{y,n,d}| \right\}, n = 1, \ldots, N,\ d = 1, \ldots, D_y \tag{3.27a}$$

$$\widehat{o}_{x,n,d} = \arg\min_{o_{x,n,d}} \left\{ \frac{1}{2}(x_{n,d} - \mathbf{f}_{n,d}^T \mathbf{x}_{n-1} - o_{x,n,d})^2 + \lambda_x |o_{x,n,d}| \right\}, n = 1, \ldots, N,\ d = 1, \ldots, D_x \tag{3.27b}$$

The scalar problems in (3.27) admit, respectively, the following closed-form solutions (see, e.g., [95]):

$$\widehat{o}_{y,n,d} = \begin{cases} 0, & \text{if } |y_{n,d} - \mathbf{h}_{n,d}^T \mathbf{x}_n| \leq \lambda_y \\ y_{n,d} - \mathbf{h}_{n,d}^T \mathbf{x}_n - \lambda_y \ \text{sign}(y_{n,d} - \mathbf{h}_{n,d}^T \mathbf{x}_n), & \text{otherwise} \end{cases} \tag{3.28a}$$

$$\widehat{o}_{x,n,d} = \begin{cases} 0, & \text{if } |x_{n,d} - \mathbf{f}_{n,d}^T \mathbf{x}_{n-1}| \leq \lambda_x \\ x_{n,d} - \mathbf{f}_{n,d}^T \mathbf{x}_{n-1} - \lambda_x \ \text{sign}(x_{n,d} - \mathbf{f}_{n,d}^T \mathbf{x}_{n-1}), & \text{otherwise} \end{cases} \tag{3.28b}$$

Substituting (3.28) in the DRS cost of (3.6), the subsequent optimization problem in $\mathbf{x}$ is

$$\begin{aligned} \widehat{\mathbf{x}} := \arg\min_{\mathbf{x}} \Bigg[& \sum_{n=1}^{N} \sum_{d=1}^{D_y} \Bigg(\frac{1}{2}(y_{n,d} - \mathbf{h}_{n,d}^T \mathbf{x}_n)^2 \mathbb{1}_{|y_{n,d} - \mathbf{h}_{n,d}^T \mathbf{x}_n| \leq \lambda_y}(\mathbf{x}) \\ & + \big(\lambda_y |y_{n,d} - \mathbf{h}_{n,d}^T \mathbf{x}_n| - \lambda_y^2/2\big) \mathbb{1}_{|y_{n,d} - \mathbf{h}_{n,d}^T \mathbf{x}_n| > \lambda_y}(\mathbf{x}) \Bigg) + \sum_{d=1}^{D_x} \left(\frac{1}{2}(x_{0,d} - m_{0,d})^2 \right) \\ & + \sum_{n=1}^{N} \sum_{d=1}^{D_x} \Bigg(\frac{1}{2}(x_{n,d} - \mathbf{f}_{n,d}^T \mathbf{x}_{n-1})^2 \mathbb{1}_{|x_{n,d} - \mathbf{f}_{n,d}^T \mathbf{x}_{n-1}| \leq \lambda_x}(\mathbf{x})) \\ & + \big(\lambda_x |x_{n,d} - \mathbf{f}_{n,d}^T \mathbf{x}_{n-1}| - \lambda_x^2/2\big) \mathbb{1}_{|x_{n,d} - \mathbf{f}_{n,d}^T \mathbf{x}_{n-1}| > \lambda_x}(\mathbf{x}) \Bigg) \Bigg]. \end{aligned} \tag{3.29}$$

Given the definition of Huber's cost, the problem in (3.29) is equivalent to (3.7). Therefore, (3.6) and (3.7) are equivalent.

3.9.3 Proof of Proposition 6

Letting $L(\mathbf{x}, \mathbf{o}_x, \mathbf{o}_y)$ denote the cost in (3.6) and $\check{\nabla}$ the subgradient operator, the optimality conditions for the non-differentiable problem in (3.6) are [100, p. 126]

$$\mathbf{0} \in \check{\nabla}_{\mathbf{o}_{y,n}} L(\hat{\mathbf{x}}, \hat{\mathbf{o}}_x, \hat{\mathbf{o}}_y) \;\; \Rightarrow \;\; \mathbf{0} \in \mathbf{R}_n^{-1}\hat{\mathbf{o}}_{y,n} - \mathbf{R}_n^{-1}(\mathbf{y}_n - \mathbf{H}_n\hat{\mathbf{x}}_n) + \lambda_y \check{\mathbf{o}}_{y,n} \tag{3.30a}$$

$$\mathbf{0} \in \check{\nabla}_{\mathbf{o}_{x,n}} L(\hat{\mathbf{x}}, \hat{\mathbf{o}}_x, \hat{\mathbf{o}}_y) \;\; \Rightarrow \;\; \mathbf{0} \in \mathbf{Q}_n^{-1}\hat{\mathbf{o}}_{x,n} - \mathbf{Q}_n^{-1}(\hat{\mathbf{x}}_n - \mathbf{F}_n\hat{\mathbf{x}}_{n-1}) + \lambda_x \check{\mathbf{o}}_{x,n} \tag{3.30b}$$

where $\check{\mathbf{o}}_{x,n} := [\check{o}_{x,n,1}, \check{o}_{x,n,2}, \ldots, \check{o}_{x,n,D_x}]^T$ and $\check{\mathbf{o}}_{y,n} := [\check{o}_{y,n,1}, \check{o}_{y,n,2}, \ldots, \check{o}_{y,n,D_y}]^T$ are the subgradients of $\|\mathbf{o}_{x,n}\|_1$ and $\|\mathbf{o}_{y,n}\|_1$, respectively, whose dth entries are given by

$$\check{o}_{y,n,d} = \begin{cases} \operatorname{sign}(\hat{o}_{y,n,d}), & \hat{o}_{y,n,d} \neq 0 \\ s_{n,d}, & \hat{o}_{y,n,d} = 0 \end{cases}, \quad \check{o}_{x,n,d} = \begin{cases} \operatorname{sign}(\hat{o}_{x,n,d}), & \hat{o}_{x,n,d} \neq 0 \\ t_{n,d}, & \hat{o}_{x,n,d} = 0 \end{cases},$$

for any $|s_{n,d}| \leq 1$ and $|t_{n,d}| \leq 1$.

DRS coincides with KS when $\hat{\mathbf{o}}_y = \mathbf{0}_{ND_y}$ and $\hat{\mathbf{o}}_x = \mathbf{0}_{ND_x}$, which implies $\hat{\mathbf{x}} := \hat{\mathbf{x}}^{\mathrm{KS}}$. For $\hat{\mathbf{o}}_y = \mathbf{0}_{ND_y}$, (3.30a) is satisfied with $\hat{\mathbf{x}} := \hat{\mathbf{x}}^{\mathrm{KS}}$ if and only if (3.8a) holds. Similarly, for $\hat{\mathbf{o}}_x = \mathbf{0}_{ND_x}$, (3.30b) is satisfied with $\hat{\mathbf{x}} := \hat{\mathbf{x}}^{\mathrm{KS}}$ if and only if (3.8b) holds. QED

Chapter 4

Robust RLS in the presence of correlated noise using outlier sparsity

All recursive least-squares (RLS) approaches promising robustness to measurement outliers assume that the additive noise is uncorrelated. However, this assumption may not hold in e.g., sensor networks where spatially close sensors can experience correlated background noise. In sensor networks, the outliers in the measurements can arise due to impulsive noise or faulty operation of unreliable low-cost sensor nodes. When the background noise is correlated, these methods have to ignore noise correlations as pre-whitening spreads the outliers to non-contaminated measurements. This results in poor utilization of available data and unsatisfactory performance. In this work, a robust RLS algorithm is derived that can handle outliers as well as noise correlations simultaneously. In the proposed method, outliers are treated as nuisance variables and jointly estimated with the wanted parameters. To ensure identifiability and account for outlier sparsity, the least-squares (LS) criterion is

regularized by the ℓ_1-norm of outlier variables leading to an optimization problem whose solution yields the robust RLS estimates. For low-complexity real-time operation, a sub-optimal online algorithm is proposed which entails closed-form updates per time step in the same spirit as RLS. Simulations demonstrate the effectiveness and improved performance of the proposed method in comparison with the clairvoyant RLS and its state-of-the-art robust renditions.

4.1 Problem statement and preliminaries

Consider an outlier-aware linear regression model where vector measurements $\mathbf{y}_k \in \mathbb{R}^{D_y}$ involving the unknown vector $\mathbf{x}_k \in \mathbb{R}^{D_x}$ arrive sequentially in time (indexed by k), in the presence of (possibly correlated) nominal noise; that is

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k + \mathbf{o}_k \tag{4.1}$$

where $\mathbf{H}_k$ is the known regression matrix, $\mathbf{v}_k$ denotes the nominal background noise with zero-mean and known covariance matrix $\mathbf{R}_k$, and $\mathbf{o}_k$ models the outliers arising from e.g., impulsive noise. Vector $\mathbf{v}_k$ is independent of $\mathbf{v}_{1:k-1} := [\mathbf{v}_1^T, \mathbf{v}_2^T, \ldots, \mathbf{v}_{k-1}^T]^T$, but can be correlated across its own entries. The wanted vector $\mathbf{x}_k$ is either constant, or, it is slowly varying with k. The goal is to estimate $\mathbf{x}_k$ and $\mathbf{o}_k$ given the measurements $\mathbf{y}_{1:k} := [\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_k^T]^T$.

Had $\mathbf{o}_k$ been equal to zero or known for that matter, one could apply the RLS algorithm on the outlier compensated measurements $\mathbf{y}_k - \mathbf{o}_k$. Therefore, at time step k, one would solve

$$\hat{\mathbf{x}}_k := \arg \min_{\mathbf{x}_k} \sum_{i=1}^{k} \beta^{k-i} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_k - \mathbf{o}_i\|^2_{\mathbf{R}_i^{-1}} \tag{4.2}$$

where $0 < \beta \leq 1$ is a known forgetting factor. Equating to zero derivatives of the cost in

(4.2) with respect to $\mathbf{x}_k$, yields

$$\left(\sum_{i=1}^{k}\beta^{k-i}\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i\right)\hat{\mathbf{x}}_k=\left(\sum_{i=1}^{k}\beta^{k-i}\mathbf{H}_i^T\mathbf{R}_i^{-1}(\mathbf{y}_i-\mathbf{o}_i)\right)\quad\Rightarrow\quad\mathbf{\Phi}_k^{-1}\hat{\mathbf{x}}_k=\mathbf{c}_k$$

which can be solved online using the following RLS-like recursions

$$\begin{aligned}\hat{\mathbf{x}}_k&=\hat{\mathbf{x}}_{k-1}+\mathbf{W}_k(\mathbf{y}_k-\mathbf{o}_k-\mathbf{H}_k\hat{\mathbf{x}}_{k-1})\\ \mathbf{W}_k&=\beta^{-1}\mathbf{\Phi}_{k-1}\mathbf{H}_k^T(\mathbf{H}_k\beta^{-1}\mathbf{\Phi}_{k-1}\mathbf{H}_k^T+\mathbf{R}_k)^{-1}\\ \mathbf{\Phi}_k&=\beta^{-1}\mathbf{\Phi}_{k-1}-\mathbf{W}_k\mathbf{H}_k\beta^{-1}\mathbf{\Phi}_{k-1}.\end{aligned}\tag{4.3}$$

Unfortunately, the outlier vector $\mathbf{o}_k$ is unknown and should be estimated alongside $\mathbf{x}_k$ by solving the under determined problem in (4.1). A quick look reveals that the criterion in (4.2) is not the right choice to be optimized jointly with respect to $\mathbf{x}_k$ and $\mathbf{o}_i$ simply because identifiability issues arise. For example, $\mathbf{o}_k=\mathbf{y}_k$ and $\mathbf{x}_k=\mathbf{0}$ is one choice of parameters that does minimize the cost in (4.2). Clearly, this is not desirable. The next section develops algorithms allowing one to jointly estimate $\mathbf{o}_k$ and $\mathbf{x}_k$.

4.2 Outlier sparsity-aware robust RLS

To ensure uniqueness in the joint optimization of $\mathbf{o}_{1:k}$ and $\mathbf{x}_k$, one can regularize the cost in (4.2). Among candidate regularization terms, popular ones include the ℓ_p-norm of outliers with $p=0,1,2$. For selecting p, it is worth observing that outliers arising from sources such as impulsive noise are sparse. This sparsity can be enforced through the ℓ_0-norm regularizer. Unfortunately, the ℓ_0-norm makes the optimization problem non-convex and in fact NP-hard to solve. Instead, the ℓ_1-norm can be selected which is both convex and can yield a solution close to that of ℓ_0-norm under certain circumstances [21]. Penalizing (4.2) with the ℓ_1-norm of the outlier vector yields

$$[\hat{\mathbf{x}}_k,\hat{\mathbf{o}}_{1:k}]=\arg\min_{\mathbf{x}_k,\mathbf{o}_{1:k}}\sum_{i=1}^{k}\beta^{k-i}\left[\frac{1}{2}\|\mathbf{y}_i-\mathbf{H}_i\mathbf{x}_k-\mathbf{o}_i\|^2_{\mathbf{R}_i^{-1}}+\lambda_i\|\mathbf{o}_i\|_1\right].\tag{4.4}$$

Besides being convex, (4.4) can be nicely linked with Huber's M-estimates [56]. Indeed, when $\mathbf{R}_i$'s are identity matrices for all i (i.e., when the nominal noise is uncorrelated), solving (4.4) amounts to solving [56]

$$\hat{\mathbf{x}}_k = \arg \min_{\mathbf{x}_k} \sum_{i=1}^{k} \sum_{d=1}^{D_y} \beta^{k-i} \rho_i(y_{i,d} - \mathbf{h}_{i,d}^T \mathbf{x}_k) \tag{4.5}$$

where $\mathbf{y}_i := [y_{i,1}, y_{i,2}, \ldots, y_{i,D_y}]$, $\mathbf{H}_i := [\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, \ldots, \mathbf{h}_{i,D_y}]^T$, and

$$\rho_i(x) := \begin{cases} \frac{x^2}{2}, & |x| \leq \lambda_i \\ |x|\lambda_i - \frac{\lambda_i^2}{2}, & |x| > \lambda_i \end{cases}$$

is the Huber function. While solving (4.4) recovers an estimate of $\mathbf{x}_k$, its practicality is compromised by the fact that the number of variables to be estimated increases with time. In addition, (4.4) can not be posed in a recursive form and at each time instant the whole optimization should be carried out anew. This shortcoming affects Huber M-estimates too, and similar limitations have been identified also by [24] and [121]. Specifically, the recursions in [24] and [121] are sub-optimal and solve the posed optimization problems only approximately. Here too, the offline benchmark solver of (4.4) will be used as a "warm-start" to initialize online methods developed in the ensuing subsection.

4.2.1 Online robust RLS (OR-RLS)

Since it is impractical to re-estimate past and current outliers per time instant k, the main idea in this section is to estimate only the most recent outliers. Consequently, the proposed OR-RLS proceeds in two steps. First, given the estimate $\hat{\mathbf{x}}_{k-1}$ from $k-1$, the current outlier vector $\mathbf{o}_k$ is estimated via

$$\hat{\mathbf{o}}_k = \arg \min_{\mathbf{o}_k} \left[\frac{1}{2} \|\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1} - \mathbf{o}_k\|_{\mathbf{R}_k^{-1}}^2 + \lambda_k \|\mathbf{o}_k\|_1 \right]. \tag{4.6}$$

Once the estimate $\hat{\mathbf{o}}_k$ becomes available, ordinary RLS is applied to the outlier-free measurements $\mathbf{y}_k - \hat{\mathbf{o}}_k$ [cf. (4.3)]. The second stage of OR-RLS incurs a complexity comparable

to ordinary RLS. Clearly, (4.6) solves a problem of fixed dimension D_y per time step; hence, it offers a practical alternative to the offline benchmark in (4.4).

When $\mathbf{R}_k$ is diagonal, denote it as $\text{diag}(r_{k,11}, r_{k,22}, \ldots, r_{k,D_yD_y})$, (4.6) decouples into scalar sub-problems across entries of $\mathbf{o}_k$ as

$$\hat{o}_{k,d} = \arg \min_{o_{k,d}} \left[\frac{(y_{k,d} - \mathbf{h}_{k,d}^T \hat{\mathbf{x}}_{k-1} - o_{k,d})^2}{2r_{k,dd}} + \lambda_k |o_{k,d}| \right], \quad \forall d = 1, \ldots, D_y.$$

The solution to these sub-problems is available in closed form using the so termed least-absolute shrinkage and selection operator (LASSO) [110]. Here, it is given by

$$\hat{o}_{k,d} = \max\left\{ |y_{k,d} - \mathbf{h}_{k,d}^T \hat{\mathbf{x}}_{k-1}| - \lambda_k r_{k,dd}, 0 \right\} \text{sign}(y_{k,d} - \mathbf{h}_{k,d}^T \hat{\mathbf{x}}_{k-1}), \quad \forall d = 1, \ldots, D_y.$$

Thus, OR-RLS amounts to simple non-iterative closed-form evaluations per time step k. However, when $\mathbf{R}_k$ is non-diagonal, (4.6) can no longer be solved in closed-form.

When $\mathbf{R}_k$ is non-diagonal, general-purpose convex solvers such as SeDuMi [103] can be utilized to solve (4.6). This is not desirable from a complexity perspective though. In the next subsections, two iterative methods based on coordinate descent (CD), and the alternating direction method of multipliers (AD-MoM) are developed to solve (4.6) iteratively. While both methods are provably convergent, it suffices to run only a few iterations of either method in practice, which will maintain the low-complexity characteristic of OR-RLS.

4.2.2 CD based OR-RLS

To iteratively solve (4.6) with respect to $\mathbf{o}_k$, consider initializing with $\mathbf{o}_k^{(0)} = \mathbf{0}$. Then, for $j = 1, \ldots, J$, vector $\mathbf{o}_k^{(j)}$ is updated one entry at a time using coordinate descent.

