

State estimation in networked systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op donderdag 26 april 2012 om 16.00 uur

door

Joris Sijs

geboren te Eindhoven

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. P.P.J. van den Bosch

Copromotor:

dr. M. Lazar

Coverdesign by J. Sijs

A catalogue record is available from the Eindhoven University of
Technology Library, ISBN: 978-90-386-3114-1

State estimation in networked systems

Eerste promotor: prof.dr.ir. P. P. J. van den Bosch

Copromotor: dr. M. Lazar

Kerncommissie:

prof.dr. K. H. Johansson

prof.dr.-ing U. D. Hanebeck

prof.dr.ir. W. M. P. H. Heemels

The Ph.D. work is supported by the research program Adaptive Multi Sensor Networks of TNO.



The Ph.D. work forms a part of the research program of the Dutch Institute of Systems and Control (DISC).

Contents

Summary	4
1 Introduction	9
1.1 Web of information	10
1.2 Stochastic state estimation	11
1.3 Challenges in networked estimation	15
1.3.1 Communication in centralized estimation	15
1.3.2 Cooperation in distributed estimation	17
1.4 Outline of the thesis	20
1.5 Notation and definitions	23
2 Event based state estimation	27
2.1 System description	28
2.2 A mathematical formulation of sampling	29
2.3 An event based state-estimator	31
2.3.1 Step 1: likelihood formulation	32
2.3.2 Step 2: state estimation	35
2.3.3 Step 3: state approximation	36
2.4 Asymptotic analysis	37
2.4.1 Preliminaries	37
2.4.2 Asymptotic analysis of the error-covariance	38
2.5 Illustrative example	40
2.6 A discussion on package loss	42
2.7 Conclusions	45
3 Event based state estimation in a control loop	47
3.1 A feedback control set-up	47
3.2 Matched sampling	50
3.2.1 Event triggering criteria	50
3.2.2 EBSE with Matched sampling	52
3.2.3 An illustrative comparison	53
3.3 Event based estimation with control actions	55
3.4 Integration with a robust MPC	57
3.4.1 A robust MPC algorithm	57
3.4.2 Integration of the EBSE and robust MPC	62
3.5 Illustrative control example	64
3.6 Conclusions	66

4	State fusion with unknown correlations: Ellipsoidal intersection	69
4.1	Fusion in sensor networks	70
4.2	The state fusion objective	71
4.3	Fusion strategy and related work	72
4.4	Ellipsoidal intersection	73
4.4.1	Characterization of the unknown correlation	73
4.4.2	An explicit formula of the mutual covariance	76
4.4.3	An explicit formula for the mutual mean	78
4.5	Illustrative example	80
4.6	Conclusions	81
5	Cooperative state estimation	83
5.1	System description	83
5.2	Cooperative Kalman filters	86
5.2.1	Justification of the design	87
5.2.2	Implementation of CKFs	88
5.2.3	Asymptotic analysis	90
5.3	Illustrative example	92
5.4	Nonlinear state estimation	94
5.4.1	Cooperative state estimation	95
5.4.2	Benchmark application of tracking shockwaves	97
5.5	Conclusions	100
6	An overview of distributed Kalman filtering	103
6.1	System description	103
6.2	Synchronization versus fusion	105
6.2.1	Synchronization	106
6.2.2	Fusion	107
6.3	Exchanging measurements	109
6.3.1	Approach 1: fusion of measurements	110
6.3.2	Approach 2: synchronization of measurements	112
6.4	Exchanging local estimates	113
6.4.1	Approach 3: synchronization of state estimates	114
6.4.2	Approach 4: fusion of state estimates	116
6.5	Illustrative examples	118
6.5.1	Object tracking with camera	119
6.5.2	Environmental diffusion process	123
6.6	Extended algorithms	128
6.7	Conclusions	129
7	Conclusions	131
7.1	Contributions	131
7.1.1	Exploiting event sampled measurements	131
7.1.2	Fusion of estimates with unknown correlation	133
7.1.3	Cooperation in distributed estimation	133

7.2	Ideas for future research	134
7.2.1	Event based estimation and control	135
7.2.2	Fusion of estimates	135
7.2.3	Cooperative estimation of reduced states	135
	Bibliography	136
	Appendix	146
A	Matrix properties	147
B	Proofs corresponding to Chapter 2	149
C	Proofs corresponding to Chapter 3	155
D	Proofs corresponding to Chapter 4	159
E	Proofs corresponding to Chapter 5	165
F	Proofs corresponding to Chapter 6	169
	Acknowledgement	171
	Curriculum Vitae	173

Summary

State estimation in networked systems

The research presented in this thesis concentrates on state estimation in networked systems. State estimation refers to a method for computing the unknown state vector of a dynamic process by combining sensor measurements with predictions from a process model. The most well known method for state estimation is the Kalman filter, which assumes a linear process model and Gaussian noise distributions. The Kalman filter started as an essential part of various space and military applications. Since then, many of its successful implementations are found in the public domain as well. Motivated by this success, additional estimation methods were designed, such as, the extended Kalman filter, unscented Kalman filter and particle filter. These methods deal with nonlinear process models and/or non-Gaussian noise distributions.

Up until now, there were no critical limitations in communication and computational resources. The amount of sensor measurements and the model complexities have been sufficiently low to satisfy the requirements for a *centralized* implementation of state estimation algorithms. This is changing with a paradigm shift in system design towards networked systems, and “sensor networks” in particular. Networked systems can maintain large amount of sensors. However, they often lack communication and/or computational resources that are required for processing the large quantity of produced measurements according to a classical centralized implementation. To solve this issue, novel state estimation strategies are presented for networked systems. In the first estimation approach, the amount of measurement samples is reduced with *event sampling* to cope with communication channels that have a limited capacity for exchanging data. In the second estimation approach, measurements are processed directly at the sensor in a *distributed state-estimator* to deal with communication and computational limitations of large-scale or ad-hoc networks. A brief motivation for studying these two approaches is given, next.

Limitations in the amount of exchanged data from sensor to estimator arise when a (wireless) network connection is used for transferring the data. To reduce the amount of exchanged data, measurements are sampled at the instants of an event on the sensor value, rather than synchronously in time. However, this complicates the estimation problem considerably, as events occur unexpectedly. Therefore, the first estimation approach proposed in this thesis focuses on stable estimation results for any type of event sampling strategy. This means that a bounded error-covariance is attained by performing an *update* on the estimation results not only at the instants of an event, when a new measurement is available, but also synchronously in time when no measurement is received. In the latter case, the update is based on

the implicit property of event sampling that not receiving a new measurement still gives information on the current sensor value. After a theoretical study, the proposed *event based state-estimator* is further integrated with a feedback control system to create a new type of event based controller. The distinguishing property of this set-up, i.e., an event based estimator prior to a time synchronous controller, is that stability of the controlled system is decoupled from the event sampling strategy. To that extent, the results of the event based state-estimator are interpreted by an integration procedure, so that the employed controller can optimize disturbance rejection depending on estimation errors.