Specifically, the d'th entry of $\mathbf{o}_k$ at iteration j is updated as

$$o_{k,d}^{(j)} := \arg\min_{o_{k,d}} \frac{1}{2} \left\| \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1} - \begin{bmatrix} \mathbf{o}_{k,1:d-1}^{(j)} \\ o_{k,d} \\ \mathbf{o}_{k,d+1:D_y}^{(j-1)} \end{bmatrix} \right\|_{\mathbf{R}_k^{-1}}^2 + \lambda_k |o_{k,d}|. \tag{4.7}$$

Problem (4.7) can be alternatively written as a scalar Lasso one, that is

$$o_{k,d}^{(j)} := \arg\min_{o_{k,d}} \frac{1}{2} \left(o_{k,d} - \gamma_{k,d}^{(j)} \right)^2 + \lambda_{k,d} |o_{k,d}| \tag{4.8}$$

where

$$\begin{aligned} \gamma_{k,d}^{(j)} :=& \frac{1}{\tilde{r}_{k,d,d}} [\alpha_{k,d} - \sum_{i=1}^{d-1} \tilde{r}_{k,i,d} o_{k,i}^{(j)} - \sum_{i=d+1}^{D_y} \tilde{r}_{k,i,d} o_{k,i}^{(j-1)}] \\ \boldsymbol{\alpha}_k \;\; :=& \;\; \mathbf{R}_k^{-1} \left(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1} \right), \qquad \lambda_{k,d} := \lambda_k / \tilde{r}_{k,d,d} \end{aligned}$$

and $\mathbf{R}_k^{-1}$ has entries $[\mathbf{R}_k^{-1}]_{d,d'} := \tilde{r}_{k,d,d'}$. The solution to (4.8) is given by (see e.g., [110])

$$o_{k,d}^{(j)} = \max \left\{ \left| \gamma_{k,d}^{(j)} \right| - \lambda_{k,d}, 0 \right\} \; \text{sign} \; \left(\gamma_{k,d}^{(j)} \right).$$

After J iterations, one sets $\hat{\mathbf{o}}_k = \mathbf{o}_k^{(J)}$. Global convergence of the iterates $o_k^{(J)}$ as J increases to the solution of (4.6) is guaranteed from the results in [113]. In practice however, a fixed small J will suffice.

4.2.3 AD-MoM based OR-RLS

To decouple the non-differentiable term of the cost in (4.6) from the differentiable one, consider introducing the auxiliary variable $\mathbf{c}_k$, and re-writing (4.6) in constrained form as

$$\begin{aligned} [\hat{\mathbf{o}}_k, \hat{\mathbf{c}}_k] \;\; =& \;\; \arg \min_{\mathbf{o}_k, \mathbf{c}_k} \left[\frac{1}{2} \| \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1} - \mathbf{o}_k \|_{\mathbf{R}_k^{-1}}^2 + \lambda_k \| \mathbf{c}_k \|_1 \right] \\ & \text{subject to} \quad \mathbf{o}_k = \mathbf{c}_k. \end{aligned}$$

The augmented Lagrangian thus becomes

$$L(\mathbf{o}_k, \mathbf{c}_k, \boldsymbol{\eta}) = \frac{1}{2}\|\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1} - \mathbf{o}_k\|^2_{\mathbf{R}_k^{-1}} + \lambda_k\|\mathbf{c}_k\|_1 + \boldsymbol{\eta}^T(\mathbf{o}_k - \mathbf{c}_k) + \frac{\kappa}{2}\|\mathbf{o}_k - \mathbf{c}_k\|_2^2$$

where $\boldsymbol{\eta}$ is the Lagrange multiplier, and κ is a positive constant controlling the effect of the quadratic term introduced to ensure strict convexity of the cost. After initializing with $\mathbf{c}_k^{(0)} = \boldsymbol{\eta}^{(0)} = \mathbf{0}$, AD-MoM recursions per iteration j amount to

$$\mathbf{o}_k^{(j)} = (\mathbf{R}_k^{-1} + \kappa\mathbf{I})^{-1}\left(\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1}) + \kappa\mathbf{c}_k^{(j-1)} + \boldsymbol{\eta}^{(j-1)}\right) \tag{4.9a}$$

$$c_{k,d}^{(j)} = \max\left\{\left|o_{k,d}^{(j)} + \frac{\mu_d^{(j)}}{\kappa}\right| - \frac{\lambda_k}{\kappa}, 0\right\}\text{sign}\left(o_{k,d}^{(j)} + \frac{\mu_d^{(j)}}{\kappa}\right), \quad \forall d = 1, \ldots, D_y \tag{4.9b}$$

$$\boldsymbol{\eta}^{(j)} = \boldsymbol{\eta}^{(j-1)} + \kappa(\mathbf{o}_k^{(j)} - \mathbf{c}_k^{(j)}). \tag{4.9c}$$

After J iterations, set $\hat{\mathbf{o}}_k = \mathbf{o}_k^{(J)}$. Global convergence of the iterates $o_k^{(J)}$ as J increases to the solution of (4.6) is guaranteed from the results in [13, p. 256]. In practice however, a fixed small J will suffice. Note also that the matrix $\mathbf{R}_k^{-1}$ needs to be computed only once, and used throughout the iterations.

4.2.4 Parameter selection

Successful performance of OR-RLS hinges on proper selection of the sparsity-enforcing coefficients λ_k. Too large a value for λ_k brings OR-RLS back to the ordinary RLS, while too small a value for λ_k creates many spurious outliers and degrades OR-RLS performance. Suppose that λ_k is constant with respect to k; that is, let $\lambda_k = \lambda_y$ for all $k \geq 1$.

Judicious tuning of λ_y has been investigated for the robust smoothing problem and two criteria for the optimal selection of λ_y have been proposed [41]. However, none of these algorithms can be run online. Besides, (4.6) should be solved for many different values of λ_y per time instant k. While these conditions are satisfied for the offline fixed-interval smoother of [41], this is hardly justified for the OR-RLS.

Table I. OR-RLS Algorithm
Initialization. Collect measurements $\mathbf{y}_k$ for $k = 1, \ldots, k_0$. Solve (4.10) to obtain $\hat{\mathbf{x}}_{k_0}$.
Use either of the two methods in [41] to determine the best $\lambda_k = \lambda_y,\ \forall\ k$
Repeat for $k \geq k_0 + 1$
Solve (4.6) to obtain $\hat{\mathbf{o}}_k$. Use CD iterations in (4.8) or AD-MoM iterations in (4.9).
Run RLS in (4.3) for the outlier compensated measurements $\mathbf{y}_k - \hat{\mathbf{o}}_k$ to obtain $\hat{\mathbf{x}}_k$.
End for

As a remedy, it is recommended to select λ_y during the initialization phase of OR-RLS and retain its value for all future time instants. OR-RLS initialization is carried out by solving the offline benchmark for $k = 1$ up to time instant $k = k_0$; that is

$$[\hat{\mathbf{x}}_{k_0}, \hat{\mathbf{o}}_{1:k_0}] = \arg \min_{\mathbf{x}_{k_0}, \mathbf{o}_{1:k_0}} \sum_{i=1}^{k_0} \beta^{k_0 - i} \left[\frac{1}{2} \|\mathbf{y}_i - \mathbf{H}_i \mathbf{x}_{k_0} - \mathbf{o}_i\|^2_{\mathbf{R}_i^{-1}} + \lambda_y \|\mathbf{o}_i\|_1 \right]. \tag{4.10}$$

Note that (4.10) can be solved via SeDuMi. Therefore, at time k_0 either one of the two criteria in [41] is applied to (4.10) to find best value for λ_y. Then, (4.10) is solved for that λ_y and a corresponding $\hat{\mathbf{x}}_{k_0}$ is obtained. Then, one proceeds to time $k_0 + 1$ and runs the OR-RLS algorithm as was set forth in the previous subsections. Table I provides a summary of OR-RLS.

4.3 Simulations

A setup with $D_x = 20$ and $D_y = 10$ is considered. In the static case, $\mathbf{x}_k = \mathbf{x},\ \forall\ k$ is selected as an all one vector added to a zero-mean, unit-variance Gaussian perturbation. The regressor entries $H_{k,ij}$ are chosen as independent zero-mean Gaussian distributed random variables with variance σ_h^2 tuned to achieve an SNR $= 2D_x\sigma_h^2$ of 2. Per time step, ambient (background) noise is taken to be highly correlated zero-mean Gaussian, simulated as the

output of an auto-regressive (AR) filter with a pole at 0.95 to a unit variance white Gaussian input. Background noise is independent from each time instant to the next. It is assumed that measurements are contaminated by impulsive noise, which generates outliers. When an outlier occurs, which has a 20% chance of happening, the corresponding observation is drawn from a uniform distribution independent of all the other variables with variance equal to $20,000$. The way outliers are generated differs from the model assumed in (4.1). This serves to signify the universality of the proposed algorithm as it demonstrates its operability even when the outliers are generated differently than modeled. The performance criterion is the root mean square error RMSE $:= \|\hat{\mathbf{x}}_k - \mathbf{x}_k\|_2$. Furthermore, $k_0 = 20$ and k ranges from 1 to $1,000$. It is assumed that the percentage of outliers is known to all estimators. OR-RLS utilizes this knowledge by using the corresponding criterion from [41] to tune the parameter λ_y.

4.3.1 Stationary scenario

Fig. 4.1 depicts RMSE versus time for a sample realization. RLS is initialized with a batch LS solution. As can be seen, RLS performs poorly in this case, while OR-RLS performs satisfactorily. Note that initialization with the batch offline estimator in (4.4) greatly improves the accuracy of OR-RLS. Different renditions of OR-RLS, which result from the way (4.6) is solved, are also compared. When (4.6) is solved exactly using SeDuMi the best performance is obtained. However, this is not desirable from a computational perspective and low-complexity CD iterations replace SeDuMi in practice. The performance of CD for $10, 20, 50$ and 100 iterations are depicted. With 100 CD iterations the performance of OR-RLS comes very close to the SeDuMi based one. Fig. 4.2 depicts RMSE for AD-MoM based OR-RLS. With 100 AD-MoM iterations, estimation performance comes close to that of SeDuMi.

OR-RLS with 100 CD iterations is compared against the robust RLS algorithm of [97]

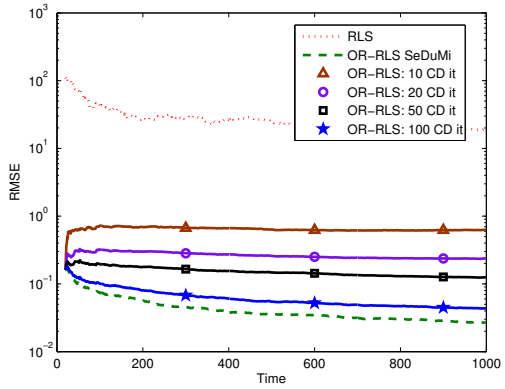

Figure 4.1: OR-RLS and its coordinate descent based implementation.

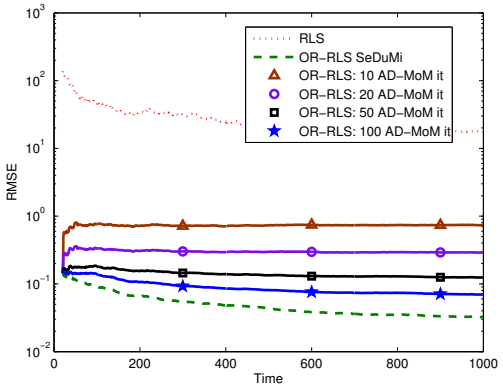

Figure 4.2: OR-RLS and its AD-MoM based implementation.

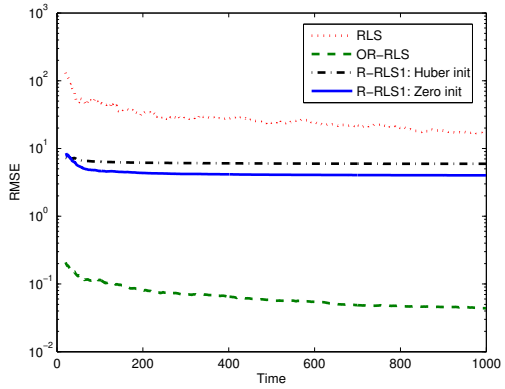

Figure 4.3: Comparison between R-RLS1 in [97] and OR-RLS.

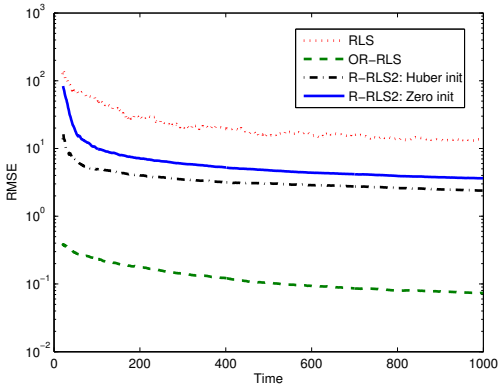

Figure 4.4: Comparison between R-RLS2 in [14] and OR-RLS.

dubbed R-RLS1 and that of [14] dubbed R-RLS2 in Figs. 4.3 and 4.4, respectively. Both of these approaches are proposed for white ambient noise, and do not take noise correlations into account. While one can pre-whiten the data by multiplying with $\mathbf{R}_k^{-1}$ and then apply R-RLS1 or R-RLS2, this spreads the outliers to all non-contaminated measurements and

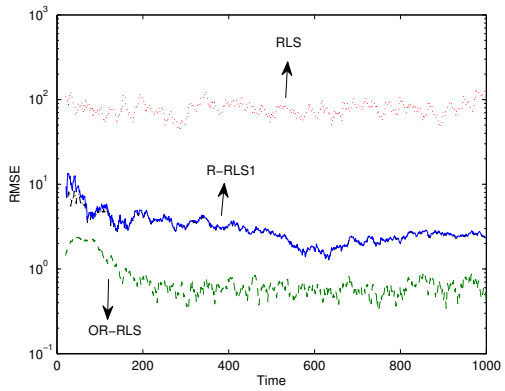

Figure 4.5: Comparison between R-RLS1 in [97] and OR-RLS for the non-stationary setting.

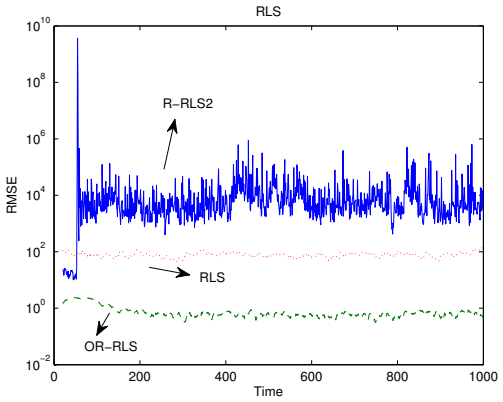

Figure 4.6: Comparison between R-RLS2 in [14] and OR-RLS for the non-stationary setting.

the resulting performance will be very poor and similar to RLS. As simulations showed, the better option is to ignore noise correlations which prevents outliers from spreading. This case has been considered in the simulations. To initialize R-RLS1 and R-RLS2 from $k = 1$ up to $k = k_0 = 20$, two different approaches are pursued: i) solving the batch Huber cost via iteratively re-weighted LS (IRLS); and ii) starting from the all zero point and then running R-RLS1 or R-RLS2. The performance of R-RLS1 is depicted in Fig. 4.3. While better than RLS, R-RLS1 still performs much worst than OR-RLS because R-RLS1 can not account for noise correlations. The two initializations perform almost similarly in this case. Fig. 4.4 depicts the performance of R-RLS2 versus OR-RLS. Huber initialization provides a considerable improvement in this case. Still, OR-RLS yields a noticeable improvement over R-RLS2.

4.3.2 Non-stationary scenario

For the non-stationary case, all parameters are selected similar to the stationary case. In addition, $\mathbf{x}_1$ is selected similar to the stationary case. To obtain $\mathbf{x}_2$, each entry of $\mathbf{x}_1$ is passed through a parallel AR 1 filter with a pole at 0.95 whose input is zero-mean unit-variance Gaussian noise. This procedure is repeated for all $k > 2$ also. As a result, $\mathbf{x}_k$ is slowly varying with time. Figs. 4.5 and 4.6 depict the RMSE of OR-RLS versus RLS, R-RLS1 of [97], and R-RLS2 of [14]. It can be seen that OR-RLS significantly outperforms all other three algorithms.

4.4 Summary

A robust RLS algorithm was developed capable of simultaneously handling measurement outliers and accounting for possibly correlated nominal noise. To achieve these desirable properties, outliers were treated as nuisance variables to be estimated jointly with the state. Identifiability was ensured by regularizing the LS cost with the ℓ_1-norm of the outlier variables. The resulting optimization problem, dubbed as offline benchmark, was too complex to be solved at each time step. For low-complexity implementation, a sub-optimal online algorithm was derived which involved only simple closed-form updates per time step. A combination of initialization with the offline benchmark followed by online updates was advocated as the OR-RLS algorithm. Numerical tests demonstrated improved performance of OR-RLS compared to RLS and state-of-the-art robust RLS renditions.

Chapter 5

Set-membership constrained particle filter: Distributed adaptation for sensor networks

Target tracking is investigated using particle filtering of data collected by distributed sensors. In lieu of a fusion center, local measurements must be disseminated across the network for each sensor to implement a centralized particle filter (PF). However, disseminating raw measurements incurs formidable communication overhead as large volumes of data are collected by the sensors. To reduce this overhead and thus enable distributed PF implementation, the present chapter develops a set-membership constrained (SMC) PF approach that: i) exhibits performance comparable to the centralized PF; ii) requires only communication of particle weights among neighboring sensors; and iii) can afford both consensus-based and incremental averaging implementations. These attractive attributes are effected through a novel adaptation scheme, which is amenable to simple distributed implementation using min- and max-consensus iterations. The resultant SMC-PF exhibits high gain over the

bootstrap PF when the likelihood is peaky, but not in the tail of the prior. Simulations corroborate that for a fixed number of particles, and subject to peaky likelihood conditions, SMC-PF outperforms the bootstrap PF, as well as recently developed distributed PF algorithms, by a wide margin.

5.1 Preliminaries and problem statement

Consider a network of N sensors distributed randomly on a field measuring their distance and/or bearing from a moving object. At time k, the object's state vector $\mathbf{x}_k \in \mathbb{R}^{d_x}$ evolves as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_k)$$

where $\mathbf{w}_k$ denotes the state noise of known distribution, which dictates the predictor density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Sensor n measures the vector

$$\mathbf{y}_k^n = \mathbf{h}^n(\mathbf{x}_k) + \mathbf{v}_k^n$$

where $\mathbf{v}_k^n \in \mathbb{R}^{d_y}$ stands for measurement noise of known distribution, giving rise to the likelihood $p(\mathbf{y}_k^n|\mathbf{x}_k)$. Functions $\mathbf{f}$ and $\mathbf{h}$ are generally nonlinear, and $\mathbf{v}_k^n$ is assumed independent across sensors. The network is connected, meaning that there is a (perhaps multi-hop) path connecting any two sensors.