Communication and computational limitations in sensor networks are rather different. These types of networked systems consist of a large amount of so called “sensor nodes” that are spatially distributed to monitor large-area processes. Typical design consideration for preventing that a networked system is deployed with large amounts of wires, imply that sensor nodes are battery powered and exchange data via a radio. The communication range of nodes is often limited to save energy, while the raised computational limitations are caused by the fact that a single node is not able to process all measurements produced by the sensor network. Therefore, distributed solutions for state estimation are being developed, in which each node typically computes a local estimate of the state vector based on its own measurement and on the data received from neighboring nodes. Some main drawbacks of current solutions is their focus on minimizing the estimation error per node individually and further, impose strict requirements on the shared data which are likely to be violated by system changes present in sensor networks. To solve these issues, the second estimation strategy proposed in this thesis does not focus on the estimator per node individually but aims to establish a cooperation in this *network of estimators*. Cooperation means that neighboring nodes share data not only to synchronize their estimation results but, more importantly, to reduce the (modeled) estimation error. In the proposed set-up each node employs a state estimation method locally, e.g., the (unscented) Kalman filter, to estimate the state based on its own measurement. This estimation result is then shared with neighboring nodes as input to a state fusion method, which computes a fused estimate of the state. A novel fusion method is developed for merging two of these estimates, such that the modeled estimation error (error-covariance) after fusion is reduced. A fulfillment of this property guarantees that the estimation error of each node in the network is in line with the smallest estimation error found across its nodes. Furthermore, the proposed distributed estimation approach is assessed in a comprehensive overview on distributed Kalman filtering. To that extent, the widely scattered solutions on this topic that were proposed in various research communities are assessed in two real-life inspired case studies.

The thesis concludes with the main contributions of the presented research, followed by ideas for future investigation.

Motto:

“Only those who will risk going too far can possibly find out how far one can go.”

- *T.S. Elliot*

1

Introduction

1.1	Web of information	1.3	Challenges in networked estimation
1.2	Stochastic state estimation	1.4	Outline of the thesis
		1.5	Notation and definitions

A recent study done by Cisco revealed that in 2008 the number of *things* connected to the Internet exceeded the *population on earth*. Moreover, this number is likely to increase, due to the growing amount of devices with Internet capabilities, such as televisions, smartphones and navigation systems. An illustration of this study is depicted in Figure 1.1. Along that line, it is worth noticing that Cisco and NASA initiated the research platform “Planetary Skin” in 2009, which will use billions of networked sensors distributed on land, in sea and in space to provide information on environmental changes. Such extreme quantities are not only future trends. For example, in 2011 alone the Internet generated 4 million terabyte of unique information. It is expected that this number will continue to increase, exponentially. Mankind will never be able to understand all this information without help. Therefore, novel algorithms are being developed to turn this excessive amount of information into knowledge and wisdom.

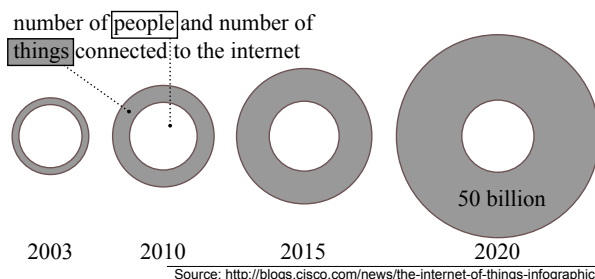


Figure 1.1: Over the years, things other than people have been connected to the Internet more frequently. Things, such as televisions, health monitors for cows, telephones and radio stations.

1.1 Web of information

The growing amount of information is generated by all kinds of information sources, such as sensor measurements, camera recordings, e-mails and text messages. They provide insight on surrounding processes that help us manage our daily activities. For example, information about stock exchange, weather conditions, traffic jams, pollution levels and energy consumptions. A key element in this web of information is a communication network, as it enables to share and manage large quantities of information sources. This thesis focusses on sensor measurements as the information source. Sensors can measure all kinds of (physical) variables, such as temperature, force, angle, light, acceleration, speed, position and magnetism. They are used in almost every electronic device that we know, from amplifiers to smartphones.

When sensors are connected to a communication network they create a so called *sensor network*. A sensor network consists of *sensor nodes* in a particular network topology, as depicted in Figure 1.2. Each sensor node combines multiple sensors, a central processing unit (CPU) and a data connection (wired or wireless) on a circuit board. Some examples of sensor nodes are the Tmote-Sky, G-node and Waspnode, though one might consider a smartphone as a sensor node as well. Sensor networks have three attractive aspects for system design: they require low maintenance, create “on-the-fly” (ad-hoc) communication networks and can maintain large amounts of sensors.

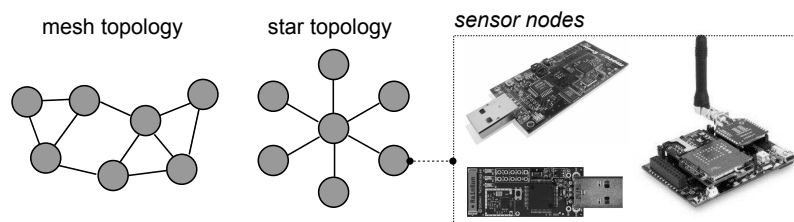


Figure 1.2: Some examples of commercially available sensor nodes, i.e., Tmote-Sky (top-left), G-node (bottom-left) and Waspnode (right), with two possible network topologies to attain a certain communication strategy.

Nowadays, sensor nodes are commercially off-the-shelf products and give system engineers new tools for acquiring measurements. Although they make sensor measurements available in large quantities, solutions for processing these measurements automatically are hampered by limitations in the available resources, such as energy, communication and computation.

Energy plays an important role in remotely located processes. Such processes are typically observed by battery-powered sensor nodes that are not easily accessible and thus should have a long lifetime. Some applications even deploy sensor nodes in the asphalt of a road to monitor traffic, or in the

forrest to collect information on habitats. See, for example, the applications described in (Papp et al., 2009; Szewczyk et al., 2004) and recent surveys on sensor networks in (Kahn et al., 1999; Akyildiz et al., 2002; Chong and Kumar, 2003; Lewis, 2005). To limit energy consumption, one often aims to minimize the usage of communication and computational resources in sensor nodes. However, there are other reasons why these latter two resources should be used wisely.

Limited communication mainly results from upper bounds on the network capacity, as it was established in the Shannon-Hartley theorem for communication channels presented in (Shannon and Weaver, 1949). It shows that the environment in which nodes communicate influences the amount of data that can be exchanged without errors. In addition, communication is affected by package loss as well, which occurs when two different data packages jointly arrive at the same sensor node. Hence, a suitable strategy for exchanging data is of importance to cope with the dynamic availability of communication resources.

Computational demand is related to the algorithms performed in sensor networks for processing the measurements. The established centralized solutions, where measurements are processed by a single node, fail for large-scale networks even when communication is not an issue: with an increasing amount of sensor nodes the computational load of a centralized solution will grow exponentially, up to a point that it is no longer feasible or highly inefficient. To that extent, non-centralized solutions are explored that aim to make clever use of local CPUs that are already present in each node.

Limitations of the above resources are important design parameters when setting up a sensor network, and networked systems in general. One aspect of this design is related to the application goal of the networked system, which further determines whether measurements are processed by, for example, a state-estimation, classification or a control method. However, current solutions that correspond to those methods do not account for the above limitations in communication and computation. Yet, networked systems have become an important part of our society. Therefore, new areas of research were triggered, in which novel solutions for these established methods are being developed. This thesis focusses on novel solutions for stochastic state estimation in networked systems. As such, let us start by presenting the main ideas of state estimation, next.

1.2 Stochastic state estimation

State estimation refers to a method for computing the unknown state vector of a dynamic process by combining measurement samples with predictions

from a process model. The state gives an instantaneous summary of the process' dynamical behavior. For example, the position and speed of a vehicle or the temperature distribution within a room, as it is depicted in Figure 1.3. To that extent, the process dynamics are represented in a so called *state-space* model, which is a dedicated mathematical model that describes the behavior of a process in time. The variables in this model are a state $x \in \mathbb{R}^n$, control action $u \in \mathbb{R}^l$ and measured quantity $y \in \mathbb{R}^m$. As an example, let us present the discrete-time state-space model that corresponds to a linear process, given that measurements are sampled synchronously in time, i.e.,

$$\begin{aligned} x[k] &= Ax[k-1] + Bu[k-1] + w[k-1], \\ y[k] &= Cx[k] + Du[k] + v[k]. \end{aligned} \quad (1.1)$$

The time that corresponds to the k -th sample instant is denoted as $t_k = k\tau_s$, for some sampling time $\tau_s \in \mathbb{R}_+$. Further, the process dynamics are modeled by $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{l \times n}$, whereas $C \in \mathbb{R}^{m \times n}$ and $D \in \mathbb{R}^{m \times l}$ characterize the measurement. The remaining variables are process noise $w \in \mathbb{R}^n$ and measurement noise $v \in \mathbb{R}^m$. They model any undetermined behavior of the process dynamics and the sensor value, respectively.