Based on its own measurements $\{\mathbf{y}_k^n\}$ as well as those of others percolated through inter-sensor communications, sensor n wishes to obtain the MMSE optimal estimate of the state, namely the posterior mean $\hat{\mathbf{x}}_k := E[\mathbf{x}_k|\mathbf{y}_{1:k}]$, and its error covariance $\hat{\mathbf{C}}_k := E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T|\mathbf{y}_{1:k}]$, where $\mathbf{y}_{1:k} := [\mathbf{y}_1^T, \ldots, \mathbf{y}_k^T]^T$ and $\mathbf{y}_k := [(\mathbf{y}_k^1)^T, \ldots, (\mathbf{y}_k^N)^T]^T$. Treating these estimates together for brevity, the goal is to form in a distributed fashion the MMSE estimate of a function $\phi(\mathbf{x}_k) : \mathbb{R}^{d_x} \to \mathbb{R}$, given by

$$I_\phi(k) := E[\phi(\mathbf{x}_k)|\mathbf{y}_{1:k}] = \int \phi(\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k}) d\mathbf{x}_k \ . \tag{5.1}$$

Clearly, (5.1) yields as a special case[1] $\hat{\mathbf{x}}_k$ when $\phi(\mathbf{x}_k) := \mathbf{x}_k(i)$, with $\mathbf{x}_k(i)$ denoting the ith entry of $\mathbf{x}_k$; and also its error covariance entries $\hat{\mathbf{C}}_k(i,j)$ when $\phi(\mathbf{x}_k) := [\mathbf{x}_k(i) - \hat{\mathbf{x}}_k(i)][\mathbf{x}_k(j) - \hat{\mathbf{x}}_k(j)]$.

Since the likelihood and the predictor density are known, Bayes' rule allows in principle for recursive evaluation of the wanted posterior as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{\int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k}\ , \quad p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}. \tag{5.2}$$

Unfortunately, nonlinearity (and hence non-Gaussianity) of the processes involved render it impossible to evaluate exactly the integrals in (5.2). Among the available approximations, a popular one pursued here relies on particle filtering (PF) – the sequential version of importance sampling (IS) [15, Chapter 11].

IS refers to the process of estimating $E_{q_t}[\phi(\mathbf{x})]$, where $\mathbf{x}$ is distributed according to a target (but inconvenient to sample from) density $q_t(\mathbf{x})$, based on samples (a.k.a. particles) $\{\mathbf{x}^{(m)}\}_{m=1}^M$ drawn from a surrogate density $q_{IS}(\mathbf{x})$, which is selected so that samples from it are "of importance" to capture $q_t(\mathbf{x})$ too. The resultant consistent estimate is a weighted average (with scale-invariant weights) given by

$$\hat{I}_\phi := \sum_{m=1}^M \bar{w}^{(m)}\phi(\mathbf{x}^{(m)}), \quad \bar{w}^{(m)} = \frac{w^{(m)}}{\sum_{m=1}^M w^{(m)}}, \quad w^{(m)} = \frac{q_t(\mathbf{x}^{(m)})}{q_{IS}(\mathbf{x}^{(m)})},\ \forall m = 1,\ldots,M\ . \tag{5.3}$$

Since densities can be expressed via moments, it turns out the same IS weights can be employed to estimate q_t as $\hat{q}_t(\mathbf{x}) = \sum_{m=1}^M \bar{w}^{(m)}\delta(\mathbf{x} - \mathbf{x}^{(m)})$, where $\delta(.)$ denotes Dirac's delta. With identical IS weights one can also estimate marginals of q_t. As far as performance, it is known that the variance of $\hat{I}_\phi$ in (5.3) is minimized when the surrogate is selected as $q_{IS}(\mathbf{x}) \propto |\phi(\mathbf{x}) - E_{q_t}[\phi(\mathbf{x})]|q_t(\mathbf{x})$ [55]. This choice is obviously challenging to sample from,

[1] Likewise, any moment and thus the posterior density can be sequentially estimated, in par with the goal of PF.

and also ϕ-dependent. The ϕ-independent selection ($q_{IS} \propto q_t$) on the other hand, emerges from a tractable approximation of the minimum IS variance given by [81]

$$\mathrm{Var}_{IS} := E\left[\left(\hat{I}_\phi - E_{q_t}[\phi(\mathbf{x})]\right)^2\right] \;\approx\; (1/M)\,\mathrm{Var}_{q_t}\left(\phi(\mathbf{x})\right) E_{q_{IS}}\left[\left(\frac{q_t(\mathbf{x})}{q_{IS}(\mathbf{x})}\right)^2\right] . \tag{5.4}$$

Returning to PF, online estimation of $I_\phi(k)$ in (5.1) amounts to sequential IS with $q_t(\mathbf{x}_k) := p(\mathbf{x}_k|\mathbf{y}_{1:k})$. In fact, it is convenient to select as target density the joint posterior $q_t(\mathbf{x}_k) := p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, bearing in mind that identical weights can be employed to estimate the marginal density of interest. With regards to the IS density $q_{IS}(\mathbf{x}_{0:k})$, consider the following convenient factorization of the joint posterior

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{\pi(\mathbf{x}_k|\mathbf{y}_k,\mathbf{x}_{k-1})} \times \frac{\pi(\mathbf{x}_k|\mathbf{y}_k,\mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \tag{5.5}$$

where π here denotes an arbitrary density function (thus integrates to 1), for which both ratios in the right hand side (r.h.s.) of (5.5) are bounded away from infinity.

Suppose that at time step k, the particles and weights $\{\bar{w}_{k-1}^{(m)}, \mathbf{x}_{0:k-1}^{(m)}\}_{m=1}^M$ are available from step $k-1$. With $q_t(\mathbf{x}_{0:k}) := p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ and $q_{IS}(\mathbf{x}_{0:k}) := \pi(\mathbf{x}_k|\mathbf{y}_k,\mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$, the IS scheme at time k draws samples $\{\mathbf{x}_{0:k}^{(m)}\}_{m=1}^M$ from $q_{IS}(\mathbf{x}_{0:k})$, forms un-normalized weights $w_k^{(m)} = p(\mathbf{y}_k|\mathbf{x}_k^{(m)})p(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})\ /\pi(\mathbf{x}_k^{(m)}|\mathbf{y}_k,\mathbf{x}_{k-1}^{(m)})$ as well as their normalized versions $\{\bar{w}_k^{(m)}\}_{m=1}^M$, and relies on the set $\{\bar{w}_k^{(m)}, \mathbf{x}_{0:k}^{(m)}\}_{m=1}^M$ to estimate $I_\phi(k)$ as in (5.3). To draw samples from the chosen $q_{IS}(\mathbf{x}_{0:k})$ requires the estimate

$$\hat{p}(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) = \sum_{m=1}^M \bar{w}_{k-1}^{(m)} \delta(\mathbf{x}_{0:k-1} - \mathbf{x}_{0:k-1}^{(m)}) . \tag{5.6}$$

With (5.6) available from time step $k-1$, the IS density approximant to sample from at step k is

$$\pi(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{y}_k)\hat{p}(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) = \sum_{m=1}^M \bar{w}_{k-1}^{(m)} \pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)},\mathbf{y}_k)\delta(\mathbf{x}_{0:k-1} - \mathbf{x}_{0:k-1}^{(m)}) \tag{5.7}$$

which amounts to re-sampling from $\{\bar{w}_{k-1}^{(m)}, \mathbf{x}_{0:k-1}^{(m)}\}$, followed by particle augmentation by sampling from $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$. While the product $\pi(\mathbf{x}_k|\mathbf{y}_k, \mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$ is the IS density at step k, the factor $\pi(\mathbf{x}_k|\mathbf{y}_k, \mathbf{x}_{k-1})$ used to augment the re-sampled particles, is often referred to as the IS density. Note that there is no need to store the whole trajectory $\{\mathbf{x}_{0:k}^{(m)}\}_{m=1}^M$, but only the current set $\{\mathbf{x}_k^{(m)}\}_{m=1}^M$.

Summarizing, after initialization the PF implements these steps per time instant k; see e.g., [35].

S1) Re-sample from the particles and weights $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$ available from time $k-1$ (cf. (5.6), (5.7));

S2) Draw the new sample $\mathbf{x}_k^{(m)} \sim \pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$ from the chosen (augmenting IS) density;

S3) Find weights through Bayes' rule, and normalize them to sum up to one, as (cf. (5.3), (5.5))

$$w_k^{(m)} = \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(m)})p(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)}, \qquad \bar{w}_k^{(m)} = \frac{w_k^{(m)}}{\sum_{m=1}^M w_k^{(m)}}\ ; \tag{5.8}$$

S4) Form $\hat{I}_\phi(k) := \sum_{m=1}^M \bar{w}_k^{(m)} \phi(\mathbf{x}_k^{(m)})$ as in (5.3), and output current state estimate and its covariance.

Selecting π is a performance-critical issue of the PF algorithm. A popular but not necessarily efficient choice is $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) := p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, which corresponds to the so-termed bootstrap PF (B-PF). Setting the IS density equal to the prior, the B-PF weights in (5.8) become $w_k^{(m)} = p(\mathbf{y}_k|\mathbf{x}_k^{(m)})$. The attractive feature of B-PF is its simplicity, since it is easy to generate particles from the prior, and have their weights simply given by the likelihood. The optimum π in S2 in the sense of minimizing the variance of $w_k^{(m)}$ is $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) := p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$, and amounts to choosing the IS density equal to true posterior [36]. Since it is impossible to draw samples from the true posterior, sub-optimal choices of π accounting for the new data $\mathbf{y}_k$ in the IS density are well motivated. They are referred to as data-adapted (or simply adapted) densities, to be differentiated from

non-adapted choices such as the one in B-PF. The significance of adaptation in the present framework will become clear in the next section.

The described PF algorithm is *centralized* because it requires knowledge of $\mathbf{y}_k$ to operate. Specifically, to generate particles in S2 and obtain weights in S3, it is necessary to disseminate $\mathbf{y}_k$ throughout the network (e.g., via flooding), which is impractical especially for a large number of sensors N, or, a large size measurement vector (d_y). To cope with this challenge, the ensuing section presents a *distributed* PF that requires exchanging only particle weights, while the follow-up section introduces a novel adaptation scheme based on *set-membership* for reducing the number of particles.

5.2 Distributed PF

At initialization ($k = 0$), each sensor draws new samples from $p(\mathbf{x}_0)$ and weighs them equally. Setting the seed of random number generators at all sensors to the same value ensures identical particles $\mathbf{x}_0^{(m)}$, $\forall m = 1, \ldots, M$ generated at all sensors. Next, recall that after step $k-1$, sensors have available $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^{M}$. Since the random number generators at all sensors are initialized with the same seed, re-sampled particles across all sensors will be identical; see also [27]. Thus, S1 can be performed locally per sensor provided that $\{\bar{w}_{k-1}^{(m)}, \mathbf{x}_{k-1}^{(m)}\}_{m=1}^{M}$ is known at all sensors. In the same spirit, S4 can be run locally provided that $\{\bar{w}_{k}^{(m)}, \mathbf{x}_{k}^{(m)}\}_{m=1}^{M}$ is commonly available to all sensors. If in addition a common non-adapted IS density is utilized (meaning $\pi(\cdot)$ is not dependent on $\mathbf{y}_k$), then S2 can also be run locally at each sensor. In a nutshell, it is possible to implement S1, S2, and S4 in a distributed fashion.

Focusing on S3, one can invoke the noise independence across sensors, and take loga-

rithms on both sides of (5.8) to obtain

$$\log\left(w_k^{(m)}\right) = \log\left(p(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})\right) + \sum_{n=1}^{N} \log\left(p(\mathbf{y}_k^n|\mathbf{x}_k^{(m)})\right) - \log\left(\pi(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)\right). \quad (5.9)$$

Because sensors have available the particles $\{\mathbf{x}_{k-1}^{(m)}\}_{m=1}^M$ at time $k-1$ from S1 as well as the newly generated ones $\{\mathbf{x}_k^{(m)}\}_{m=1}^M$ from S2, all terms except the sum on the r.h.s. of (5.9) are locally known when $\pi(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) = \pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$. This sum can become available per sensor either through incremental averaging or via consensus averaging, as described next.

Consider that each sensor n in the network has a scalar ψ_n. Through collaborative exchanges all sensors wish to find the sample mean $\bar{\psi}_N := (1/N)\sum_{n=1}^N \psi_n$, when no fusion center is available to receive and centrally average $\{\psi_n\}_{n=1}^N$. For the problem at hand, $\psi_n = \log\left(p(\mathbf{y}_k^n|\mathbf{x}_k^{(m)})\right)$. Operation without fusion centers is desirable for scalability, and also because isolated points of failure are avoided.

Incremental averaging relies on passing partial sums over a Hamiltonian path, which goes through every sensor exactly once. With sensors indexed by the order they appear in this Hamiltonian path, the algorithm commences with sensor 1 transmitting ψ_1 to sensor 2, which forms $\psi_1 + \psi_2$ and transmits it to sensor 3. Likewise, sensor n receives the partial sum $\psi_1 + \cdots + \psi_{n-1}$ from sensor $n-1$, adds ψ_n to it and communicates the sum to sensor $n+1$. The desired $\bar{\psi}_N$ formed at sensor N is then percolated through the Hamiltonian path in order for all sensors to have a copy of the sample average. The incremental averaging scheme converges in finite time. But finding a Hamiltonian path is NP-hard, and requires perfect knowledge of the communication graph at every sensor [117]. Furthermore, the algorithm is not robust because a new Hamiltonian cycle must be established each time a sensor fails.

Alternatively, it is possible to make the sum in (5.9) available per sensor via consensus averaging, see e.g., [118]. Here sensors do not need to know the communication graph or establish Hamiltonian paths, but only need to communicate with their immediate neighbors.

The desired $\bar{\psi}_N$ is obtained per sensor iteratively. After iteration $i-1$, sensor n broadcasts $\psi_n(i-1)$ to its neighbors. Having received $\{\psi_\ell(i-1)\}_{\ell\in\mathcal{N}(n)}$ from its one-hop neighbors denoted by the set $\mathcal{N}(n)$, sensor n updates its iterate as

$$\psi_n(i) = \psi_n(i-1) + \mu \sum_{\ell\in\mathcal{N}(n)} [\psi_\ell(i-1) - \psi_n(i-1)], \qquad \psi_n(0) = \psi_n$$

where μ is a constant stepsize. If μ is chosen small enough, and the communication graph remains connected, the iterates $\psi_n(i)$ converge asymptotically (as $i \to \infty$) to the desired sample mean [118]; that is, $\lim_{i\to\infty} \psi_n(i) = \bar{\psi}_N, \ \forall n$. Consensus-averaging is robust to sensor failures so long as the network remains connected. A remark is now due on the practical operation of consensus averaging.

Remark 7. Since only a finite number of consensus-averaging iterations can be afforded in practice, some residual error will be inevitable. To mitigate these residual weight mismatches, one can apply min- (max-) consensus iterations to form the $\min_n \ \psi_n(i)$ (correspondingly $\max_n \ \psi_n(i)$), after the consensus-averaging iterations are completed. In the simulations, both min- and max- consensuses will be run on the $\psi_n(i)$s and the final weight value will be set equal to their average.

Clearly from (5.9), either consensus or incremental averaging should be run per particle $\mathbf{x}_k^{(m)}$. Hence, the number of particles increases the communication overhead of S3 considerably, because M such consensus or incremental operations must be run in parallel. It will become evident that this requirement also limits the options for distributing S2, when adapted IS densities are employed. As mentioned earlier, performing S2 in a distributed fashion is possible when using a non-adapted IS density, such as the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ in B-PF, which does not require knowledge of $\mathbf{y}_k$. Such a selection of π however, requires a large number of particles to cope with particle depletion [22], which in turn increases the communication cost of S3 and renders the distributed algorithm inefficient. Indeed, recall from (5.4) that for M sufficiently large the error variance per PF step is $E[(\hat{I}_\phi(k) - E[\phi(\mathbf{x}_k)|\mathbf{y}_{1:k}])^2] \approx C/M$.

Constant C, and the threshold value for M above which this per-step PF variance decreases as $\mathcal{O}(M^{-1})$, depend on the similarity between the true posterior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$, and the selected IS density $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$ [32,36]. To ensure that p and π are "similar," information in the observations $\mathbf{y}_k$ must be exploited by the selected π.

This certainly advocates data adaptation when selecting the IS density. In non-adapted PF renditions, such as the B-PF, many particles $\mathbf{x}_k^{(m)}$ receive almost zero weights in S3, and have to be discarded since they are generated from a density far different from the true posterior. The latter leads to particle depletion, which compromises performance. In contrast, data-adapted IS densities ensure that the new measurements $\mathbf{y}_k$ are accounted for during the particle generation phase, and thus fewer particles are needed to approximate the posterior accurately. Therefore, adaptation brings significant benefits because it reduces the number of required particles, which translates to a commensurate reduction in communication overhead of the proposed distributed PF. From the large list of sub-optimal adaptation methods for centralized PF mentioned in the Introduction, none of them is amenable to efficient distributed implementation.

To bypass this challenge, the next section introduces a novel adaptation method, which renders the distributed implementation of S2 affordable by reducing the overhead of inter-sensor communications.

5.3 Set-membership based adaptation

The key idea behind the proposed low-overhead distributed adaptation scheme is to approximate the posterior density $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k)$ with an appropriately scaled (distorted) version of the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. To this end, suppose temporarily that a *local* set $\mathcal{E}_k^n$ can be constructed per sensor n from the domain of $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$ to capture most of its probability mass. And based on these local sets, consider the *global* set $\mathcal{E}_k := \bigcap_n \mathcal{E}_k^n$, over which *all*

sensors have high local posterior masses. With $\mathbb{1}_{\{.\}}$ representing the indicator function, the proposed adapted IS density is

$$\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}) := \frac{\alpha_k \mathbb{1}_{\{\mathbf{x}_k \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\mathbf{x}_k \notin \mathcal{E}_k\}}}{c_k} \, p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{5.10}$$

where $\beta_k \ll \alpha_k$ (w.l.o.g. $\alpha_k = 1$), and c_k is a normalization constant so that $\pi_{\mathcal{E}_k}(.)$ integrates to 1.

Before describing the process for selecting $\{\mathcal{E}_k^n\}_{n=1}^N$, which define the global set $\mathcal{E}_k$, the ensuing subsection explains why $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1})$ offers an attractive approximation of the global posterior.

5.3.1 Posterior density approximation

Consider that each sensor n has determined the local set $\mathcal{E}_k^n$ containing most of the probability mass of its local posterior $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$, and approximate the latter using $\pi_{\mathcal{E}_k^n}(\mathbf{x}_k|\mathbf{x}_{k-1})$. Since the noise across sensors is independent, the global posterior can be expressed as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k) &\propto \left[\prod_{n=1}^N \frac{p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)}{p(\mathbf{x}_k|\mathbf{x}_{k-1})}\right] p(\mathbf{x}_k|\mathbf{x}_{k-1}) \\ &\approx \left[\prod_{n=1}^N \frac{\pi_{\mathcal{E}_k^n}(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{x}_k|\mathbf{x}_{k-1})}\right] p(\mathbf{x}_k|\mathbf{x}_{k-1}) \approx \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}) \end{aligned} \tag{5.11}$$

where the product $\prod_{n=1}^N \frac{\pi_{\mathcal{E}_k^n}(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{x}_k|\mathbf{x}_{k-1})}$ is approximated using the function $\left(\alpha_k \mathbb{1}_{\{\mathbf{x}_k \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\mathbf{x}_k \notin \mathcal{E}_k\}}\right)/c_k$, which is valid for $\beta_k \ll 1$ since $\mathcal{E}_k := \bigcap_n \mathcal{E}_k^n$. Compared to the prior itself, the scaled prior in (5.10), later referred to as the set-membership approximation, can be much closer to the posterior, which is the density of interest; see also Fig. 5.1.

5.3.2 Local set selection

In view of (5.11), the local set $\mathcal{E}_k^n$ at sensor n must be constructed so that $\pi_{\mathcal{E}_k^n}(\mathbf{x}_k|\mathbf{x}_{k-1}) \approx p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$. Note though that it is impossible to specify directly the set $\mathcal{E}_k^n$ that contains

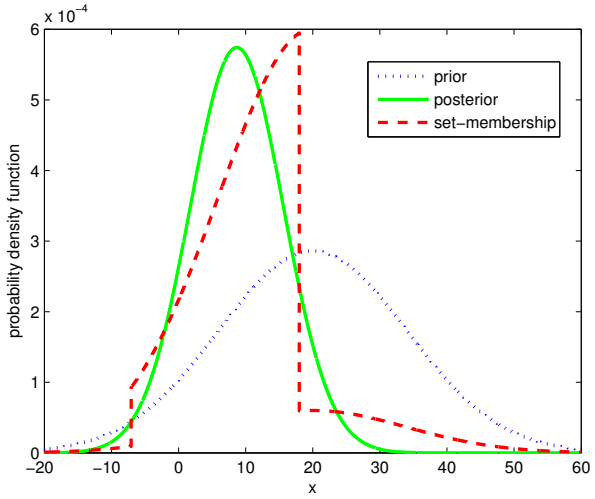

Figure 5.1: Similarity between scaled prior (set-membership approximation) and posterior densities

a high probability mass of $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$ because the latter is conditioned on $\mathbf{x}_{k-1}$, which is not available. One way around this obstacle is to construct a separate local $\mathcal{E}_k^n$ for every $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k^n)$ corresponding to each particle $m = 1, \ldots, M$. However, the computational burden of this construction defeats the purpose of introducing the local sets at the first place. (Indeed, instead of computing a single $\mathcal{E}_k^n$ per sensor which requires consensus on a single $\mathcal{E}_k$, multiple local sets necessitate consenting on one $\mathcal{E}_k$ per particle m, which increases the communication overhead prohibitively.)

The remedy at an intuitive level is to construct $\mathcal{E}_k^n$ so that it contains a high probability mass of $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$ on the *average* performed over $\mathbf{x}_{k-1}$. This motivates selecting $\mathcal{E}_k^n$ to contain a large probability mass of the density $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$. Such a choice is prudent because

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n) d\mathbf{x}_{k-1}$$

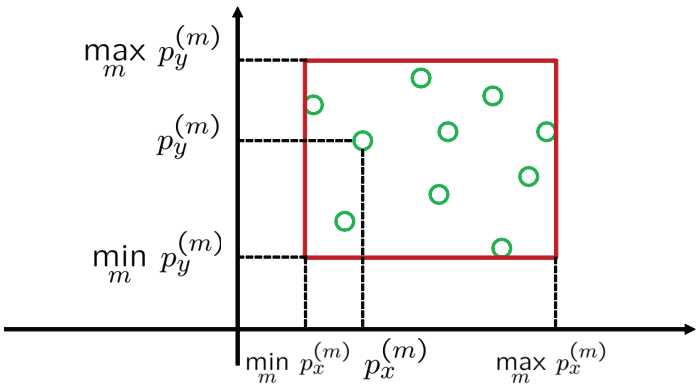

Figure 5.2: Particle bounding box approach

where $\mathbf{x}_{k-1}$ is averaged over the posterior $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$ that is estimated using particles and their weights at step $k-1$. Through this density, averaging out $\mathbf{x}_{k-1}$ takes into account not only the locally available current measurement $\mathbf{y}_k^n$, but also all past measurements $\mathbf{y}_{1:k-1}$ through $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$.

In the remainder of this subsection, three schemes will be developed to select local sets $\mathcal{E}_k^n$ containing most of the probability mass under $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$. Their efficacy will be tested via simulations.

Particle bounding box

The crux of this construction is reliance on local B-PF to form the corresponding local set $\mathcal{E}_k^n$. Toward this objective, consider (re-) sampling per sensor as in S1-S2 from the prior $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k^n) := p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, which is the IS density of B-PF. Since this local prior is non-adapted, it is expected that many less than M particles re-sampled as in S1-S2 will

fall in the region where the likelihood (and hence the local posterior) is high. Thus, each sensor must over-sample particles as in S1 by a factor $L > 1$ ($L \in \mathbb{N}$), so that a few of the LM particles will fall in the region that the set $\mathcal{E}_k^n$ is sought to capture. Recall also that with the same seed across sensors, these LM particles at all sensors will be identical. Once the local likelihood $p(\mathbf{y}_k^n|\mathbf{x}_k^{(m)})$ is weighted in, the sensors generate different weights, which along with the LM particles yield an estimate of $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$. Re-sampling M particles from the latter, and fitting an axes-aligned box around them yields the desired $\mathcal{E}_k^n$, which represents the region where $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$ contains most of its mass.

Letting $\{\bar{\tilde{\mathbf{x}}}_k^{n,(m)}\}_{m=1}^M$ denote these re-sampled particles, the axes-aligned bounding box describing the local set at sensor n at time k is specified by the following polyhedron:

$$\mathcal{E}_k^n := \left\{\mathbf{x} : \min_m \bar{\tilde{\mathbf{x}}}_k^{n,(m)} \preceq \mathbf{x} \preceq \max_m \bar{\tilde{\mathbf{x}}}_k^{n,(m)}\right\} := \left\{\mathbf{x} : \mathbf{x}_{k,\min}^n \preceq \mathbf{x} \preceq \mathbf{x}_{k,\max}^n\right\} \tag{5.12}$$

where the minimum, maximum, and inequalities ($\preceq$) should be understood component-wise. Fig. 5.2 depicts the min and max operations for $d_x = 2$.

In summary, to arrive at the set in (5.12) each sensor runs the following three local steps.

LS1) Over re-sample particles and weights $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$ to obtain $\{\tilde{\mathbf{x}}_{k-1}^{(m')}\}_{m'=1}^{LM}$ with $L \in \mathbb{N}$.

LS2) Generate LM new particles from the prior $\tilde{\mathbf{x}}_k^{(m')} \sim p(\mathbf{x}_k|\tilde{\mathbf{x}}_{k-1}^{(m')})$; form their weights using the local likelihood $\tilde{w}_k^{n,(m')} = p(\mathbf{y}_k^n|\tilde{\mathbf{x}}_k^{(m')})$; and normalize them to obtain $\bar{\tilde{w}}_k^{n,(m')}$.

LS3) Re-sample M particles from $\{\tilde{\mathbf{x}}_k^{(m')}, \bar{\tilde{w}}_k^{n,(m')}\}_{m'=1}^{LM}$, and construct $\mathcal{E}_k^n$ as in (5.12).

Note in closing that the weights in LS1 and LS2 are not used subsequently, since the aim here is not PF but construction of the local sets, which are specified only by the (min and max per entry) particles re-sampled at LS3. Fig. 5.3 depicts the particle bounding box approach for a sensor measuring its distance from the target.

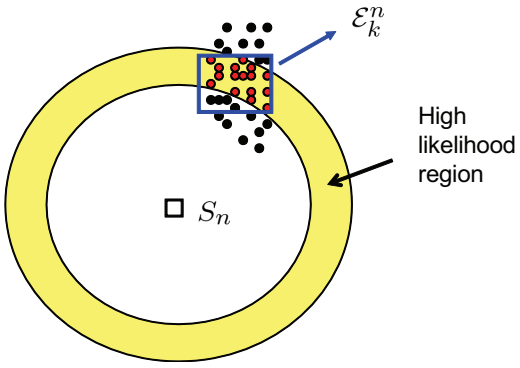

Figure 5.3: Graphical depiction of selecting the local set $\mathcal{E}_k^n$

UKF-based local set

This construction capitalizes on the unscented particle filter (UPF) [86], which enables fitting to the local posterior, namely $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$, a Gaussian density specified by its mean and covariance. Per time step k, the latter can be found through a UKF update. Given the obtained mean and covariance, each local set is constructed to encapsulate a prescribed amount of probability mass from the corresponding local posterior of interest. For the Gaussian approximating density, this set takes the form of an ellipsoid. The local set $\mathcal{E}_k^n$ is then specified as the smallest axes-aligned box containing this ellipsoid. Specifically, the following local steps are performed at each sensor n.

LS1′) Given the particles and weights $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$ from step $(k-1)$, find the mean and covariance matrix of the Gaussian density approximating the posterior at time step $(k-1)$, that is

$$\hat{\mathbf{x}}_{k-1} = \sum_{m=1}^{M} \bar{w}_{k-1}^{(m)} \mathbf{x}_{k-1}^{(m)}, \quad \hat{\mathbf{C}}_{k-1} = \sum_{m=1}^{M} \bar{w}_{k-1}^{(m)} (\mathbf{x}_{k-1}^{(m)} - \hat{\mathbf{x}}_{k-1})(\mathbf{x}_{k-1}^{(m)} - \hat{\mathbf{x}}_{k-1})^T \ .$$

LS2′) With input the estimates $\hat{\mathbf{x}}_{k-1}$ and $\hat{\mathbf{C}}_{k-1}$ found in LS1′, run a UKF prediction-correction iteration as in [76]. Incorporating the local measurement $\mathbf{y}_k^n$ in the correction

step, the UKF output yields the conditional mean $\hat{\mathbf{x}}_k^n$, and covariance matrix $\hat{\mathbf{C}}_k^n$ per sensor n at time step k.

LS3′) Based on $\hat{\mathbf{x}}_k^n$ and $\hat{\mathbf{C}}_k^n$, form $\mathcal{E}_k^n$ as the smallest axes-aligned box containing the ellipsoid

$$(\mathbf{x}_k - \hat{\mathbf{x}}_k^n)^T \left(\hat{\mathbf{C}}_k^n\right)^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k^n) \leq \eta^2 \tag{5.13}$$

where η is set depending on the probability mass the designer chooses to capture by the ellipsoid.

As an example, suppose one requires 99% of the probability mass to be contained in the ellipsoid. The linear transformation $\mathbf{x}_{\text{new}} = \left(\hat{\mathbf{C}}_k^n\right)^{-1/2} (\mathbf{x}_{\text{old}} - \hat{\mathbf{x}}_k^n)$ expresses the quadratic form in (5.13) as a sum of d_x independent, zero-mean, and unit-variance Gaussian-squared random variables, which amounts to a random variable distributed according to a centralized $\chi^2_{d_x}$ density with d_x degrees of freedom. Parameter η^2 is the point for which the integral of this density over $[0, \eta^2]$ equals 0.99.

To visualize the smallest axes-aligned box containing the ellipsoid, consider as an example the two-dimensional case ($d_x = 2$), illustrated graphically in Fig. 5.4. The smallest axes-aligned bounding box is completely determined by the four boundary points $x_{\min}, x_{\max}, y_{\min}$, and $y_{\max}$. These points are obtained by projecting the ellipsoid on the x- and y-axes. To generalize for any d_x and specify analytically the smallest axes-aligned box in LS3′, let $\mathbf{e}_d$ denote the elementary $d_x \times 1$ vector having its dth entry equal to 1, and all other entries equal to 0. Clearly, the projection of a point $\mathbf{x}$ on the ellipsoid over the dth side of the polyhedral box is given by the inner product $\mathbf{e}_d^T \mathbf{x}$. Hence, the minimum $[\mathbf{x}_k^n(d)]_{\min}$ along the dth coordinate axis can be obtained as the solution of the following constrained optimization problem:

$$\begin{cases} \min\limits_{\mathbf{x}} & \mathbf{e}_d^T \mathbf{x} \\ \text{subject to} & (\mathbf{x} - \hat{\mathbf{x}}_k^n)^T \left(\hat{\mathbf{C}}_k^n\right)^{-1} (\mathbf{x} - \hat{\mathbf{x}}_k^n) \leq \eta^2 \end{cases}$$

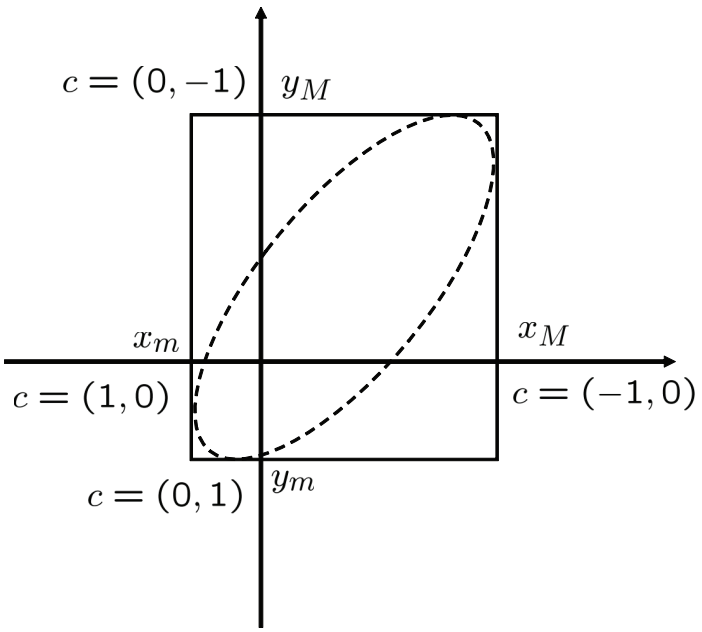

Figure 5.4: The smallest axis-aligned bounding box for a 2-D ellipsoid

This problem is convex, and admits the following closed-form solution

$$[\mathbf{x}_k^n(d)]_{\min} = \hat{\mathbf{x}}_k^n(d) - \eta \sqrt{\hat{\mathbf{C}}_k^n(d,d)} \; .$$

Clearly, solving the related optimization problem with $\mathbf{e}_d$ replaced by $-\mathbf{e}_d$ will yield the maximum $[\mathbf{x}_k(d)]_{\max}$ along the dth coordinate axis. Finding likewise the minima and maxima for all $d = 1, \ldots, d_x$, specifies the desired box.

Remark 8. While an EKF update can be employed instead of the UKF in LS2′, the superior performance of UKF confirmed by simulations suggests that not much is gained when adopting the EKF. Notwithstanding, UKF is not used here as a tracker, but only to specify the local sets $\{\mathcal{E}_k^n\}_{n=1}^N$.

UKF with Gaussian mixture model

When the local posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$ is multi-modal, UKF can not approximate it well with a single Gaussian bell, which is uni-modal. To cope with multi-modality, a GMM can be trained from the particles representing the prior, using the expectation-maximization (EM) algorithm; see e.g., [15, Chapter 9]. Then, the UKF-based algorithm of the uni-modal case can be applied to each mixture component separately. This amounts to one axes-aligned box per mixture component. The local set $\mathcal{E}_k^n$ is then constructed as the smallest box containing the union of these boxes. While the number of components (modes) can be preselected based on complexity considerations, it is also possible to fit a GMM with unknown model order. In this case, one of the criteria outlined in e.g., [106, Appendix C] can be utilized for model order determination.

5.3.3 Global set determination

Once sensor n processes its own $\mathbf{y}_k^n$ to obtain the local set $\mathcal{E}_k^n = \left\{\mathbf{x} : \mathbf{x}_{k,\min}^n \preceq \mathbf{x} \preceq \mathbf{x}_{k,\max}^n\right\}$, the global set $\mathcal{E}_k$ can be constructed as

$$\mathcal{E}_k := \bigcap_{n=1}^{N} \mathcal{E}_k^n = \left\{\mathbf{x} : \max_n \mathbf{x}_{k,\min}^n \preceq \mathbf{x} \preceq \min_n \mathbf{x}_{k,\max}^n\right\}. \tag{5.14}$$

Since boxes are used to represent local sets, distributing their intersection is very simple because it only requires finding minima and maxima of scalar quantities available per sensor; see also Fig. 5.5. Distinct from the distributed B-PF algorithm of Section 5.2 that requires consensus averaging iterations, constructing the global set $\mathcal{E}_k$ in a distributed fashion entails consenting on the minimum and maximum points of the local set boundaries.

It is possible to find, say these maximum points, using a max-consensus approach along the lines of the consensus averaging scheme outlined in Section 5.2. Except that instead of $\bar{\psi}_N$, sensors now wish to compute $\psi_{\max} := \max_n \psi_n$ in a distributed fashion. Clearly, one

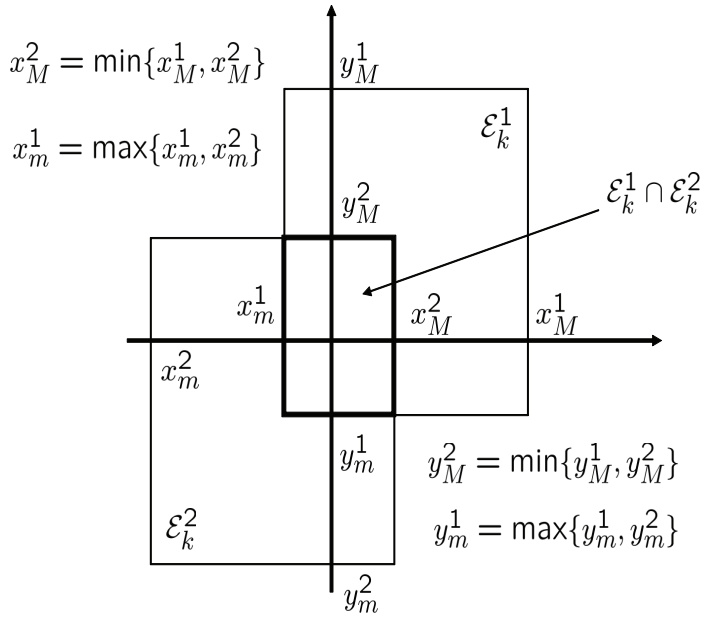

Figure 5.5: Intersection of two boxes amounts to computing minima and maxima

scalar $\psi_n = \mathbf{x}_{k,\min}^n(d)$ is involved here per coordinate d. In iteration i of the max-consensus algorithm, sensor n communicates $\psi_n(i-1)$ to its immediate neighborhood $\mathcal{N}(n)$, and updates the local auxiliary variables $\psi_n(i)$ as

$$\psi_n(i) = \max\{\psi_n(i-1), \max_{l\in\mathcal{N}(n)} \psi_l(i-1)\}, \qquad \psi_n(0) = \psi_n \ .$$

Most importantly when compared to consensus averaging, the iterates $\psi_n(i)$ converge to the exact $\psi_{\max}$ in finite time, and the number of required iterations does not exceed the diameter of the communication graph; see e.g., [28]. The min-consensus scheme operates similarly with the obvious substitutions.