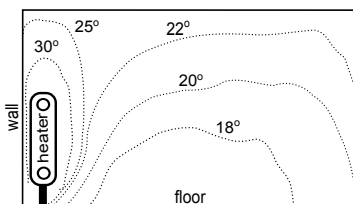


Figure 1.3: An example of the state vector representing an instantaneous summary of the temperature distribution in a cross-section of a room.

The objective of state estimation is to reconstruct $x[k]$ from the measurement samples $y[0], \dots, y[k-1], y[k]$ and a process model, e.g., the model in (1.1). Typical for *stochastic* state estimation is that the variables w and v (and as a result x) are considered as random vectors described with some probability density function (PDF). A breakthrough in stochastic state estimation is the Kalman filter, formally presented in (Kalman, 1960), in which w and v are characterized by Gaussian PDFs, i.e.,

$$p(w[k]) = G(w[k], 0, Q) \quad \text{and} \quad p(v[k]) = G(v[k], 0, V). \quad (1.2)$$

The covariances $Q \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{m \times m}$ are nowadays treated as design parameters, though they were originally introduced to denote the covariance of the process-noise and the measurement-noise, respectively. Further note that a Gaussian distributed w and v in combination with the linear process model of (1.1) implies that the state x is Gaussian distributed as well, i.e., it is

characterized with a certain mean $\mathbb{E}[x]$ and covariance $\text{cov}(x)$. The Kalman filter computes an estimated value of this mean and covariance based on the available measurements. These estimated values thus characterize the following PDF, which is to be determined by the Kalman filter, or by any stochastic state-estimator in general, i.e.,

$$p(x[k]|y[0], \dots, y[k-1], y[k]) := G(x[k], \hat{x}[k], P[k]). \quad (1.3)$$

The estimated mean is denoted as $\hat{x} \in \mathbb{R}^n$, while $P \in \mathbb{R}^{n \times n}$ is a model for the estimation error $\text{cov}(x - \hat{x})$. A common term for P is the *error-covariance*.

The Kalman filter computes an updated value for $\hat{x}[k]$ and $P[k]$ at each sample k in two steps, see also Figure 1.4. A prediction of $x[k]$ is determined first by applying the process model (1.1) on the estimation results of the previous sample instant, i.e., on $\hat{x}[k-1]$ and $P[k-1]$. The notation k^- is used to emphasize the *predictive* character of $\hat{x}[k^-]$ and $P[k^-]$ as an estimated value for $\mathbb{E}[x[k]]$ and $\text{cov}(x[k] - \hat{x}[k^-])$, respectively. This prediction is then corrected with the current measurement $y[k]$ in the second step to compute the updated estimation results $\hat{x}[k]$ and $P[k]$ of (1.3). Extension of the Kalman filter to nonlinear processes and/or non-Gaussian distributions, e.g. the unscented Kalman filter of (Julier and Uhlmann, 1997a) and the particle filter presented in (Ristic et al., 2004), follow the same prediction-update scheme. Thorough evaluations of the Kalman filter are found in (Anderson and Moore, 1979; Grewal and Andrews, 1993; Welch and Bishop, 1995).

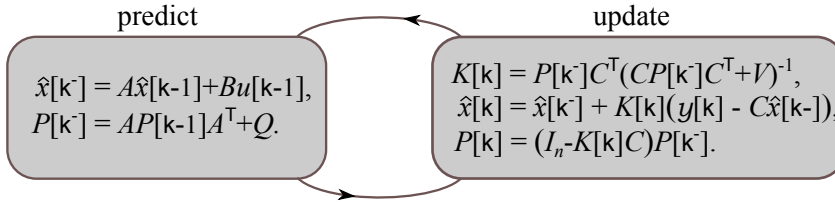


Figure 1.4: An illustration of the Kalman filtering algorithm.

The success of the Kalman filter is based on three aspects.

- Measurements are included iteratively.
- The estimation error is unbiased, i.e., $\lim_{k \rightarrow \infty} (x[k] - \hat{x}[k]) = 0$, and attains the minimal $\text{tr}(P[k])$ for linear processes and Gaussian PDFs;
- The algorithm involves a limited set of (relatively simple) equations.

Among others, the Kalman filter is popular in navigation applications, for example, in recent navigation systems used in vehicles.

Example of Kalman filtering in navigation systems

On-board navigation systems measure the vehicle's position in noisy latitude and longitude coordinates using a GPS device. A Kalman filter is used to reconstruct the vehicle's position and speed in the two dimensions. As such, the process model is characterized by two times the double integrator, in which $a \in \mathbb{R}^2$ denotes the unknown acceleration of the vehicle, i.e.,

$$x[k] = \begin{pmatrix} 1 & \tau_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau_s \\ 0 & 0 & 0 & 1 \end{pmatrix} x[k-1] + \begin{pmatrix} \frac{1}{2}\tau_s^2 & 0 \\ \tau_s & 0 \\ 0 & \frac{1}{2}\tau_s \\ 0 & \tau_s \end{pmatrix} a[k-1],$$

$$y[k] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} x[k] + v[k].$$

Realistic values are a sampling time $\tau_s = 1$ [s], a maximum acceleration $a < 2$ [m/s²] and a measurement noise v ranging 10 [m]. Further, the effects of an unknown acceleration a on the state x are modeled as process noise w . To complete the process model of (1.1), let $u = 0$. Based on these settings, while assuming that vehicles drive on roads, the navigation system gives an estimated position and speed of the vehicle. An illustrative impression of such an estimated trajectory based on GPS measurements is depicted in the street map of Figure 1.5. Notice that navigation without a Kalman filter is difficult, as the inaccurate GPS measurements themselves result in an incorrect vehicle position.

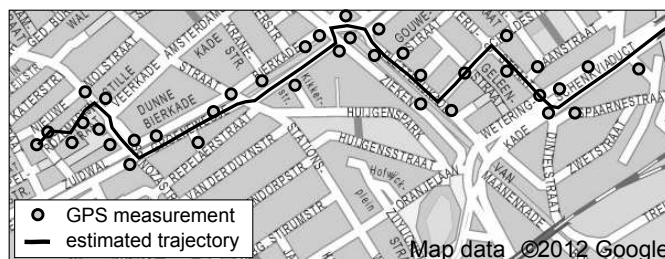


Figure 1.5: An illustrative impression of the estimated trajectory that is computed by a navigation system based on GPS measurements.

The Kalman filter, alike many other state estimators, is a centralized method where new measurements are acquired synchronously in time. Fulfilling these two requirements is unrealistic for networked systems, due to limited communication and computational resources. Furthermore, the spatial distribution of sensor nodes when deploying a sensor network introduces additional challenges for state estimation, which are presented next.

1.3 Challenges in networked estimation

Research in the field of sensor networks emphasized on creating ad-hoc and mobile networks. An important outcome was the standardized connection model for sensor nodes illustrated in Figure 1.6, which is similar to the OSI-stack for connecting computers via the internet.