Because min- and max-consensus algorithms converge in finite iterations bounded by the diameter of the communication graph, a distributed implementation becomes available at much lower communication cost compared to propagating (or performing consensus of)

raw measurements across sensors.

Remark 9. When the intersection $\mathcal{E}_k$ in (5.14) is empty, local sets are stretched out by a constant factor before re-computing the intersection. This is repeated until the process arrives at a non-empty intersection. Simulations indicate that two repetitions are typically sufficient, while the first intersection is non-empty most of the time.

5.3.4 Distributed SMC-PF

Once $\mathcal{E}_k$ is available to all sensors, samples are drawn per sensor (cf. S2) from the scaled prior density $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$. To generate these samples, rejection sampling (RS) is employed, see e.g., [15, Chapter 11], with IS density $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ and target density $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, upper-bounded as: $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) \leq (\alpha_k/c_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$. Specifically, samples are drawn first from the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$. Then, they are accepted with probability $c_k\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})/[\alpha_k p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})] = \mathbb{1}_{\{\mathbf{x}_k\in\mathcal{E}_k\}} + (\beta_k/\alpha_k)\mathbb{1}_{\{\mathbf{x}_k\notin\mathcal{E}_k\}}$. Therefore, if a sample belongs to $\mathcal{E}_k$, it is accepted with probability 1; otherwise, it is rejected with high probability. In other words, the accept probability satisfies $\beta_k/\alpha_k \ll 1$.

Now all pieces of the novel distributed SMC-PF can be put together. Sensors start with a common seed, and (re-) sample identical particles as per S1 locally. For S2, RS-based sampling from $\pi_{\mathcal{E}_k}(\cdot)$ is run also locally after $\mathcal{E}_k$ becomes available to every sensor via consensus on the set boundaries. Incremental or consensus averaging distributes the weights in S3, which is all every sensor needs to perform S4. Specifically for consensus averaging, one uses $\pi(.) = \pi_{\mathcal{E}_k}(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})$ in (5.9) to obtain the weights as

$$\log\left(w_k^{(m)}\right) \;=\; \sum_{n=1}^{N}\log\left(p(\mathbf{y}_k^n|\mathbf{x}_k^{(m)})\right) + \log\left(c_k^{(m)}\right) - \log\left(\alpha_k\mathbb{1}_{\{\mathbf{x}_k^{(m)}\in\mathcal{E}_k\}} + \beta_k\mathbb{1}_{\{\mathbf{x}_k^{(m)}\notin\mathcal{E}_k\}}\right). \tag{5.15}$$

The SMC-PF algorithm is summarized in Table II.

Remark 10. The novel approach is referred to as set membership constrained (SMC) PF

Table II. SMC-PF Algorithm

Initialization. Draw $\mathbf{x}_0^{(m)} \sim p(\mathbf{x}_0), \quad \forall m = 1, \ldots, M$. Set $\bar{w}_0^{(m)} = \frac{1}{M}$.

Repeat for time $k \geq 1$

Repeat for sensors $n = 1, \ldots, N$ (can be implemented in parallel)

Given $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^{M}$ and $\mathbf{y}_k^n$, run either LS1-LS3 or LS1′-LS3′ to obtain $\mathcal{E}_k^n$.

End for

Given $\{\mathcal{E}_k^n\}_{n=1}^{N}$, construct $\mathcal{E}_k$ as in (5.14) in a distributed fashion.

At each sensor run in parallel

Re-sample from $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^{M}$, [S1].

Draw the new sample $\mathbf{x}_k^{(m)} \sim \pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) := \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ [S2].

Update the weights as in (5.15) [S3]. Use incremental or consensus averaging.

Obtain $\{\mathbf{x}_k^{(m)}, \bar{w}_k^{(m)}\}_{m=1}^{M}$ at the output of S2-S3.

Form $\hat{I}_\phi(k) := \sum_{m=1}^{M} \bar{w}_k^{(m)} \phi(\mathbf{x}_k^{(m)})$ as the final estimate [S4].

End for

for two reasons: i) successive set intersections are computed, which is a trade-mark of set-membership approaches [102]; and, ii) rejection sampling relies on an accept-reject criterion assessing membership in the set $\mathcal{E}_k$.

Remark 11. (Communication Cost) Let κ denote the number of consensus averaging iterations in either B-PF or SMC-PF, and D_{graph} the diameter of the communication graph. Then, B-PF requires $\kappa M + 2D_{\text{graph}}M$ scalars to be communicated per time step and sensor, while SMC-PF requires $2d_x D_{\text{graph}} + \kappa M + 2D_{\text{graph}}M$ scalars; hence, for the same number of particles M, SMC-PF needs communication of $2d_x D_{\text{graph}}$ more scalars than B-PF. Note that κM denotes the number of scalars that are transmitted per time step and sensor for consensus-averaging iterations. Afterwards, min- and max- consensuses should be run, as a consequence of Remark 7, which requires communication of $2D_{\text{graph}}M$ scalars. Finally,

SMC-PF needs to run min- and max- consensuses to find the global set which requires transmission of $2d_x D_{\text{graph}}$ scalars.

5.4 Performance analysis

This section analyzes the performance of SMC-PF using the state estimator's mean-square error (MSE) as figure of merit. First, the MSE is expressed as the superposition of its minimum MSE (MMSE) value plus the PF-related error variance denoted by Var_{PF}. Subsequently, a per-step tractable approximation of the Var_{PF} is introduced to allow for comparing SMC-PF with the B-PF. This finite-sample approximate analysis is also complemented with the asymptotic MSE performance of SMC-PF as the number of particles grows large to establish that the MMSE is indeed attained asymptotically; see also [30] and [65].

Consider first the MSE at iteration k, namely

$$\begin{aligned}
\text{MSE}(k) \quad &:= \quad E\left[\left(\hat{I}_\phi(k) - \phi(\mathbf{x}_k)\right)^2\right] = E\left[\left(\hat{I}_\phi(k) - I_\phi(k) + I_\phi(k) - \phi(\mathbf{x}_k)\right)^2\right] \\
&= \quad E\left[\left(\hat{I}_\phi(k) - I_\phi(k)\right)^2\right] + E\left[\left(I_\phi(k) - \phi(\mathbf{x}_k)\right)^2\right] \\
&\quad\quad +2E\left[(\hat{I}_\phi(k) - I_\phi(k))(I_\phi(k) - \phi(\mathbf{x}_k))\right] \\
&= \quad \text{Var}_{PF}(k) \;+\; \text{MMSE}(k) \;+\; \text{Cross Term}. \qquad (5.16)
\end{aligned}$$

The cross term vanishes because

$$\begin{aligned}
\text{Cross Term} \quad &= \quad 2E_{\mathbf{y}_{1:k}}\left[E\left[(\hat{I}_\phi(k) - I_\phi(k))(I_\phi(k) - \phi(\mathbf{x}_k))\Big|\mathbf{y}_{1:k}\right]\right] \\
&= \quad 2E_{\mathbf{y}_{1:k}}\left[E\left[\hat{I}_\phi(k) - I_\phi(k)\big|\mathbf{y}_{1:k}\right] E\left[I_\phi(k) - \phi(\mathbf{x}_k)\Big|\mathbf{y}_{1:k}\right]\right] \\
&= \quad 2E_{\mathbf{y}_{1:k}}\left[E\left[\hat{I}_\phi(k) - I_\phi(k)\big|\mathbf{y}_{1:k}\right] \times 0\right] = 0 \qquad (5.17)
\end{aligned}$$

where the second equality holds because $I_\phi(k)$ is constant when conditioned on $\mathbf{y}_{1:k}$ and the operations required to obtain the PF estimate $\hat{I}_\phi(k)$, namely re-sampling in S1 and drawing new particles in S2, do not depend on the true state $\mathbf{x}_k$ when conditioned on $\mathbf{y}_{1:k}$.

Equations (5.16) and (5.17) imply that $\mathrm{MSE}(k) = \mathrm{Var}_{PF}(k) + \mathrm{MMSE}(k)$. The $\mathrm{MMSE}(k)$ that lower bounds the $\mathrm{MSE}(k)$ depends on the way the latent state variables and measurements are related - a dependency specified by the nonlinear functions $\mathbf{f}$ and $\mathbf{h}$ of the model. Therefore, it is prudent to minimize $\mathrm{Var}_{PF}(k)$ per step k since this amounts to minimizing the $\mathrm{MSE}(k)$.

In the next subsection, $\mathrm{Var}_{PF}(k)$ will be approximated using the IS variance in (5.4) in order to obtain a performance metric, which will be useful to assess the finite-sample efficiency of the SMC-PF tracker.

5.4.1 Finite-sample analysis

Consider specializing (5.4) for the following target and IS densities:

$$q_t(\mathbf{x}_k) := p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \tag{5.18a}$$

$$q_{IS}(\mathbf{x}_k) := \pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = \pi(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{y}_k)p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) \tag{5.18b}$$

and express the per-step variance of PF as

$$\mathrm{Var}_{PF}(k) = E_{\mathbf{y}_{1:k}}\left[E\left[\left(\hat{I}_\phi(k) - I_\phi(k)\right)^2 \Big| \mathbf{y}_{1:k}\right]\right] \tag{5.19}$$

where the inner expectation in (5.19) corresponds to Var_{IS} in (5.4) with the specific choices in (5.18). Substituting (5.4) into (5.19) one arrives at

$$\mathrm{Var}_{PF}(k) \approx (1/M)\, E_{\mathbf{y}_{1:k}}\left[\mathrm{Var}_p\left(\phi(\mathbf{x}_k)\right) E_\pi\left[\left(\frac{p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}\right)^2 \Big| \mathbf{y}_{1:k}\right]\right] . \tag{5.20}$$

Recall that the choice of π is what differentiates B-PF from SMC-PF. Using (5.18), the π-dependent inner expectation in (5.20) can be written as

$$E_\pi\left[\left(\frac{p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}{\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}\right)^2 \Big| \mathbf{y}_{1:k}\right] = \int \frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p^2(\mathbf{x}_k|\mathbf{x}_{k-1})}{p^2(\mathbf{y}_k|\mathbf{y}_{1:k-1})\pi(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{y}_k)} p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) d\mathbf{x}_{0:k}$$

$$\approx \frac{1}{p^2(\mathbf{y}_k|\mathbf{y}_{1:k-1})}\sum_{m=1}^{M}\bar{w}_{k-1}^{(m)}\int\frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)},\mathbf{y}_k)}d\mathbf{x}_k \tag{5.21}$$

where for the approximation $p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$ was replaced by its estimate in (5.6).

For comparison with B-PF, set $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)},\mathbf{y}_k) := p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ to specialize (5.21) to

$$\begin{aligned}\int\frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)},\mathbf{y}_k)}d\mathbf{x}_k &= \left(\int p^2(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})d\mathbf{x}_k\right)\times 1\\ &= \left(\int_{\mathcal{E}_k} p^2(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})d\mathbf{x}_k + \int_{\bar{\mathcal{E}}_k} p^2(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})d\mathbf{x}_k\right)\\ &\quad\times\left(\int_{\mathcal{E}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})d\mathbf{x}_k + \int_{\bar{\mathcal{E}}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})d\mathbf{x}_k\right)\\ &= A^{(m)}(\mathcal{E}_k)B^{(m)}(\mathcal{E}_k) + A^{(m)}(\mathcal{E}_k)B^{(m)}(\bar{\mathcal{E}}_k)\\ &\quad + A^{(m)}(\bar{\mathcal{E}}_k)B^{(m)}(\mathcal{E}_k) + A^{(m)}(\bar{\mathcal{E}}_k)B^{(m)}(\bar{\mathcal{E}}_k)\end{aligned} \tag{5.22}$$

where $\bar{\mathcal{E}}_k$ denotes the complement set of $\mathcal{E}_k$, and $A^{(m)},\ B^{(m)}$ are defined as

$$A^{(m)}(\mathcal{E}_k) := \int_{\mathcal{E}_k} p^2(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})d\mathbf{x}_k,\quad B^{(m)}(\mathcal{E}_k) := \int_{\mathcal{E}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})d\mathbf{x}_k.$$

For the SMC-PF, set $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)},\mathbf{y}_k) := \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, to obtain

$$\int\frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)},\mathbf{y}_k)}d\mathbf{x}_k = c_k^{(m)}\int\frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\alpha_k\mathbb{1}_{\{\mathbf{x}_k\in\mathcal{E}_k\}}+\beta_k\mathbb{1}_{\{\mathbf{x}_k\in\bar{\mathcal{E}}_k\}}}d\mathbf{x}_k = \frac{c_k^{(m)}}{\alpha_k}A^{(m)}(\mathcal{E}_k)+\frac{c_k^{(m)}}{\beta_k}A^{(m)}(\bar{\mathcal{E}}_k) \tag{5.23}$$

where the normalization constant $c_k^{(m)}$ is given by

$$c_k^{(m)} = \int\left(\alpha_k\mathbb{1}_{\{\mathbf{x}_k\in\mathcal{E}_k\}}+\beta_k\mathbb{1}_{\{\mathbf{x}_k\in\bar{\mathcal{E}}_k\}}\right)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})d\mathbf{x}_k = \alpha_k B^{(m)}(\mathcal{E}_k)+\beta_k B^{(m)}(\bar{\mathcal{E}}_k).$$

Substituting $c_k^{(m)}$ back into (5.23) one arrives at

$$\int\frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)},\mathbf{y}_k)}d\mathbf{x}_k = A^{(m)}(\mathcal{E}_k)B^{(m)}(\mathcal{E}_k)+\frac{\beta_k}{\alpha_k}A^{(m)}(\mathcal{E}_k)B^{(m)}(\bar{\mathcal{E}}_k)$$

$$+ \quad \frac{\alpha_k}{\beta_k} A^{(m)}(\bar{\mathcal{E}}_k) B^{(m)}(\mathcal{E}_k) + A^{(m)}(\bar{\mathcal{E}}_k) B^{(m)}(\bar{\mathcal{E}}_k). \quad (5.24)$$

Comparison of (5.22) with (5.24) reveals that SMC-PF is a generalization of B-PF, which allows for more flexibility by adjusting the ratio α_k/β_k to minimize $\mathrm{Var}_{PF}(k)$. For a given set $\mathcal{E}_k$, it is possible to plug (5.24) back into (5.21), differentiate with respect to this ratio, and solve to find the optimal

$$\left(\frac{\alpha_k}{\beta_k}\right)^* = \sqrt{\frac{\sum_{m=1}^{M} \bar{w}_{k-1}^{(m)} A^{(m)}(\mathcal{E}_k) B^{(m)}(\bar{\mathcal{E}}_k)}{\sum_{m=1}^{M} \bar{w}_{k-1}^{(m)} A^{(m)}(\bar{\mathcal{E}}_k) B^{(m)}(\mathcal{E}_k)}}.$$

Unfortunately, this optimal ratio is given in terms of intractable integrals emerging in the definitions of $A^{(m)}(\mathcal{E}_k)$ and $B^{(m)}(\mathcal{E}_k)$. While numerical methods or additional IS estimates can in principle be considered per step to approximate these integrals, this is hardly justified from a complexity perspective. In practice, it is thus reasonable to adjust the ratio α_k/β_k heuristically. Simulations will confirm that the MSE(k) is robust to changes of α_k/β_k, and a heuristic selection yields satisfactory performance.

It is of interest now to investigate the conditions under which the SMC-PF outperforms the B-PF by the largest margin. To this end, consider the following operating assumptions.

A1. Let $\mathcal{E}_k \subseteq \mathbb{R}^{d_x}$ be a set in the state-space of $\mathbf{x}_k$ such that

$$p^2(\mathbf{y}_k|\mathbf{x}_k) \leq \epsilon_1 \ll 1, \quad \forall \mathbf{x}_k \in \bar{\mathcal{E}}_k.$$

A2. For the set $\mathcal{E}_k$ in A1, it holds that

$$\int_{\mathcal{E}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \leq \epsilon_2 \ll 1, \quad \forall\ m = 1, \ldots, M.$$

A3. The likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$ does not have most of its mass in the tail of the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$; thus,

$$\int_{\mathcal{E}_k} p^2(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k := C^{(m)} \gg \epsilon := \max\{\epsilon_1, \epsilon_2\}, \quad \forall\ m = 1, \ldots, M.$$

A4. Given $\epsilon_0 \ll \epsilon$, the number of particles M is selected large enough so that the approximation errors in (5.20) and (5.21) are bounded by ϵ_0.

For notational brevity, the dependence of ϵ_1, ϵ_2 and $C^{(m)}$ on k is suppressed. Condition A1 ensures that the likelihood takes high values for $\mathbf{x}_k$ lying inside $\mathcal{E}_k$; while A2 guarantees that the same set $\mathcal{E}_k$ contains only a small probability mass of the prior. Together, A1 and A2 enforce the so-called *peaky likelihood* condition. A3 asserts that the likelihood does not fall in the tail of the prior density by requiring the relevant integral of their product to be large. Combination of A1-A3 suggests that the likelihood is peaky, but does not fall in the tail of the prior. A4 is needed to ensure that (5.21) offers an accurate approximation of the MSE(k). Under conditions A1-A4, the SMC-PF provides considerable MSE improvement per step k when compared to the B-PF, as summarized in the following proposition.

Proposition 9. *Under A1-A4, for a fixed number of particles M, and with $\beta_k/\alpha_k := \epsilon$, the SMC-PF error variance in (5.20) lowers that of B-PF by a factor $\mathcal{O}(\epsilon)$.*

Proof: Condition A2 implies that $B^{(m)}(\mathcal{E}_k) \leq \epsilon_2$, and hence $1 - \epsilon_2 \leq B^{(m)}(\bar{\mathcal{E}}_k) \leq 1$; while A3 implies that $A^{(m)}(\mathcal{E}_k) = C^{(m)}$. On the other hand, A1 yields

$$A^{(m)}(\bar{\mathcal{E}}_k) = \int_{\bar{\mathcal{E}}_k} p^2(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \leq \int_{\bar{\mathcal{E}}_k} \epsilon_1 p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \leq \epsilon_1 \, .$$

For the B-PF it thus follows that (cf. (5.22))

$$\int \frac{p^2(\mathbf{y}_k|\mathbf{x}_k) p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k \;\; \geq A^{(m)}(\mathcal{E}_k) B^{(m)}(\bar{\mathcal{E}}_k) \geq C^{(m)}(1 - \epsilon_2) \tag{5.25}$$

while for the SMC-PF it holds that (cf. (5.24))

$$\int \frac{p^2(\mathbf{y}_k|\mathbf{x}_k) p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k \leq C^{(m)}\epsilon_2 + \frac{\beta_k}{\alpha_k} C^{(m)} + \frac{\alpha_k}{\beta_k}\epsilon_1\epsilon_2 + \epsilon_1.$$

Upon selecting $\beta_k/\alpha_k = \epsilon$, the π-dependent factor in the PF error variance can be bounded as

$$\int \frac{p^2(\mathbf{y}_k|\mathbf{x}_k) p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k \leq C^{(m)}\epsilon_2 + \epsilon C^{(m)} + \frac{\epsilon_1\epsilon_2}{\epsilon} + \epsilon_1 \leq 2(C^{(m)} + 1)\epsilon. \tag{5.26}$$

Consider two cases. In the first case $C^{(m)} \geq 1$, which implies that the upper bound in (5.26) becomes $4C^{(m)}\epsilon$. Comparing this against (5.25), shows a factor $4\epsilon/(1-\epsilon_2)$ reduction in the evaluated integral for SMC-PF compared to B-PF. In the second case $C^{(m)} \leq 1$, the upper bound in (5.26) becomes 4ϵ. Comparing this against (5.25), shows a factor $4\epsilon/(C^{(m)}(1-\epsilon_2))$ reduction in the evaluated integral for SMC-PF compared to B-PF. With $C^{(m)} \leq 1$, there is at least a reduction of $4\epsilon/(1-\epsilon_2)$ in the second case as well. For both cases, the improvement of SMC-PF over B-PF is $4\epsilon/(1-\epsilon_2) = \mathcal{O}(\epsilon)$. This gain is achieved for every summand and thus for the overall sum in (5.21). Hence, under A4 a factor $\mathcal{O}(\epsilon)$ improvement is established for the per-step variance in (5.20), and the proof is complete. □

5.4.2 Asymptotic analysis

Proposition 9 is important because it provides an approximate performance analysis for finite M. Even though PFs in practice always entail a finite M, it is also critical to ensure that the novel SMC-PF achieves zero error asymptotically as $M \to \infty$. For B-PF, this line of convergence results can be found in [30] and [65]. In order to tailor related theorems for the SMC-PF, it is necessary to define an equivalent system with a properly modified likelihood, and state-transition kernel. To this end, consider factorizing the posterior as

$$\begin{aligned} p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) &\propto p(\mathbf{x}_0)\prod_{\tau=1}^{k} p(\mathbf{y}_\tau|\mathbf{x}_\tau)p(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) \\ &= p(\mathbf{x}_0)\prod_{\tau=1}^{k} \frac{p(\mathbf{y}_\tau|\mathbf{x}_\tau)p(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})}{\pi_{\mathcal{E}_\tau}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})} \times \pi_{\mathcal{E}_\tau}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) \\ &:= p(\mathbf{x}_0)\prod_{\tau=1}^{k} \tilde{p}(\mathbf{y}_\tau|\mathbf{x}_\tau)\tilde{p}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) \end{aligned} \tag{5.27}$$

where $\tilde{p}(\mathbf{y}_\tau|\mathbf{x}_\tau) := p(\mathbf{y}_\tau|\mathbf{x}_\tau)p(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})/\pi_{\mathcal{E}_\tau}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})$ and $\tilde{p}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) := \pi_{\mathcal{E}_\tau}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})$. To guarantee that $\tilde{p}(\mathbf{y}_k|\mathbf{x}_k)$ and $\tilde{p}(\mathbf{x}_k|\mathbf{x}_{k-1})$ satisfy the conditions for convergence in [30] and [65], it will be necessary to invoke the following additional assumptions.

A5. It holds that $\alpha_k, \beta_k > 0$, and $p(\mathbf{y}_k|\mathbf{x}_k)$ is a bounded function of $\mathbf{x}_k$ for the given $\mathbf{y}_k$.

A6. The following lower bound (ν_k) is valid:

$$\int_{\mathcal{E}_k} p(\mathbf{y}_k|\mathbf{x}_k) \left(\int_{\mathcal{E}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} \right) d\mathbf{x}_k > \nu_k > 0.$$

A7. The prior density is finite, that is $p(\mathbf{x}_k|\mathbf{x}_{k-1}) < \infty$.

A8. With $p(\mathbf{y}_k|\mathbf{x}_k)$ satisfying A5, let $\mathcal{L}_k^4$ denote the class of all Borel-measurable functions $\phi : \mathbb{R}^{d_x} \to \mathbb{R}$ upper-bounded by a constant function $\bar{B}(\mathbf{y}_{1:k})$, that is

$$\sup_{\mathbf{x}_k} |\phi(\mathbf{x}_k)|^4 p(\mathbf{y}_k|\mathbf{x}_k) < \bar{B}(\mathbf{y}_{1:k}). \tag{5.28}$$

Corollary 1. *Given A5 and for any bounded Borel-measurable function $\phi : \mathbb{R}^{d_x} \to \mathbb{R}$, the SMC-PF estimator has its per-step error variance bounded as*

$$E\left[\left(\hat{I}_\phi(k) - I_\phi(k) \right)^2 \right] \leq C(k) \frac{\|\phi\|^2}{M} \tag{5.29}$$

where $C(k)$ is a constant, and $\|\phi\| := \sup_{\mathbf{x}} |\phi(\mathbf{x})|$.

Proof: If A5 holds, then the required conditions for [30, Theorem 2] are satisfied for the SMC-PF, which readily establishes the validity of (5.29). □

Corollary 1 asserts convergence in the mean-square sense of the SMC-PF estimate $\hat{I}_\phi(k)$ to the MMSE estimate $I_\phi(k)$. It implies that as $M \to \infty$, the per-step error variance of SMC-PF vanishes, and MMSE(k) is achieved. However, Corollary 1 applies only to bounded ϕ's, which excludes many functions of interest such as $\phi(\mathbf{x}_k) = \mathbf{x}_k(i)$. Those are accommodated under Corollary 2.

To apply the next corollary, two minor modifications of SMC-PF are needed. Let $\left\{ \xi_m^l \right\}_{m=1:M}^{l=1:M}$ denote a set of non-negative weights such that $\sum_{m=1}^M \xi_m^l = \sum_{l=1}^M \xi_m^l = 1$. Instead of drawing $\mathbf{x}_k^{(m)} \sim \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, consider that particles in S2 are generated from a

mixture IS density as follows

$$\mathbf{x}_k^{(l)} \sim \sum_{m=1}^{M} \xi_m^l \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}), \quad \forall l = 1, \ldots, M. \tag{5.30}$$

For each l, drawing particles as in (5.30) amounts to re-sampling one particle from the IS density described by $\{\xi_m^l, \mathbf{x}_{k-1}^{(m)}\}_{m=1}^M$, and subsequently generating one particle from the corresponding $\pi_{\mathcal{E}_k}$. Selecting $\xi_m^m = 1$ and $\xi_m^l = 0$ with $l \neq m$, one arrives at the unmodified SMC-PF. Once all the M samples are generated as in (5.30), the second modification proceeds to check whether the generated particles satisfy

$$\frac{1}{M} \sum_{m=1}^{M} \tilde{p}(\mathbf{y}_k|\mathbf{x}_k^{(m)}) \geq \check{\nu}_k > 0 \tag{5.31}$$

where $\tilde{p}(\cdot)$ is the density defined after (5.27), and $\check{\nu}_k$ denotes an estimate of ν_k in A6.

If (5.31) is satisfied, the weights can be obtained as in S3 of the unmodified SMC-PF; otherwise, the generated particles are discarded, and a new set of particles is drawn according to (5.30). This process is repeated until (5.31) is satisfied. It is known that iterating between (5.30) and (5.31) is not an infinite loop, so long as $\check{\nu}_k$ is chosen close to or smaller than ν_k in A6 [65].

Corollary 2. *Suppose that A5-A8 hold, and the SMC-PF is modified as in (5.30) and (5.31). For any function $\phi \in \mathcal{L}_k^4$ the SMC-PF estimator thus satisfies*

$$E\left[\left(\hat{I}_\phi(k) - I_\phi(k)\right)^4\right] \leq C(k)\frac{\|\phi\|_{k,4}^4}{M^2}$$

where

$$\|\phi\|_{k,4} := \max\left\{1, \left[\int |\phi(\mathbf{x}_s)|^4 p(\mathbf{x}_s|\mathbf{y}_{1:s}) d\mathbf{x}_s\right], \ s = 0, 1, \ldots, k\right\}.$$

It then follows from [65, Corollary 6.1] that

$$\lim_{M\to\infty} \hat{I}_\phi(k) = I_\phi(k), \quad \text{almost surely.}$$

Proof: A5-A8 ensure that SMC-PF satisfies the conditions H0-H2 of [65, Theorem 6.1]. □

Corollary 2 considerably broadens the range of allowable ϕ's by allowing for unbounded functions. It only requires the product of ϕ raised to the fourth power with the likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$ to be bounded away from infinity as a function of $\mathbf{x}_k$ (cf. (5.28)). All such ϕ's belong to $\mathcal{L}_k^4$. When the measurement noise is Gaussian, the likelihood decays exponentially as $\mathbf{x}_k(i)$ grows. Therefore, all polynomial functions of $\mathbf{x}_k(i)$ will belong to $\mathcal{L}_k^4$. For any such function $\phi \in \mathcal{L}_k^4$, Corollary 2 asserts that the fourth-order moment of the per-step error of the SMC-PF vanishes as $M \to \infty$. Furthermore, the SMC-PF estimator converges to the MMSE one with probability 1.

It should be stressed that Corollaries 1 and 2 are applicable only when exact weights are known in SMC-PF. This pertains to the incremental version of SMC-PF or the consensus-based version with the number of consensus iterations approaching infinity. For consensus with finite number of iterations, a residual error in particle weights remains, and more sophisticated analysis would be necessary to derive results similar to Corollaries 1 and 2.

5.5 Efficient sampling from the SMC IS density

This section deals with an efficient means of sampling from the adapted IS density $\pi_{\mathcal{E}_k}(\cdot)$. Recall from Section 5.3.4 that sampling from $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ in S2 relies on RS. The well-known problem with RS is its variable (and possibly large) delay before a sample is accepted [15, Chapter 11]. If $\beta_k \ll 1$ in the present setup and $B^{(m)}(\mathcal{E}_k) \ll 1$, then RS incurs unreasonably large delays because almost all samples fall outside $\mathcal{E}_k$, and are discarded with high probability. Implementing SMC-PF would thus benefit from bounding the maximum delay of RS by allowing only a prescribed maximum of P samples to be rejected. If P samples are rejected and no sample is yet accepted, the recommendation is to resort to IS with a properly chosen surrogate density.

To perform IS, choose as target density $q_t(\mathbf{x}_k) := \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, and select the IS one as the mixture

$$q_{IS}(\mathbf{x}_k) := \gamma \ \mathrm{U}(\mathbf{x}_k|\mathcal{E}_k) + (1-\gamma)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) \tag{5.32}$$

where $\gamma \in [0,1]$ is a design parameter, and $\mathrm{U}(\mathbf{x}_k|\mathcal{E}_k)$ denotes the density which is uniform over $\mathcal{E}_k$, and zero outside $\mathcal{E}_k$. Although γ and q_{IS} are not indexed by m for brevity, these as well as other quantities in this subsection are for a given m, but the results apply to all particles. Since $\mathcal{E}_k$ represents an axes-aligned box, drawing samples uniformly inside $\mathcal{E}_k$ amounts to drawing one independent uniform sample per coordinate. Therefore, sampling from the surrogate $q_{IS}(\mathbf{x}_k)$ in (5.32) is easy.

Next, draw M' samples $\check{\mathbf{x}}_k^{(m')}$ from $q_{IS}(\mathbf{x}_k)$, and weigh them as

$$\check{w}_k^{(m')} \propto \frac{\left(\alpha_k \mathbb{1}_{\{\check{\mathbf{x}}_k^{(m')} \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\check{\mathbf{x}}_k^{(m')} \in \bar{\mathcal{E}}_k\}}\right) p(\check{\mathbf{x}}_k^{(m')}|\mathbf{x}_{k-1}^{(m)})}{\gamma \ \mathrm{U}(\check{\mathbf{x}}_k^{(m')}|\mathcal{E}_k) + (1-\gamma)p(\check{\mathbf{x}}_k^{(m')}|\mathbf{x}_{k-1}^{(m)})} \ .$$

If a particle is now re-sampled from $\{\check{\mathbf{x}}_k^{(m')}, \check{w}_k^{(m')}\}_{m'=1}^{M'}$, it will be approximately coming from $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$.

To optimize this IS process, γ will be selected to minimize the mean-square of IS weights, which is equivalent to minimizing Var_{IS} (cf. (5.4)). This mean-square is given by

$$\begin{aligned}
E[\check{w}_k^2] &= \int \frac{\left(\alpha_k \mathbb{1}_{\{\mathbf{x}_k \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\mathbf{x}_k \in \bar{\mathcal{E}}_k\}}\right)^2 p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\gamma \ \mathrm{U}(\mathbf{x}_k|\mathcal{E}_k) + (1-\gamma)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})} d\mathbf{x}_k \\
&= \int_{\mathcal{E}_k} \frac{\alpha_k^2 p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\gamma \ \mathrm{U}(\mathbf{x}_k|\mathcal{E}_k) + (1-\gamma)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})} d\mathbf{x}_k + \int_{\bar{\mathcal{E}}_k} \frac{\beta_k^2 p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{(1-\gamma)} d\mathbf{x}_k \\
&\approx \frac{\alpha_k^2 \ \mathrm{Vol}(\mathcal{E}_k)}{\gamma} \int_{\mathcal{E}_k} p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k + \frac{\beta_k^2}{1-\gamma} \int_{\bar{\mathcal{E}}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k := \frac{C_1}{\gamma} + \frac{C_2}{1-\gamma}
\end{aligned} \tag{5.33}$$

where in obtaining the last approximation it was assumed that $\mathrm{U}(\mathbf{x}_k|\mathcal{E}_k) \gg p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ for all $\mathbf{x}_k \in \mathcal{E}_k$; and used the fact that $\mathrm{U}(\mathbf{x}_k|\mathcal{E}_k) = 1/\mathrm{Vol}(\mathcal{E}_k)$, where Vol represents the

space volume. The aforementioned assumption is justified because performing IS presumes that the RS stage has failed. Hence, $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ has a very low probability mass inside $\mathcal{E}_k$. Minimizing (5.33) with respect to γ leads to

$$\gamma^* = \frac{\sqrt{C_1}}{\sqrt{C_1} + \sqrt{C_2}} \ .$$

What remains is to obtain, at least approximately, the constants C_1 and C_2. To specify C_2, it suffices to recognize that most of $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ mass should lie outside $\mathcal{E}_k$; hence,

$$C_2 := \beta_k^2 \int_{\bar{\mathcal{E}}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \approx \beta_k^2 \ .$$

For C_1, note that out of P samples drawn from $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ none falls inside $\mathcal{E}_k$; hence, $\mathcal{E}_k$ contains less than $1/P$ of the probability mass. In essence, the approximation $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) \approx \mathrm{U}(\mathbf{x}_k|\mathcal{E}_k)/P$ is invoked inside $\mathcal{E}_k$. Thus, C_1 can be expressed as

$$C_1 := \alpha_k^2 \ \mathrm{Vol}(\mathcal{E}_k) \int_{\mathcal{E}_k} p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \approx \alpha_k^2 \ \mathrm{Vol}(\mathcal{E}_k) \int_{\mathcal{E}_k} \frac{\mathrm{U}^2(\mathbf{x}_k|\mathcal{E}_k)}{P^2} d\mathbf{x}_k = \frac{\alpha_k^2}{P^2} \ .$$

Upon substituting C_1 and C_2 back into γ^*, one arrives at

$$\gamma^* = \frac{\alpha_k}{\alpha_k + P\beta_k} \ .$$

Remark 12. (Computational Cost) Due to the RS step, the exact computational complexity of SMC-PF is a random quantity. However, the IS algorithm proposed in this section provides a deterministic upper bound on complexity. Denoting the complexity of B-PF with M particles as $\mathrm{bpf}(M)$, which is $\mathcal{O}(M)$, the worst- case complexity per time step and sensor of SMC-PF is approximately $\mathrm{bpf}(LM) + M \ \mathrm{bpf}(M') + M \ \mathrm{Sample}_{\mathrm{Comp}}(P) + \mathrm{Weight}_{(5.15)}(M)$, where $\mathrm{Sample}_{\mathrm{Comp}}(P)$ represents the complexity of sampling P particles from the prior and checking if they fall in the set $\mathcal{E}_k$, and $\mathrm{Weight}_{(5.15)}(M)$ stands for the complexity of updating the weights in (5.15). Note that at the beginning of each step sensors run a local B-PF with LM particles; hence, the term $\mathrm{bpf}(LM)$, which is $\mathcal{O}(M)$, appears in

the complexity. After consenting on the global set, sensors sample from the set-membership density. In the worst case, the RS steps fail and one has to resort to the IS method described in this section. For the failed RS steps, the complexity is M $\text{Sample}_{\text{Comp}}(P)$, that is also $\mathcal{O}(M)$; while for the IS step, the complexity is M bpf(M'), which is $\mathcal{O}(M)$ as well. Finally, one has to update the weights of the sampled particles whose complexity is given by $\text{Weight}_{(5.15)}(M)$ that is also $\mathcal{O}(M)$. Since each individual summand in the complexity of SMC-PF is $\mathcal{O}(M)$, the sum is also $\mathcal{O}(M)$. In a nutshell, SMC-PF is computationally more demanding than B-PF, but the complexity of both is linear with respect to the number of particles M.

5.6 Simulations

A network with 100 sensors is considered organized in 20 clusters. Each sensor communicates its distance measurement from the target to its cluster-head sensor. Cluster-head sensors collaboratively perform tracking. For a fixed number of clusters, as the number of sensors increases so does the dimension of data available to cluster-head sensors. In this case, it is prohibitive to communicate raw measurements across the network and distributed approaches should be utilized instead. Fig. 5.6 depicts the network setup along with the associated communication graph. Sensors are depicted as circles and cluster-heads as squares.

The target moves according to a white Gaussian acceleration model [7, p. 273]. The continuous-time state vector $\mathbf{x}(t) := [x_1(t), x_2(t), \dot{x}_1(t), \dot{x}_2(t)]^T$ comprises the target coordinates in two dimensions along with their derivatives. After sampling the continuous model with period T_s (set throughout the simulations to $T_s = 1$), the state equation in discrete

time becomes

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{w}_k, \quad \mathbf{F} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{bmatrix}.$$

The measurement equation expressing the distance from sensor n (located at position $\mathbf{s}_n$) to the target in the presence of zero-mean, unit-variance, additive white Gaussian noise v_k^n, is given by

$$y_k^n = \frac{c_n}{250 + \|\mathbf{s}_n - \mathbf{x}_k(1:2)\|^2} + v_k^n \tag{5.34}$$

where the target position is denoted by the vector $\mathbf{x}_k(1:2)$ using MATLAB notation; and the constant c_n is adjusted so that the signal-to-noise ratio (SNR) in (5.34) is 10dB at a 60 meter distance. These numbers are selected to ensure a sufficiently peaky likelihood function highlighting the gains of SMC-PF relative to B-PF. On the other hand, these numbers and specifically 250 in the denominator of (5.34) prevent a very peaky likelihood, which may lead to divergence of all filters.