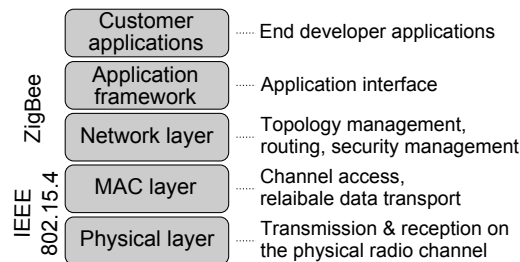


Figure 1.6: A model for connecting nodes that employ ZigBee technology.

The connection model of Figure 1.6 indicates that state estimation is an example of “customer applications”. This means that estimation data will pass four different layers when it is exchanged from one node to another. Especially the final “physical layer”, for which the required energy increases exponentially with the physical distance between nodes, causes that exchanging one bit consumes much more energy than processing one bit. Therefore, it is often desired to reduce the amount of exchanged data at the costs of more complex computations. Another consequence of the communication network is that computational and sensing resources of the individual nodes are physically dislocated. Since both these resources are required for any estimation method, the network topology has a vital impact on the applicability of any novel state estimation solution. This aspect has led to two different types of estimation approaches that each correspond to one of the fundamental network topologies. A *centralized* estimation strategy for star topologies and a *distributed* estimation strategy for mesh topologies, see also Figure 1.7. Each strategy aims to solve the characteristic issues of the corresponding set-up. Note that any other network topology can be constructed as a combination of star and mesh topologies.

1.3.1 Communication in centralized estimation

In centralized estimation strategies a single node acquires all measurement samples produced in the sensor network for estimating the state vector. The classical approach is to perform an estimation algorithm that receives new measurement samples synchronously in time, such as the Kalman filter. However, when measurements are exchanged via a communication network, the amount of data that can be exchanged per second is limited. One way to

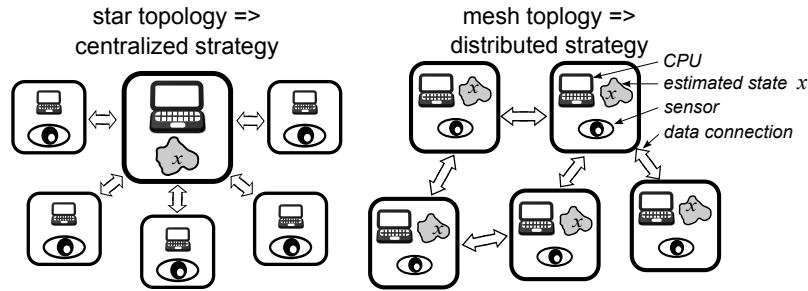


Figure 1.7: The two fundamental design strategies that are found in the literature on networked estimation. In a centralized strategy one node combines all the measurements, whereas in a distributed approach local measurements are already processed in the corresponding sensor node to estimate the state.

deal with this limited communication, is to reduce the size of data packages by defining a quantization-level on the synchronously sampled measurements and thereby, represent their value in fewer digits. See, for example, a wide variety of quantized state-estimators in (Ribeiro et al., 2010). A drawback of this approach is that data is still exchanged synchronously in time, imposing package loss and the exchange of irrelevant measurement samples. Package loss can be moderated via asynchronous sampling strategies, as it will reduce the probability that packages are sent at equal time instants. Furthermore, a collection of asynchronous strategies that evaluates the sensor value before taking a measurement sample is known as *event sampling*.

The advantage of evaluating the sensor value for state estimation, is that it gives additional knowledge on the sensor value at those instants that no new measurement sample is received by the estimator. This inherent knowledge can thus be used to improve estimation results, due to which fewer measurements are required for achieving a similar performance. Event sampling has the potential to provide such knowledge, as samples are not generated synchronously in time but only when an a priori defined event occurs in the sensor value. Some examples of event sampling, as depicted in Figure 1.8, are *Send-on-Delta* (or Lebesgue sampling) and *Integral sampling*, as proposed in (Heemels et al., 1999; Åström and Bernhardsson, 2002; Miskowicz, 2006, 2007). Note that Send-on-Delta could be regarded as a particular quantization strategy. Without going into a detailed explanation, it is worth to point out that event sampling is already used in “networked control systems” to design event based controllers, e.g., (Heemels et al., 1999; Xu and Hespanha, 2004; Imer and Basar, 2006; Henningsson et al., 2008; Dimarogonas and Johansson, 2009; Cogill, 2009; Lehmann and Lunze, 2010; Donkers, 2012). In relation to event based control, (Tabuada, 2007; Heemels et al., 2008; Wang and Lemmon, 2009) define an event sampling strategy such that *stability* of the controlled system is guaranteed.

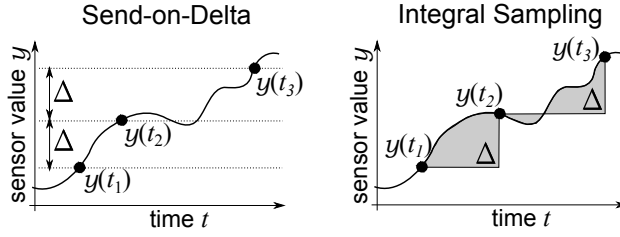


Figure 1.8: Two examples of event sampling: *Send-on-Delta*, in which the k -th event instant is triggered at $t_k = t$ such that $|y(t) - y(t_{k-1})| \geq \Delta$, and; *Integral sampling*, in which the k -th event instant is triggered at $t_k = t$ such that $\int_{t_{k-1}}^t |y(\tau) - y(t_{k-1})| d\tau \geq \Delta$, for any $k \in \mathbb{Z}_+$ and some $\Delta \in \mathbb{R}_+$.

The term *stability* in the area of state estimation refers to a non-diverging error-covariance, as it further implies a non-diverging estimation error. Some first solutions for attaining stable estimation results with event sampling are concentrated on *when* to send new measurements, for example, to minimize estimation error as proposed in (Imer and Basar, 2005; M. Rabi and Basar, 2006; Cogill et al., 2007). Therein, an updated estimate of the state is computed when new measurements are received. However, it was already mentioned that additional knowledge on the sensor value becomes available at instants that *no* new measurement is sampled. The usage of this knowledge was considered by the estimator presented in (Nguyen and Suh, 2007), though the method requires that *Send-on-Delta* is used to trigger the events. Moreover, an asymptotic analysis to prove stability of the estimator is omitted.

Therefore, an open issue in centralized estimation is a stochastic state-estimator suitable for *any* type of event sampling strategy. Beside this generalization, the main aspect of such an estimator are proven asymptotic bounds on the error-covariance, i.e., stability. Stability is an issue in distributed estimation as well, though it is not the only one.

1.3.2 Cooperation in distributed estimation

The distributed estimation strategy depicted in Figure 1.7 is often adopted in large-scale, mobile or ad-hoc sensor networks. Circumstances in these type of networked systems call for a communication set-up where nodes exchange data with their neighboring nodes only, such as a mesh topology. This is done for two reasons: to limit the energy consumption required for exchanging data and to maintain a feasible communication set-up in the presence of system changes¹. Employing a centralized estimator in these type of systems imposes significant communication requirements that are often impractical. A realistic alternative are distributed state-estimators, as they seek for a

¹Broken data connections; newly added nodes to a network during operation.

more efficient use of resources in line with the network topology, i.e., each node relies on its neighboring nodes for computing a local estimate. Further, an inspiring solution for improving the robustness of distributed estimators to system changes and unknown circumstances is found in a colony of ants.

A description of ants, as it is found in (Wikipedia, Ants), demonstrates that their kind creates adaptive colonies that effectively respond to unknown circumstances. In addition, ants are highly successful species: except for the north-pole they inhabit every part of Earth and determine 15 – 25 % of its terrestrial animal biomass. Among others, the success of ants is based on their social organization, which consists of three main social rules:

- Division of labour;
- Communication between individuals;
- Cooperation, and thereby solving complex problems together.

Since ants are successful species, an interesting question is to what extent distributed state estimation has already gained from these three social rules.

A recurring solution in distributed estimation along the first social rule is that each node employs a state-estimator for processing local measurements, e.g., the Kalman filter. As such, each node computes a local estimate of the global state x . See the illustration of Figure 1.7 and the methods proposed in (Speyer, 1979; Hashmipour et al., 1988; Durant-Whyte et al., 1990; Alriksson and Rantzer, 2006; Olfati-Saber, 2009). Some methods continue on this line and estimate a part of the global state in each node, e.g., the solutions presented in (Hassan et al., 1978; Mutambara and H.F., 2000; Khan and Moura, 2007). Nonetheless, most research focusses on estimating the global state x in each node, as there are still open issues related to the other social rules. For example, the second rule on communication, which concerns nodes that are able to interpret the received measurements or estimates shared by neighboring nodes. Current methods impose strict requirements to attain this property, such as admitting linear measurements only or, defining the communication topology on forehand. However, the main challenge is the third social rule of cooperation. Existing solutions focus on optimizing the local estimate per node *individually*. Instead, cooperation is an *emerging behavior*, i.e., it arises from a network of state estimating nodes that interact with each other. As such, any approach that solves the estimation problem cooperatively should be designed from a network point of view rather than from individual nodes. Some existing methods might be considered as a cooperative approach, for example, the methods presented in (Alriksson and Rantzer, 2006; Olfati-Saber, 2009), which synchronize the different local estimates to attain a consensus throughout the network. However, the objective of these consensus methods is still to minimize the estimation error per node individually. Moreover, attaining a consensus is settling for a compromise, i.e., a superficial interaction. Instead, cooperation means the will to assist others, i.e., it involves a profound interaction with other nodes.

Therefore, an open issue in distributed state estimation is a cooperative approach, for which the term *cooperative state estimation* will be used. One of the main challenges is to come up with a clear definition of this profound interaction between nodes that characterizes their will to assist other nodes (cooperation). To that extent, note that each node computes a local estimate of the state. As a result, each node determines an error-covariance locally, see Section 1.2, which is a model for the estimation error of the corresponding local estimate. When nodes assist each other in estimating the state, information from the local estimate of one node in the network will be noticeable in the other nodes. The performance of a local estimate is often analyzed by its error-covariance and thus cooperation should result in a property that the error-covariance of one node will be noticeable in the error-covariance of other nodes in the network. Therefore, a distributed state-estimator is said to be cooperative, if the *global covariance* property is attained.

Property 1.3.1 Global covariance: the error-covariance of each node in the network is a combination of all error-covariances found across its nodes.

The open issues in *cooperative state estimation* is to propose a characterization of global covariance first, after which a distributed estimation strategy should be designed that fulfills this property.

A practical issue in distributed state estimation is deciding which of the available solutions is “most” suitable for a given sensor network application. This issue arises from the fact that a large variety of theoretical and practical contributions are available. Moreover, proposed solutions are scattered in literature, since many research communities are active on this area, such as the control, multi-agent and fusion community. Each community complies on its own assumptions and arguments that are not clearly stated for outsiders, making such decisions more difficult. Often, novel solutions in distributed state estimation are proposed for linear processes with Gaussian distributions, as it allows to focus on assessing the distributed character of the proposed solution. Therefore, many theoretical contributions to this area of *distributed Kalman filtering* are found in the literature, see, for example, (Durant-Whyte et al., 1990; Franken and Hupper, 2005; Alriksson and Rantzer, 2006; Olfati-Saber, 2007; Khan and Moura, 2007; Speranzon et al., 2008; Sawo et al., 2008). Furthermore, some examples of practical implementations are available as well, e.g., (Regazzoni and Tesei, 1996; Vadigepalli and Doyle, 2003; D’Antona et al., 2006; Alriksson and Rantzer, 2007; Di Cairano et al., 2007; Cortés, 2009; Papp et al., 2009). Several survey studies were performed in (Felter, 1990; Hespanha et al., 2007; Ribeiro et al., 2010; Garin and Schenato, 2011) but due to growing amount of distributed Kalman filters these surveys could only address a selective area. Therefore, a comprehensive overview with an in-depth comparison that supports design decisions for employing a particular distributed Kalman filter is still missing.

To summarize, the current challenges on state estimation in networked systems addressed in this thesis are the following:

Problem 1.3.2 A centralized state-estimator that supports any event sampling strategy and has stable estimation results.

Problem 1.3.3 A distributed state-estimator that attains the *global covariance* property.

Problem 1.3.4 A comprehensive overview on distributed Kalman filters that assists in choosing a suitable solution for a sensor network application.

1.4 Outline of the thesis

The above three problems are addressed according to the following outline.

Chapter 2, entitled *event based state estimation*, is based on the articles (Sijs and Lazar, 2009, 2012a). The chapter addresses Problem 1.3.2: *A centralized state-estimator that supports any event sampling strategy and has stable estimation results*. To support any event sampling strategy, a first step is to rewrite the process model in (1.1) from a constant sampling time τ_s into a model that is generalized for any sampling time $\tau \in \mathbb{R}_+$. As such, let $t \in \mathbb{R}_+$ and $t-\tau \in \mathbb{R}_+$ denote the current and preceding sampling instants, respectively. Then, this generalized discrete-time state-space model, yields

$$\begin{aligned} x(t) &= A(\tau)x(t-\tau) + B(\tau)u(t-\tau) + w(t, \tau), \\ y(t) &= Cx(t) + Du(t) + v(t). \end{aligned} \quad (1.4)$$

Basically, the above description could be perceived as a discretized version of a continuous time state-space model $\dot{x} = A_c x + B_c u + w_c$, where

$$A(\tau) := e^{A_c \tau}, \quad B(\tau) := \int_0^\tau e^{A_c \eta} d\eta B_c \quad \text{and} \quad w(t, \tau) := \int_0^\tau e^{A_c \eta} w_c(t-\eta) d\eta.$$

Therein, $A_c \in \mathbb{R}^{n \times n}$ and $B_c \in \mathbb{R}^{n \times l}$ characterize the process dynamics of the continuous model. Further, $w_c(t) \in \mathbb{R}^n$ denotes the process-noise in continuous time, due to which $p(w(t, \tau)) = G(w(t, \tau), 0, Q(\tau))$ of the discrete-time state-space model in (1.4) has a covariance matrix $Q(\tau) \in \mathbb{R}^{n \times n}$ depending on τ . Before is continued with the contribution of Chapter 2, let us point out that $A(\tau)$, $B(\tau)$ and $w(t, \tau)$ are independent on the sampling instants.

Proposition 1.4.1 *The model in (1.4) satisfies $A(\tau_1 + \tau_2) = A(\tau_1)A(\tau_2)$, $B(\tau_1 + \tau_2) = A(\tau_1)B(\tau_2)A^\top(\tau_1) + B(\tau_1)$ and $w(\tau_1 + \tau_2) = A(\tau_1)w(\tau_2) + w(\tau_1)$, for any $t, \tau_1, \tau_2 \in \mathbb{R}_+$.*

In the considered estimation set-up with event sampling, new measurement samples are triggered when an a prior defined event occurs on the sensor

value. As such, the main challenge for event based state estimation is to cope with the fact that the instant of a next event is unknown. Nonetheless, it is often desired that new estimation results are determined synchronously in time rather than at the instants of an event. A typical solution is to perform a prediction on the estimation results of the preceding event instant whenever a synchronous time instant occurs. However, since triggering a new event depends on the sensor value, note that *not* receiving a new measurement sample at these synchronous instants still induces some knowledge on the sensor value. This knowledge will be used by the proposed event based state estimator to curtail runaway estimation errors that are caused by predictions and thereby, obtain stable estimation results. Stability is proven in an asymptotical analysis of the error-covariance. Moreover, a mathematical formulation of event sampling is presented, due to which the proposed estimator is suitable for any event sampling strategy. An object tracking example further demonstrates the applicability of the proposed estimator.