The time-averaged root MSE (RMSE) of $\hat{\mathbf{x}}_k(1:2)$ is used to assess performance and is defined as

$$\text{Average RMSE} := \sqrt{\frac{1}{J}\sum_{j=1}^{J} \frac{1}{(K_{\max} - K_{\min})} \sum_{k=K_{\min}+1}^{K_{\max}} \|\mathbf{x}_k^{(j)}(1:2) - \hat{\mathbf{x}}_k^{(j)}(1:2)\|_2^2},$$

where j indexes Monte Carlo runs, $J = 100$ represents the total number of Monte Carlo runs, and $K_{\min} = 8$, $K_{\max} = 16$ specify the time interval over which the MSE is averaged. A non-zero $K_{\min}$ is selected to ensure that choice of initialization does not greatly impact the average RMSE, while $K_{\max} = 16$ is chosen to ensure that the target stays in the sensing area for most realizations. Furthermore, the state noise standard deviation is $\sigma_w = 10$. SMC-PF parameters are $\beta_k = 0.001$, $P = 200$, $L = 10$, and $M' = 100$. Unless stated

otherwise, the particle bounding box method is employed for SMC-PF. Furthermore, the efficient sampling approach of Section 5.5 is utilized for SMC-PF. The simulation run-time for SMC-PF directly depends on the choice of parameters β_k and P. A larger value of β_k increases simulation speed as particles are accepted with higher probability, at the price of a possible performance degradation. On the other hand, a smaller P also increases simulation speed as transition from the RS stage to the IS stage is faster, but can also incur loss in performance. The values selected for these parameters lead to a reasonable trade-off between run-time and performance.

5.6.1 Comparison with B-PF

Fig. 5.7 compares the performance of SMC-PF with that of B-PF and a benchmark centralized PF, which comprises the combination of the auxiliary particle filter (APF) and the unscented particle filter (UPF). In the APF - UPF combination, the two-stage weighting approach of APF is combined with the importance density obtained from UPF. The consensus averaging required for distributed implementation of B-PF and SMC-PF is assumed to be perfect in these figures. Clearly, SMC-PF outperforms B-PF and its performance comes very close to that of the benchmark filter.

The effect of finite number of consensus-averaging iterations is demonstrated in Fig. 5.8. The approach described in Remark 7 is utilized to mitigate residual mismatch between particle weights. With as few as 6 consensus-averaging iterations (note that 6 is also the diameter of the communication graph), the performance of SMC-PF comes close to that with perfect consensus averaging.

Finally, RMSE is plotted versus communication cost in Fig. 5.9. To generate each square in this plot, a single data point corresponding to a given number of particles and given number of consensus iterations is selected from Fig. 5.8. Then, the communication cost for this data point is computed and the result is plotted as a single square in Fig.

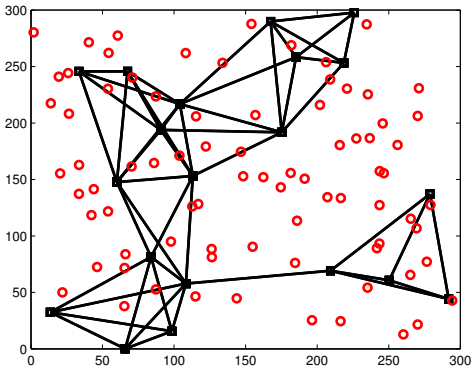

Figure 5.6: Sensor network and its communication graph.

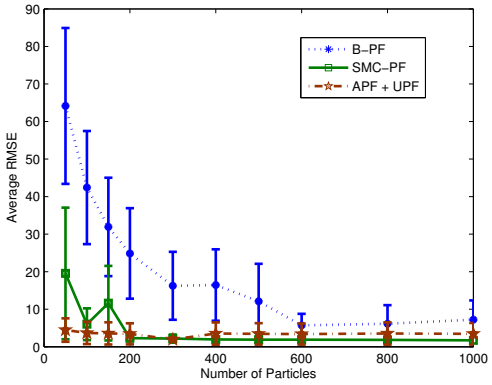

Figure 5.7: Comparison (mean $\pm$ one standard deviation) of B-PF, SMC-PF, and APF combined with UPF as benchmark.

5.9. Repeating this process for every data point in Fig. 5.8 yields all the squares (and similarly triangles) in Fig. 5.9. Thanks to its distributed adaptation, SMC-PF entails lower communication overhead relative to B-PF for a given RMSE in Fig. 5.9. The difference is considerable especially for small RMSE values.

As the number of particles grows large, both SMC-PF (all variants) and B-PF converge to the true MMSE estimator $E[\mathbf{x}_k|\mathbf{y}_{1:k}]$. However, these asymptotic results can be deceiving as the finite-sample performance of the two approaches can be dramatically different; see Fig. 5.7. In addition, the threshold at which the asymptotic results "kick in" can vary considerably depending on the choice of the IS density $\pi(.)$ [32]. Our simulations corroborate that in a practical setup with a finite number of particles, the SMC-PF markedly outperforms the B-PF, while also offering an affordable distributed implementation.

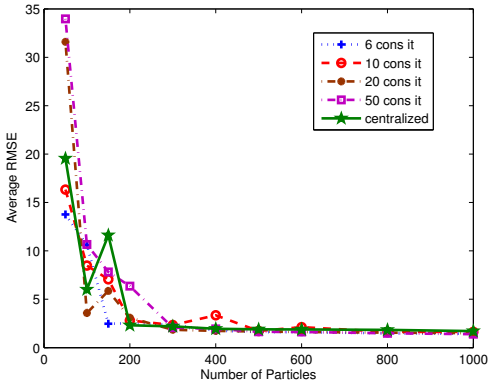

Figure 5.8: SMC-PF with different number of consensus iterations.

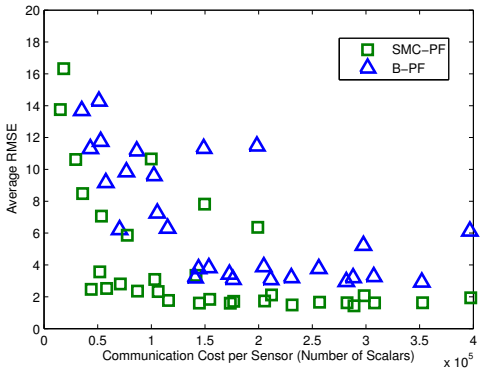

Figure 5.9: RMSE versus communication complexity for B-PF and SMC-PF.

5.6.2 Comparison with distributed GMM-based PFs

SMC-PF is compared here against the consensus-based algorithm in [87], which will be referred to as the distributed PF-1 (DPF-1), and the incremental algorithm in [64], which will be referred to as DPF-2. While comparing an incremental algorithm with a consensus-based one might be unfair as they belong to different classes, the sensor selection approach introduced in [64] to establish the incremental loop allows one to conduct a fair comparison. The proposed method does not require a Hamiltonian cycle. In fact, each sensor in the incremental loop checks its neighbors and determines the ones who have not participated in the incremental loop per round. Among these, the "current" sensor selects the one closest to the target as the next sensor in the loop. When one arrives at a sensor whose neighbors have already participated, the incremental loop ends and the loop for the next round of measurements begins. This approach may lead to sensors left out.

Fig. 5.10 compares SMC-PF with DPF-1 and DPF-2. Here, 200 consensus averaging iterations are performed for DPF-1, and the number of GMM components in DPF-2 is set to 4. It can be seen that SMC-PF outperforms both methods in terms of RMSE. The

communication cost of all methods is plotted in Fig. 5.11 versus RMSE. One observes that DPF-2 incurs the lowest communication overhead, but its performance is worse than what is achievable by DPF-1 and SMC-PF for higher communication cost. SMC-PF also outperforms DPF-1 with smaller communication complexity. Note that the stars corresponding to DPF-2 in Fig. 5.11 are obtained from their respective data points in Fig. 5.10 for different number of particles. Since DPF-2 communicates GMMs rather than particles, its communication complexity is independent of the number of particles and thus all stars fall on the same vertical line in Fig. 5.11. Similarly, the data point for DPF-1 is obtained from Fig. 5.10. Only one data point is plotted as all other ones yielded much higher RMSEs. Finally, we should note that while 6 consensus-averaging iterations seem to be enough for SMC-PF, DPF-1 requires close to 200 or more iterations to yield sufficiently small RMSEs. The main reason for this difference is that consenting on the average of values that are spread, requires more iterations than that of concentrated values. Due to the re-sampling phase, the weights are all given by $1/M$ prior to multiplication by the likelihood. After multiplication by the local likelihood, these weights take values in the interval $[0, 1/M]$; hence, consensus-averaging is run among very concentrated values. On the other hand, the means and covariances that DPF-1 aims to consent on, can have values spread far away from each other; hence, consensus-averaging requires more iterations to converge.

5.6.3 Different SMC-PF implementations

In this section, 20 sensors are considered without cluster heads, collecting measurements

$$y_k^n = \frac{c_n}{1 + \|\mathbf{s}_n - \mathbf{x}_k(1:2)\|^2} + v_k^n \tag{5.35}$$

where c_n is chosen to have an SNR of 10 dB at distance equal to 150 meters. Due to the peaky-likelihood conditions, divergent estimated tracks appear and affect the RMSE. To assess performance of each algorithm, the percentage of accurate tracks is plotted as a

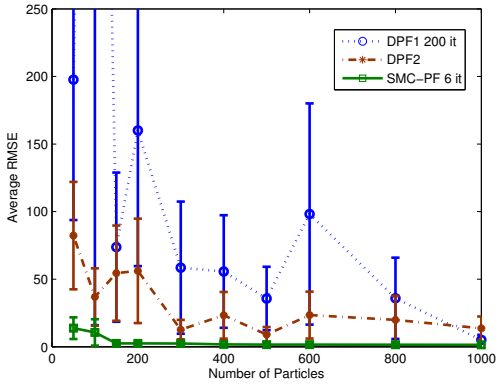

Figure 5.10: RMSE comparison (mean ± one standard deviation) of different DPF algorithms.

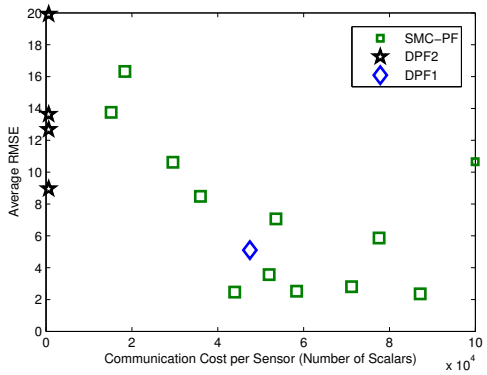

Figure 5.11: RMSE versus communication complexity comparison of different DPF algorithms.

metric. The latter is defined as the fraction of Monte Carlo runs in which the estimated position error stays below a given RMSE threshold (set equal to 10 in the simulated tests)

$$\left\| \hat{\mathbf{x}}_k^{(j)}(1:2) - \mathbf{x}_k^{(j)}(1:2) \right\| \leq \text{RMSE}_{thr} \ , \ \forall \ k: \ K_{\min} \leq k \leq K_{\max} \ .$$

Fig. 5.12 demonstrates the effect of β_k on performance. SMC-PF exhibits robustness to changes in β_k, and only when β_k increases beyond 0.1 its performance begins to deteriorate.

The effect of measurement noise is illustrated in Fig. 5.13, where the SNR is reduced by a factor of 9, and approaches 0 dB at 150 meters distance. In this scenario, an interesting feature emerges. B-PF at SNR= 0 dB outperforms the one at SNR= 10 dB. This is a manifestation of particle depletion, which is more pronounced at 10 dB (more accurate measurements). However, this trend is not observed in SMC-PF whose performance degrades as the SNR decreases.

Different methods for computing the local sets are compared in Fig. 5.14, where the percentages of accurate tracks are plotted. This figure compares the particle bounding box,

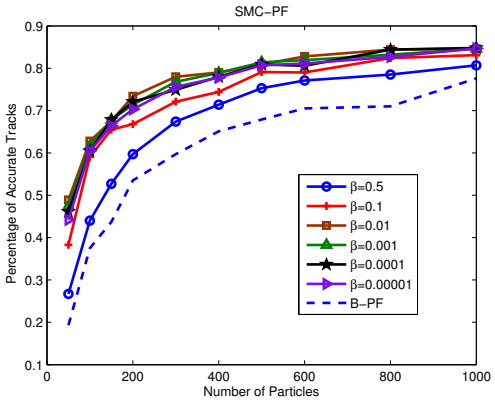

Figure 5.12: SMC-PF percentage of accurate tracks for different values of β_k.

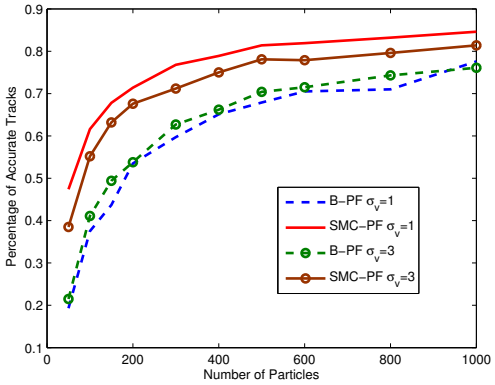

Figure 5.13: SNR effect on performance: Percentage of accurate tracks.

the UKF-based set selection, and the UKF with a 4-component GMM for $\beta_k = 0.001$, and $\sigma_w = 10$. Parameter $\eta^2 = 13.277$ is chosen to ensure that 99% of the probability mass is contained in the ellipsoid defined in LS3′. It is observed that the particle bounding box and UKF with GMM approaches outperform the UKF-only by a small margin, which increases as the number of particles grows. In terms of computational complexity, UKF with GMM is the most demanding while UKF-only and the particle bounding box methods exhibit similar and noticeably lower complexity. Overall, for these simulations, the particle bounding box approach offers the best performance-complexity trade-off.

5.7 Summary

A distributed PF scheme was developed to approximate the practically infeasible optimum MMSE state estimator involved when tracking targets using a network of wireless sensors. The novel non-parametric approach offers a decentralized tracker with performance approaching that of centralized PF, affordable complexity, and inter-sensor communications

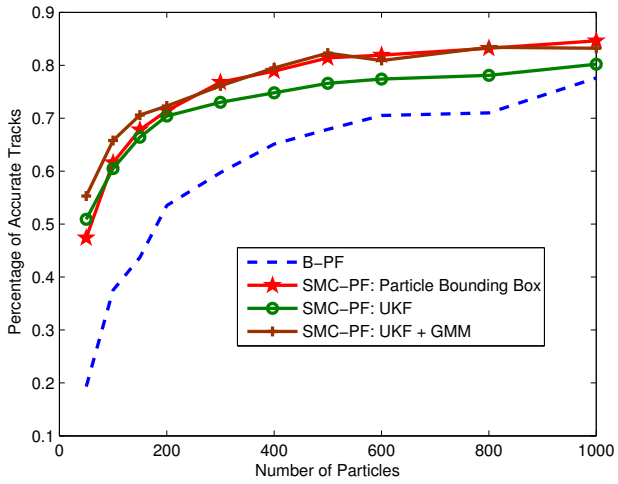

Figure 5.14: Local set selection methods: Percentage of accurate tracks

at reduced overhead.

Instrumental to these desirable features was a data-adapted IS density introduced to draw particles per sensor. The novel IS density relies on a judiciously constructed global set, which becomes available to all sensors using a finite number of min- or max-consensus iterations. Membership to this set allows for distributed PF iterations to be run in a connected network by exchanging only particle weights (but not particles) among neighboring sensors.

Asymptotic analysis established consistency of the distributed PF tracker, while finite-sample analysis based on the reduction of the state-estimator variance per PF iteration revealed conditions under which the novel SMC-PF tracker improves upon a bootstrap PF alternative. Simulated tests confirmed that the novel SMC-PF outperforms the bootstrap PF by a wide margin, which becomes more pronounced as measurements become more accurate, and the peaky likelihood conditions are satisfied.

Chapter 6

Conclusions and future work

Several major challenges in target tracking were investigated and (at least partially) addressed in this thesis. Tackling the nonlinearity present in the measurement model was the subject of Chapter 2. Specifically, sensor measurements were viewed as superposition of the received target signal strengths. Such sensors need not necessarily represent sophisticated radar or sonar systems; simple energy detectors can do the job. Fitting a grid to the space where tracking is performed renders the measurement model linear at the price of having a state with higher dimensions. Fortunately, the state of interest is sparse in this higher dimension. These facts lead to the development of sparsity-aware TSSG-KF and TSSG-IEKF trackers which were shown to improve upon their sparsity-agnostic counterparts. When multiple targets are present, the proposed trackers can bypass data association and track TSSG without the need to know the number of targets. If individual target tracks are desired, data association can be performed on the TSSG-based estimates to recover target tracks. Due to their independent and parallel operation, the proposed TSSG trackers are resilient to data association errors.

Chapter 3 dealt with outliers possibly present in the measurements and/or state that

can degrade KS performance considerably. In the proposed DRS approach outliers were considered as nuisance parameters and jointly estimated along with the desired state. Sparsity in the outlier domain was exploited to satisfactorily perform this joint estimation by penalizing the objective with relevant ℓ_1-norms of the outliers. Methods to tune penalizing coefficients were also presented which rendered the algorithm operational under varying degrees of outlier contamination. Low-complexity iterative methods based on CD and AD-MoM were also developed. A fixed-lag DRS for real-time tracking applications were also established. Numerical tests corroborated the improvement DRS brings about compared to alternative state-of-the-art robust smoothers.

Chapter 4 developed an outlier-resilient RLS algorithm particularly attractive for cases where the measurements arrive in correlated batches. While the general approach was similar to that in Chapter 3, some major modifications were needed. These modifications stemmed from the fact that the fixed-interval DRS in Chapter 3 need not be a real-time algorithm. Indeed, fixed-interval DRS waits until all the data are collected and then estimates the state. This was not the case however, for the fixed-lag smoother. Likewise, the novel robust RLS is an online algorithm and its robust renditions should also enjoy low-complexity and online operation. These issues were addressed in Chapter 4 by deriving a sub-optimal but real-time robust RLS algorithm referred to as OR-RLS. OR-RLS was illustrated to be capable of markedly improving performance in the correlated noise setup which was the case of interest in Chapter 4.

Finally, Chapter 5 developed a consensus-based distributed particle filter referred to as SMC-PF for WSN settings. Consensus-based algorithms are known to be: i) robust to sensors and link failures; and ii) operable with minimum network knowledge i.e., only the knowledge of nearest neighbors for each sensor. The first contribution of SMC-PF was to provide a distributed algorithm which required the communication of particle weights only. Its second contribution was to provide a distributed adaptation mechanism. This second

contribution is unique in the sense that no alternative exists in the currently available literature.