Chapter 3, entitled *event based state estimation in a control loop*, is based on the articles (Marck and Sijs, 2010; Sijs et al., 2010a). The chapter studies a control system, in which the estimation results of the event based state-estimator proposed in Chapter 2 are used as input to a controller that runs synchronously in time. The distinguishing property of this set-up is that stability of the control system is decoupled from the event triggering criteria. A first aspect that is addressed are the variations on estimation results caused by current event sampling strategies, as they further induce variations on the control performance. This issue is solved by proposing a novel event sampling strategy that determines the relevance of new measurements for estimation. A second aspect is designing an integration procedure that interprets the results of the event based estimator for the employed controller (robust MPC). This robust MPC then allows an optimization of the disturbance rejection depending on estimation errors, which is demonstrated in an illustrative control example for various event sampling strategies.

Chapter 4, entitled *state-fusion with unknown correlations: Ellipsoidal intersection*, is based on the articles (Sijs et al., 2010b; Sijs and Lazar, 2012b). The chapter presents novel research on state fusion, a topic that will be used in the proceeding chapters on distributed estimation. It addresses a typical problem of fusing two prior Gaussian estimates into a single fused estimate. A challenge for the considered state fusion objective is that correlation of the prior estimates is unknown. To solve this issue, the proposed solution introduces a novel parametrization, so that a realistic value of the maximum correlation can be characterized. This maximum correlation is then used to fuse those parts of the prior estimates that are guaranteed to be independent, i.e., they are based on unique information that is not available in any of the other parts. Moreover, it results in a distinguishing property of the proposed fusion method that the uncertainty after fusion is reduced.

Chapter 5, entitled *cooperative state estimation*, is based on the articles (Sijs et al., 2011; Sijs and Lazar, 2011a,b). The chapter addresses Problem 1.3.3: *A distributed state-estimator that attains the global covariance property*. The proposed cooperative state-estimator of this chapter makes use of preliminary results presented in (Durant-Whyte et al., 1990). Therein, an alternative implementation of the Kalman filter was derived, which is commonly known as the *Information filter*. This Information filter obtains equivalent estimation results as the original implementation of the Kalman filter illustrated in Figure 1.4 but with a different set of equations, i.e.,

$$\begin{aligned}
 \hat{x}[k^-] &= A\hat{x}[k-1] + Bu[k-1], \\
 P[k^-] &= AP[k-1]A^\top + Q, \\
 P[k] &= (P^{-1}[k^-] + C^\top V^{-1}C)^{-1}, \\
 \hat{x}[k] &= P[k] (P^{-1}[k^-]\hat{x}[k^-] + C^\top V^{-1}(y[k] - Du[k])).
 \end{aligned} \tag{1.5}$$

Nodes employ the above Information filter to compute a local estimate of the global state based on their own local measurement. However, the objective is a distributed state-estimator that results in a *cooperative* emerging behavior, which implies that the *global covariance* property should be satisfied. To that extent, a characterization of Property 1.3.1 (*global covariance*) is proposed first. It is then shown that this property will be fulfilled when nodes share their local estimate with neighboring nodes, after which the state fusion method of Chapter 4 is employed in each node to merge its local estimate obtained by the Information filter with the estimates received from neighboring nodes. Furthermore, stability of the proposed cooperative state-estimator is analyzed by studying the asymptotic behavior of the error-covariance per node. The chapter concludes with a demonstration of cooperative estimation in two realistic case-studies, i.e., a platoon of intelligent vehicles (linear) and tracking shockwaves on a highway (nonlinear). This latter extension of the proposed cooperative estimation approach towards nonlinear processes is supported by substituting the Information filter with a nonlinear estimator.

Chapter 6, entitled *an overview of distributed Kalman filtering*, is based on the articles (Sijs et al., 2008; Sijs and Papp, 2012). The chapter addresses Problem 1.3.4: *A comprehensive overview of distributed Kalman filters that assists in choosing a suitable solution for a sensor network application*. To that end, existing solutions on distributed Kalman filtering from various research communities are studied. Their solutions are divided into four different types of distributed Kalman filtering approaches. The theoretical results of these four approaches are compared in an asymptotic analysis by proposing a unified description of the corresponding algorithms. Moreover, a critical assessment is performed in two real-life inspired sensor networks to fulfill the main objective: provide insight and argumentation for choosing a suitable distributed Kalman filter when deploying a sensor network.

1.5 Notation and definitions

In this section, some basic mathematical notation and standard definitions are recalled that make the manuscript self-contained.

Sets and operations with sets

- \mathbb{R} , \mathbb{R}_+ , \mathbb{Z} and \mathbb{Z}_+ denote the set of real numbers, nonnegative reals, integers and non-negative integers, respectively;
- $\mathbb{Z}_{\geq c}$ and $\mathbb{Z}_{\mathcal{P}}$, for some $c \in \mathbb{R}$ and $\mathcal{P} \subset \mathbb{R}$, denote the sets $\{n \in \mathbb{Z} | n \geq c\}$ and $\mathbb{Z} \cap \mathcal{P}$, respectively;
- For two arbitrary sets $\mathcal{P}_1 \subseteq \mathbb{R}^n$ and $\mathcal{P}_2 \subseteq \mathbb{R}^n$, $\mathcal{P}_1 \cup \mathcal{P}_2$ denotes their union, $\mathcal{P}_1 \cap \mathcal{P}_2$ denotes their intersection, $\mathcal{P}_1 \setminus \mathcal{P}_2$ denotes their set difference and $\mathcal{P}_1 \subseteq \mathcal{P}_2$ denotes “ \mathcal{P}_1 is a subset of, or equal to \mathcal{P}_2 ”;
- For two arbitrary sets $\mathcal{P}_1 \subseteq \mathbb{R}^n$ and $\mathcal{P}_2 \subseteq \mathbb{R}^n$, $c\mathcal{P}_1 \oplus \mathcal{P}_2 := \{x + y | x \in \mathcal{P}_1, y \in \mathcal{P}_2\}$ denotes their Minkowski sum;
- For a set $\mathcal{P} \subseteq \mathbb{R}^n$, $\text{int}(\mathcal{P})$ denotes the interior of \mathcal{P} , $\#\mathcal{P}$ denotes the number of elements of \mathcal{P} , $\text{cl}(\mathcal{P})$ denotes the closure of \mathcal{P} and $\text{Co}(\mathcal{P})$ denotes the convex hull of \mathcal{P} ;
- A polyhedron (or a polyhedral set) in \mathbb{R}^n is a set obtained as the intersection of a finite number of open and/or closed half-spaces;
- A piecewise polyhedral set is a set obtained as the union of a finite number of polyhedral sets;
- For a given set $\mathcal{P} \subset \mathbb{R}$, $\min \mathcal{P} \in \mathcal{P}$ denotes the smallest value that is found across all elements of \mathcal{P} . Similarly, $\max \mathcal{P} \in \mathcal{P}$ denotes the largest value that is found across all elements of \mathcal{P} ;
- For two arbitrary sets $\mathcal{P} \subset \mathbb{R}$ and $\mathcal{S} \subseteq \mathbb{R}$, such that $\mathcal{P} \subset \mathcal{S}$, $\inf \mathcal{P} \in \mathcal{S}$ denotes the smallest value found across all elements of \mathcal{S} that is larger than each element in \mathcal{P} . Similarly, $\sup \mathcal{P} \in \mathcal{S}$ denotes the largest value found across all elements of \mathcal{S} that is smaller than each element in \mathcal{P} ;
- For a given vector $\mu \in \mathbb{R}^n$ and positive definite matrix $\Sigma \in \mathbb{R}^{n \times n}$, let $\mathcal{E}_{\mu, \Sigma} := \{x \mid (x - \mu)^\top \Sigma^{-1} (x - \mu) \leq 1\}$ denote the ellipsoidal sub-level-set characterized by μ and Σ . See, also, Figure 1.9.

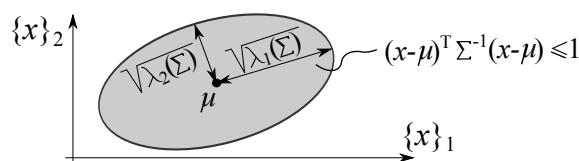


Figure 1.9: An illustrative interpretation of the sub-level-set $\mathcal{E}_{\mu, \Sigma}$.

Vectors, matrices and norms

- 0 denote either the zero number or a vector with all elements equal to zero. Its dimension will be clear from the context. Similarly, $0_{n \times m}$ denotes a $n \times m$ matrix with all elements equal to zero;
- I_n denotes an $n \times n$ identity matrix;
- For some matrices A_1, \dots, A_q , $\text{diag}_{i \in \mathbb{Z}_{[1,q]}}(A_i)$ denotes a diagonal matrix of appropriate dimensions with the matrices A_1, \dots, A_q on the diagonal;
- For some matrices $A_1, \dots, A_q \in \mathbb{R}^{m \times n}$, or vectors if $m = 1$, let us denote $\text{col}_{i \in \mathbb{Z}_{[1,q]}}(A_i) := (A_1^\top \ A_2^\top \ \dots \ A_q^\top)^\top$ as an augmented matrix by arranging A_1, \dots, A_q column-wise;
- For a real number $a \in \mathbb{R}$, $|a|$ denotes its absolute value, whereas for a matrix $A \in \mathbb{R}^{n \times n}$, $|A|$ denotes the determinant;
- The transpose, rank, trace and inverse (when it exists) of an arbitrary matrix A are denoted as A^\top , $\text{rank}(A)$, $\text{tr}(A)$ and A^{-1} , respectively, while $A^{-\top} := (A^\top)^{-1}$;
- For a given A , $\{A\}_{qr} \in \mathbb{R}$ denotes the element on the q -th row and r -th column of A ;
- The Hölder p -norm of a vector $x \in \mathbb{R}^n$ is defined as:

$$\|x\|_p \triangleq \begin{cases} (\{x_1\}^p + \dots + \{x_n\}^p)^{\frac{1}{p}}, & p \in \mathbb{Z}_{[1,\infty)} \\ \max_{i=1,\dots,n} \{x_i\}, & p = \infty, \end{cases}$$

where $\{x\}_i$, $i = 1, \dots, n$ is the i -th component of x , $\|x\|_2$ is also called the Euclidean norm and $\|x\|_\infty$ is also called the infinity (or the maximum) norm. For brevity, let $\|\cdot\|$ denote an arbitrary p -norm;

- Given that $A, B \in \mathbb{R}^{n \times n}$ are positive definite, denoted with $A \succ 0$ and $B \succ 0$ (or $A, B \succ 0$ in short), then $A \succ B$ denotes $A - B \succ 0$. $A \succeq 0$ denotes that A is positive semi-definite;
- For a positive definite and symmetric matrix A , $A^{\frac{1}{2}}$ denotes its Cholesky factor, which satisfies $(A^{\frac{1}{2}})^\top A^{\frac{1}{2}} = A$;
- For a square matrix A , $\lambda_q(A)$, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the q -th, smallest and largest eigenvalue of A , respectively;
- For a matrix A , $\sigma_q(A)$, $\sigma_{\min}(A)$ and $\sigma_{\max}(A)$ denote the q -th and smallest and largest singular value of A , respectively;
- For a signal $x \in \mathbb{R}^n$, $x(t) \in \mathbb{R}^n$ denotes the value of x at the time-instant $t \in \mathbb{R}_+$. In case $x(t)$ is sampled in time, then $x[k] := x(t_k)$ and $x[0:k] := (x(t_0), \dots, x(t_k))$, for some $k \in \mathbb{Z}_+$, where t_k is defined as the time at the k -th sampling instant²;

²Sample instants have different fonts to distinguish them from signals.

- For a signal $x \in \mathbb{R}$, $\dot{x} := \frac{\delta x(t)}{\delta t}$ and $\ddot{x} := \frac{\delta^2 x(t)}{\delta t^2}$;
- A transition-matrix $A(t_2-t_1) \in \mathbb{R}^{n \times m}$ is defined to relate a vector $u(t_1) \in \mathbb{R}^m$ to a vector $x(t_2) \in \mathbb{R}^n$ as follows: $x(t_2) = A(t_2-t_1)u(t_1)$.

Functions and decompositions

- The probability on a certain event A is denoted as $\Pr(A)$;
- A random vector $x \in \mathbb{R}^n$ is characterized by its probability density function (PDF), which is denoted as $p(x)$. Further, a conditional PDF of x given $y \in \mathbb{R}^q$ is denoted as $p(x|y)$;
- For a random vector $x \in \mathbb{R}^n$, $\mathbb{E}[x]$ and $\text{cov}(x)$ denote the expectation and covariance of x , respectively, where

$$\mathbb{E}[x] := \int_{\mathbb{R}^n} xp(x)dx \quad \text{and} \quad \text{cov}(x) := \mathbb{E}[xx^\top] - \mathbb{E}[x]\mathbb{E}[x^\top];$$

- The Gaussian function (Gaussian in short) of the vectors $x, \mu \in \mathbb{R}^n$ and the covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ is defined as $G(x, \mu, \Sigma) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$, i.e., $G(x, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)}$. If $p(x) = G(x, \mu, \Sigma)$, then by definition $\mathbb{E}[x] = \mu$ and $\text{cov}(x) = \Sigma$ hold;
- Following the definition of the Delta-function given in (Solodovnikov, 1960), the element-wise Delta-function $\delta(x)$ of a vector $x \in \mathbb{R}^n$ is a function which vanishes at all values of x except at $x = 0$, it is infinity when $x = 0$ and, moreover, $\int_{\mathbb{R}^n} \delta(x)dx = 1$;
- For a random vector $x \in \mathbb{R}^n$ and a bounded Borel set $\mathcal{P} \subset \mathbb{R}^n$, see, e.g. (Aggoun and Elliot, 2004), the PDF $\Pi_{\mathcal{P}}(x) : \mathbb{R}^n \rightarrow \{0, \varrho\}$, in which $\varrho \in \mathbb{R}$ denotes the Lebesgue measure of \mathcal{P} (Lebesgue, 1902), yields

$$\Pi_{\mathcal{P}}(x) = \begin{cases} 0 & \text{if } x \notin \mathcal{P}, \\ \varrho^{-1} & \text{if } x \in \mathcal{P}; \end{cases}$$

- For given $A \in \mathbb{R}^{n \times n}$ with real eigenvectors, denoted as $\nu_q(A) \in \mathbb{R}^n$, and eigenvalues, i.e., $\lambda_q(A) \in \mathbb{R}$, for all $q \in \mathbb{Z}_{[1,n]}$, the eigenvalue decomposition of A , i.e., $A = SDS^{-1}$, is obtained as $S := (\nu_1(A) \nu_2(A) \dots \nu_n(A))$ and $D := \text{diag}_{q \in \mathbb{Z}_{[1,n]}}(\lambda_q(A))$.

Standard definitions

Definition 1.5.1 Let $A \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{m \times n}$ be given. Then the pair (A, C) is said to be *observable*, if $\text{rank}(C^\top (CA)^\top \dots (CA^{n-1})^\top)^\top = n$.