6.1 Future work

One major factor limiting the application of TSSG-KF and TSSG-IEKF algorithms derived in Chapter 2 is grid resolution. Due to the introduction and utilization of covariance matrices in TSSG trackers which contain grid size square scalars, large grid sizes can not be handled as they require too much memory and processing power. Two directions can be pursued to remedy this problem. The first approach is to avoid covariance matrices altogether, or, use them implicitly. This calls for developing algorithms that take into account the covariance matrix implicitly without requiring to really define it for the actual implementation. The second direction concerns the use of multi-resolution grids. In this approach, one starts with a fixed-size grid which is manageable in terms of memory and processing power for the TSSG trackers; and successively shrinks the surveillance area. Beginning with a large surveillance region, one successively zooms into the region(s) where the targets of interest are located. This approach also has the potential to alleviate the grid size limitation. Further research is required in either direction to increase the application range of TSSG trackers.

Numerical integration uses non-overlapping equal length intervals on the range of state space to yield MMSE state estimates. However, this strategy might not be recommended because most of the probability mass of the underlying densities is usually concentrated on a small region of the state space and the mass on other areas is insignificant. Therefore, a lot of computation is wasted to integrate over intervals which will have a very minor effect on the final outcome. PFs tackle this problem by concentrating the particles, and hence integration intervals, on the regions where the probability mass is high. In fact, PF approximates the

underlying distribution with a superposition of impulses. Due to the "zero width" of an impulse, the approximation can become degenerate if only a few particles survive each resampling step, thus leading to particle depletion. To mitigate particle depletion, one can use a more elaborate constituent than an impulse to represent each particle. Still, it should be simple enough to ensure that the complexity of the overall algorithm remains manageable. One possible alternative for impulses are boxes. Each box corresponds to a uniform density. Hence, using boxes instead of impulses amounts to approximating the underlying density with a mixture of uniforms. Subsequently, the results of Chapter 5 can be extended in this direction not only for the distributed setting but for the centralized setup as well.

Bibliography

[1] A. Ahmad and M. R. Gani, "Distributed estimation for nonlinear stochastic signals over sensor networks," *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, Las Vegas, NV, April 2008.

[2] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ, USA: Prentice Hall, 1979.

[3] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the ℓ_1-norm," *IEEE Trans. on Signal Processing*, vol. 58, no. 7, pp. 3436–3447, July 2010.

[4] D. Angelosante, S. Roumeliotis, and G. B. Giannakis, "Lasso Kalman Filtering for Tracking Sparse Signals," *Proc. of 43rd Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1-4, 2009.

[5] A. Y. Aravkin, B. M. Bell, J. V. Burke, and G. Pillonetto, "An ℓ_1-Laplace robust Kalman smoother," *IEEE Trans. on Automatic Control*, June 2010 (submitted); [online] `http://www.math.washington.edu/~aleksand/Publications/l1_rks.pdf`.

[6] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter: Estimation in the presence of measurement origin uncertainty," *IEEE Control Systems Magazine*, pp. 82–100, Dec. 2009.

[7] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*, New York: Wiley, 2001.

[8] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1847–1862, March 2010.

[9] B. M. Bell and G. Pillonetto, *Matlab R/octave Package*, 2007; [online] `http://www.seanet.com/~bradbell/ckbs/ckbs.xml`.

[10] J. Benesty and T. Gaensler, "A robust fast recursive least squares adaptive algorithm," *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001.

[11] D. P. Bertsekas, "Incremental least-squares methods and the extended Kalman filter," *SIAM Journal of Optimization*, vol. 6, no. 3, pp. 807–822, Aug. 1996.

[12] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[13] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods, 2nd ed.* Nashua, NH, USA: Athena Scientific, 1999.

[14] M. Z. A. Bhotto and A. Antoniou, "Robust recursive least-squares adaptive-filtering algorithms for impulsive-noise environments," *IEEE Signal Processing Letters*, vol. 18, no. 3, pp. 185-188, March 2011.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[16] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.

[17] M. Borkar, V. Cevher, and J. H. McClellan, "Estimating target state distributions in a distributed sensor network using a Monte Carlo approach," *Proc. of Wrksp. on Machine Learning for Signal Processing*, Mystic, CT, Sep. 2005.

[18] N. Bouaynaya and D. Schonfeld, "Complete system with head tracking using motion-based particle filter and randomly perturbed active contour," *Proc. of SPIE*, vol. 5685, pp. 864-873, 2005.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, UK: Cambridge University Press, 2004.

[20] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 61–89, 2010.

[21] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. on Information Theory*, vol. 51, no. 12, pp. 4203-4215, Dec. 2005.

[22] O. Cappe, S. J. Godsil, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899-924, May 2007.

[23] V. Cevher, M. F. Duarte, and R. G. Baraniuk, "Distributed target localization via spatial sparsity," *Proc. of the European Signal Processing Conf.*, Laussane, Switzerland, Aug. 2008.

[24] S. C. Chan and Y. X. Zou, "A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: Fast algorithm and convergence performance analysis," *IEEE Trans. on Signal Processing*, vol. 52, no. 4, pp. 975–991, April 2004.

[25] D. E. Clark and J. Bell, "Data association for PHD filter," *Proc. of Intl. Conf. on Intelligent Sensor, Sensor Networks and Information Processing*, Melbourne, Australia, Dec. 2005.

[26] D. E. Clark, K. Panta, and B.-N. Vo, "The GM-PHD filter multiple target tracker," *Proc. of Intl. Conf. on Information Fusion*, Florence, Italy, July 2006.

[27] M. Coates, "Distributed particle filters for sensor networks," *Proc. of Intl. Conf. on Information Processing in Sensor Networks*, Berkeley, CA, April 2004.

[28] J. Cortes, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, pp. 1993-2000, Nov. 2006.

[29] C. Coue, C. Pradalier, C. Laugier, T. Fraichard, P. Bessiere, "Bayesian occupancy filtering for multitarget tracking: An automative application," *Intl. J. of Robotics Research*, vol. 25, no. 1, pp. 19–30, Jan. 2006.

[30] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. on Signal Processing*, vol. 50, no. 3, pp. 736-746, March 2002.

[31] E. Daeipour and Y. Bar-Shalom, "IMM tracking of maneuvering targets in the presence of glint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 996 –1003, Jul. 1998.

[32] F. Daum and J. Huang, "Curse of dimensionality and particle filters," *Proc. of Aerosp. Conf.*, Big Sky, MT, March 2003.

[33] F. Daum and J. Huang, "Nonlinear filters with generalized particle flow," *Proc. of Aerosp. Conf.*, Big Sky, MT, March 2010.

[34] P. M. Djuric, M. Vemula, and M. F. Bugallo, "Target tracking by particle filtering in binary sensor networks," *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2229-2238, June 2008.

[35] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[36] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *J. of Stat. and Computing*, vol. 10, no. 3, pp. 197-208, 2000.

[37] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613-624, Mar. 2001.

[38] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Ed. Oxford University Press, 2010.

[39] O. Erdinc, P. Willett, and Y. Bar-Shalom, "The bin-occupancy filter and its connection to the PHD filters," *IEEE Trans. on Signal Processing*, vol. 57, no. 11, pp. 4232–4246, Nov. 2009.

[40] M. Evans, "Chaining via annealing," *Annals of Statistics*, vol. 19, no. 1, pp. 382-393, 1991.

[41] S. Farahmand, D. Angelosante, and G. B. Giannakis, "Doubly robust Kalman smoothing by exploiting outlier sparsity," *Proc. of Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2010.

[42] S. Farahmand and G. B. Giannakis, "Robust RLS in the presence of correlated noise using outlier sparsity," *IEEE Trans. on Signal Processing*, (submitted: June 2011).

[43] S. Farahmand and G. B. Giannakis, "Robust RLS in the presence of correlated noise using outlier sparsity," *Intl. Wrksp. on Computational Advances in Multi-Sensor Adaptive Processing*, (submitted: July 2011).

[44] S. Farahmand, G. B. Giannakis, and D. Angelosante, "Doubly robust smoothing of dynamical processes via outlier sparsity constraints," *IEEE Trans. on Signal Processing*, (revised: April 2011); [online] `http://arxiv.org/abs/1104.5286`.

[45] S. Farahmand, G. B. Giannakis, G. Leus, and Z. Tian, "Sparsity-aware Kalman tracking of target signal strengths on a grid," *Proc. of Intl. Conf. on Information Fusion*, Chicago, IL, July 2011.

[46] S. Farahmand, G. B. Giannakis, G. Leus, and Z. Tian, "Tracking target signal strengths on a grid using sparsity," *IEEE Trans. on Signal Processing*, (submitted: April 2011); [online] `http://arxiv.org/abs/1104.5288`.

[47] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, "Particle filter adaptation for distributed sensors via set-membership," *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, Dallas, TX, March 2010.

[48] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, "Set-membership constrained particle filter: Distributed adaptation for sensor networks," *IEEE Trans. on Signal Processing*, (re-revised: May 2011).

[49] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, June 1981.

[50] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, "Particle filters for mobile robot localization," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Ed. Springer, 2001, pp. 405-428.

[51] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani, "Pathwise coordinate optimization," *Annals of Applied Statistics*, vol. 1, pp. 302–332, 2007.

[52] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, Feb. 2010.

[53] J.-J. Fuchs, "An inverse problem approach to robust regression," in *Proceedings of the International Conf. on Acoustics, Speech, and Signal Processing*, Phoenix, AZ, 1999, pp. 1809–1812.

[54] W. Gao, H. Zhao, C. Song, and J. Xu, "A new distributed particle filtering for WSN target tracking," *Proc. of Intl. Conf. on Signal Processing Systems*, Singapore, May 2009.

[55] J. Geweke, "Bayesian inference in econometric models using Monte Carlo integration," *Econometrica*, vol. 57, no. 6, pp. 1317-1339, Nov. 1989.

[56] G. B. Giannakis, G. Mateos, S. Farahmand, V. Kekatos, and H. Zhu, "USPACOR: Universal sparsity-controlling outlier rejection," *Proc. of Intl. Conf. on Acoust., Speech, and Sig. Processing*, Prague, Czech Republic, May 2011.

[57] S. Godsill and T. Clapp, "Improvement strategies for Monte Carlo particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Ed. Springer, 2001, pp. 139-158.

[58] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed., Baltimore, MD, USA, 1996.

[59] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140, no. 2, pp. 107-113, April 1993.

[60] D. Gu, "Distributed particle filter for target tracking," *Proc. of Intl. Conf. on Robotics and Automation*, Rome, Italy, April 2007.

[61] D. Gu, J. Sun, Z. Hu, and H. Li, "Consensus based distributed particle filter in sensor networks," *Proc. of Intl. Conf. on Information and Automation*, Zhangjiajie, P. R. China, June 2008.

[62] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd Ed., Cambridge University Press, 2004.

[63] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY: Springer, 2009.

[64] O. Hlinka, P. M. Djuric, and F. Hlawatsch, "Time-space-sequential distributed particle filtering with low-rate communications," *Proc. of Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2009.

[65] X.-L. Hu, T. B. Schon, and L. Ljung, "A basic convergence result for particle filtering," *IEEE Trans. on Signal Processing*, vol. 56, no. 4, pp. 1337-1348, April 2008.

[66] P. J. Huber, "Robust estimation of a location parameter," *Annals of Math. Stats.*, vol. 35, pp. 73–101, Oct. 1964.

[67] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. New York, NJ, USA: Wiley, 2009.

[68] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. on Automatic Control*, vol. 45, no. 5, pp. 910-927, May 2000.

[69] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, Jan. 2000.

[70] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, Inc., 1970.

[71] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators,", *IEEE Trans. on Automatic Control*, vol. 45, no. 3, pp. 477-482, March 2000.

[72] N. Kabaoglu and H. A. Cirpan, "Wideband target tracking by using SVR-based sequential Monte Carlo method," *Signal Processing*, vol. 88, no. 11, pp. 2804-2816, Nov. 2008.

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

[74] V. Kekatos and G. B. Giannakis, "From sparse signals to sparse residuals for robust sensing," *IEEE Trans. on Signal Processing*, vol. 59, 2011 (to appear); [online] `http://spincom.umn.edu/journal.html`.

[75] H. W. Kuhn, "The Hungarian method for the assignment problem," M. Jnger et al Eds., *50 Years of Integer Programming 1958-2008*, pp. 29-47, Springer-Verlag, 2010.

[76] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Comment on "A new method for nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. on Auto. Control*, vol. 47, pp. 1406-1408, Aug. 2002.

[77] S. Lee and M. West, "Markov chain distributed particle filters (MCDPF)," *Proc. of Conf. on Decision and Control*, Shanghai, P. R. China, Dec. 2009.

[78] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part V. Multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255 – 1321, Oct. 2005.

[79] L. Lin, Y. Bar-Shalom, and T. Kirubarajan, "Track labeling and PHD filter for multitarget tracking," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 42, no. 3, pp. 778–795, July 2006.

[80] Q. Ling, G. Wu, C. Jiang, and Z. Tian, "Joint multi-source localization and environment perception in wireless sensor networks," *Proc. of Chinese Control and Decision Conf.*, Xuzhou, China, May 2010.

[81] J. S. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statistics and Computing*, vol. 6, no. 2, pp. 113-119, June 1996.

[82] H. Liu, H. So, F. Chan, and K. Lui, "Distributed particle filtering for target tracking in sensor networks," *Progress in Electromagnetics Research C*, vol. 11, pp. 171-182, 2009.

[83] R. Mahler, "Multi-target Bayes filtering via first-order multi-target moments," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

[84] C. Masreliez and R. Martin, "Robust Bayesian estimation for the linear model and robustifying the Kalman filter," *IEEE Transactions on Automatic Control*, vol. 22, no. 3, pp. 361–371, Jun. 1977.

[85] K. Mekhnacha, Y. Mao, D. Raulo, and C. Laugier, "The "Fast Clustering-Tracking" algorithm in the Bayesian occupancy filter framework," *Proc. of Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Korea, Aug. 2008.

[86] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," *Technical Report CUED/F-INFENG/TR 380,* Cambridge Univ., 2000.

[87] B. N. Oreshkin and M. J. Coates, "Asynchronous distributed particle filter via decentralized evaluation of Gaussian products," *Proc. of Intl. Conf. on Information Fusion*, Edinburgh, Scotland, July 2010.

[88] M. Orton and W. Fitzgerald, "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 216-223, Feb. 2002.

[89] 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.

[90] K. Panta, D. E. Clark, and B.-N. Vo, "Data association and track management for the Gaussian mixture probability hypothesis density filter," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1003–1016, July 2009.

[91] K. Panta, B.-N. Vo, and S. Singh, "Novel data association techniques for the probability hypothesis density filter," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 43, no. 2, pp. 556–570, April 2007.

[92] P. Petrus, "Robust Huber adaptive filter," *IEEE Trans. on Signal Processing*, vol. 47, no. 4, pp. 1129–1133, April 1999.

[93] E. Phelps, P. Willett, T. Kirubarajan, and C. Brideau, "Predicting time to failure using the IMM and excitable tests," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 37, no. 5, pp. 630 –642, Sep. 2007.

[94] M. K. Pitt and N. Shephard, "Auxiliary variable based particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Ed. Springer, 2001, pp. 273-293.

[95] A. Puig, A. Wiesel, and A. Hero, "A multidimensional shrinkage-thresholding operator," in *Proceedings of the 15th Workshop on Statistical Signal Processing*, Cardiff, UK, Sep. 2009.

[96] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *American Institute of Aeronautics and Astronautics Journal*, vol. 3, no. 8, pp. 1445-1450, Aug. 1965.

[97] L. Rey Vega, H. Rey, J. Benesty, and S. Tressens, "A fast robust recursive least-squares algorithm," *IEEE Trans. on Signal Processing*, vol. 57, no. 3, pp. 1209–1216, March 2009.

[98] M. Rosencrantz, G. Gordon, and S. Thrun, "Decentralized sensor fusion with distributed particle filters," *Proc. of Conf. on Uncertainty in Artificial Intelligence*, Acapulco, Mexico, Aug. 2003.

[99] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, New York, NY: Wiley, 1987.

[100] A. Ruszczynski, *Nonlinear Optimization*. Princeton, NJ, USA: Princeton University Press, 2006.

[101] I. C. Schick and S. K. Mitter, "Robust recursive estimation in the presence of heavy-tailed observation noise," *The Annals of Statistics*, vol. 22, no. 2, pp. 1045-1080, 1994.

[102] F. C. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Trans. on Automatic Control*, vol. AC-13, no. 1, pp. 22-28, Feb. 1968.

[103] SeDuMi. [Online.] `http://sedumi.ie.lehigh.edu/`.

[104] Z. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor

network," *Proc. of Intl. Conf. on Information Processing in Sensor Networks*, Los Angeles, CA, April 2005.

[105] C. Song, H. Zhao, and W. Jing, "Asynchronous distributed PF algorithm for WSN target tracking," *Proc. of Intl. Conf. on Wireless Communications and Mobile Computing*, Leipzig, Germany, June 2009.

[106] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Pearson Prentice Hall, 2005.

[107] M. Tanaka and T. Katayama, "Robust fixed-lag smoother for linear systems including outliers in the system and observation noises," *International Journal of Systems Science*, vol. 19, no. 11, pp. 2243–2259, Nov. 1988.

[108] M. P. Tarvainen, P. O. Ranta-aho, and P. A. Karjalainen, "Time-varying ARMA modeling of nonstationary EEG using Kalman smoother algorithm," *Proc. of the Finnish Signal Processing Symposium*, Espoo, Finland, June 2001.

[109] G. Taylor and L. Kleeman, *Visual Perception and Robotic Manipulation*, Springer-Verlag, 2006.

[110] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of Royal Statistical Society Series B*, vol. 58, no. 1, pp. 267-288, 1996.

[111] P. Torma and C. Szepesvari, "Local importance sampling: A novel technique to enhance particle filtering," *Journal of Multimedia*, vol. 1, no. 1, pp. 32-43, April 2006.

[112] J. Tropp, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Trans. on Info. Theory*, pp. 1020-1051, March 2006.

[113] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475-494, June 2001.

[114] N. Vaswani, "Kalman filtered compressed sensing," *Proc. of the 15th Intl. Conf. on Image Processing*, pp. 893–896, San Diego, CA, Oct. 2008.

[115] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.

[116] T. Wu and K. Lange, "Coordinate descent algorithms for Lasso penalized regression," *Annals of Applied Statistics*, vol. 2, pp. 224–244, 2008.

[117] J. Wu and H. Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *Telecommun. Syst. J.*, vol. 3, pp. 6384, Sep. 2001.

[118] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Sys. and Control Ltrs.*, pp. 65-78, Sept. 2004.

[119] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Trans. on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.

[120] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society, Series B*, vol. 68, no. 1, pp. 49–67, 2006.

[121] Y. Zou, S. C. Chan, and T. S. Ng, "A recursive least M-estimate (RLM) adaptive filter for robust filtering in impulsive noise," *IEEE Signal Processing Letters*, vol. 7, no. 11, pp. 324-326, Nov. 2000.