Definition 1.5.2 Two random vectors $x, y \in \mathbb{R}^n$ are called *independent* if $\text{cov}(x, y) = 0_{n \times n}$. Otherwise, they are *correlated*.

To model the sensor network, consider an undirected, bounded, connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{C})$, where $\mathcal{V} = \{v_1, \dots, v_N\}$ is the set of vertices, for some $N \in \mathbb{Z}_{<\infty}$, and $\mathcal{C} \subseteq (\mathcal{V} \times \mathcal{V})$ is the set of network connections (edges) with (v_i, v_j) the edge from v_i to v_j (if $(v_i, v_j) \in \mathcal{C}$, then $(v_j, v_i) \in \mathcal{C}$). Further, let $\mathcal{N} := \mathbb{Z}_{[1, N]}$ denote the set of node indices, due to which v_i represents node i .

Definition 1.5.3 Let $r, s \in \mathcal{N}$ and a finite, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{C})$ be given. Then a graph path that starts at $v_r \in \mathcal{V}$ and ends at $v_s \in \mathcal{V}$ is a sequence of vertices $\tau_{r,s} = \{v^{(1)}, \dots, v^{(l)}\} \subseteq \mathcal{V}$, where $(v^{(j)}, v^{(j+1)}) \in \mathcal{C}$, for all $j \in \mathbb{Z}_{[1, l-1]}$, and $v^{(1)} = v_r, v^{(l)} = v_s$. Furthermore, the length of the path is $L(\tau_{r,s}) := l$ and $L(\tau_{r,r}) := 0$.

Definition 1.5.4 Let $r, s \in \mathcal{N}$ and a finite, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{C})$ be given. Then the graph distance of $v_r, v_s \in \mathcal{V}$, denoted with $d(v_r, v_s)$, is the length of the shortest path between them, i.e., $d(v_r, v_s) := \min_{\tau_{r,s} \in \mathcal{T}_{r,s}} L(\tau_{r,s})$, where $\mathcal{T}_{r,s}$ is the set of all graph paths from v_r to v_s .

For any $i \in \mathcal{N}$, let $\mathcal{N}_{i(q)} := \{j \in \mathcal{N} | d(v_i, v_j) = q\}$ denote the set of node-indices corresponding to the q -th order neighbors of node i and let $\mathcal{N}_{i(0,1)} := \mathcal{N}_{i(1)} \cup \{i\}$.

Beside estimation, also some control is presented in this thesis. To that extent, let us consider the following discrete-time process model, for some non-linear function $f : \mathbb{R}^n \times \mathbb{R}^l \rightarrow \mathbb{R}^n$ and the k -th sample instant, i.e.,

$$x[k+1] := f(x[k], w[k]), \quad (1.6)$$

Further, an introduced function $\varphi : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ belongs to class \mathcal{K} if it is continuous, strictly increasing and $\varphi(0) = 0$. A function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ belongs to class \mathcal{KL} if for each fixed $k \in \mathbb{R}_+$, $\beta(\cdot, k) \in \mathcal{K}$ and for each fixed $s \in \mathbb{R}_+$, $\beta(s, \cdot)$ is decreasing and $\lim_{k \rightarrow \infty} \beta(s, k) = 0$. Then the following definition of input-to-state stability (ISS) is employed.

Definition 1.5.5 Let $\mathbb{X} \in \mathbb{R}^n$ with $0 \in \text{int}(\mathbb{X})$ and $\mathbb{W} \in \mathbb{R}^m$ be given subsets. Then, system (1.6) is called *ISS in \mathbb{X} for inputs in \mathbb{W}* , if a \mathcal{KL} -function $\beta(\cdot, \cdot)$ and a \mathcal{K} -function $\gamma(\cdot)$ exist such that, for each $x[0] \in \mathbb{X}$ and $w[k] \in \mathbb{W}$, for all $k \in \mathbb{Z}_+$, it holds that all corresponding state trajectories of (1.6) satisfy $\|x[k]\| \leq \beta(\|x[0]\|, k) + \gamma(\|w[0:k]\|), \forall k \in \mathbb{Z}_{\geq 1}$.

Furthermore, $\gamma(\cdot)$ is called the *ISS gain* of system (1.6).

Event based state estimation

2.1	System description	2.4	Asymptotic analysis
2.2	A mathematical formulation of sampling	2.5	Illustrative example
2.3	An event based state-estimator	2.6	A discussion on package loss
		2.7	Conclusions

The next *two* chapters are concerned with event based sampling in a centralized set-up, as it is illustrated in Figure 2.1. Such sampling strategies do not exchange measurements synchronously in time but when an a priori defined event occurs on the sensor value. For clarity of exposition, solutions are presented by considering one sensor node connected to an estimator or controller, instead of the multiple nodes depicted in Figure 2.1.

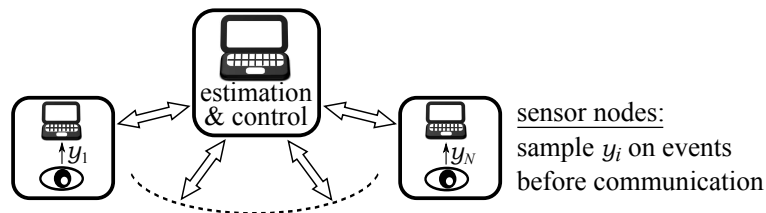


Figure 2.1: A centralized set-up with event based sampling.

This chapter in particular proposes an *event based state-estimator*. The main challenge of such an estimator, due to the fact that the instant of an event is not known in advance, is to prevent a diverging estimation error by exploiting the information from the event sampled measurements. To that extent, a mathematical formulation of event sampling is introduced, which supports a state-estimator with a hybrid update. At the instants of an event the estimated state is updated using the measurement, while at synchronous time instants the update is based on knowledge that the measured value lies within a *bounded* set used to define the event. This bounded set and hybrid update are key for proving asymptotic bounds on the error-covariance (stability). An object tracking example further demonstrates applicability of the proposed algorithm compared to alternative estimators and is further used to discuss realistic solutions for coping with package loss.

2.1 System description

Event based state estimation deals with the set-up depicted in Figure 2.1, i.e., a sensor node exchanges measurements with a centralized state-estimator via a (wireless) data connection. To limit communication requirements and prevent package loss, measurements are exchanged at the instants of a *pre-defined* event rather than synchronously in time, see also Section 1.3.1. The other parts of the state estimation set-up have the following description.

Communication The sensor node employs an event sampling strategy to obtain a measurement $y \in \mathbb{R}^m$ at the event instants $t_e \in \mathbb{R}_+$, where $e \in \mathbb{Z}_+$ denotes the e -th event sample.

Process Let us consider the autonomous, linear process of (1.4), for some sampling interval $\tau \in \mathbb{R}_{>0}$ and $u(t) = 0$, i.e.,

$$x(t) = A(\tau)x(t-\tau) + w(t, \tau), \quad (2.1a)$$

$$y(t) = Cx(t) + v(t). \quad (2.1b)$$

The state vector is denoted as $x \in \mathbb{R}^n$ and both the process-noise $w \in \mathbb{R}^n$ and measurement-noise $v \in \mathbb{R}^m$ are characterized by Gaussian PDFs, for some $Q(\tau) \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{m \times m}$, i.e.,

$$p(w(t, \tau)) := G(w(t, \tau), 0, Q(\tau)) \quad \text{and} \quad p(v(t)) := G(v(t), 0, V).$$

The event measurements y are used to determine an estimate of x according to a Gaussian PDF, i.e., characterized by some mean $\hat{x}[k] \in \mathbb{R}^n$ and error-covariance $P[k] \in \mathbb{R}^{n \times n}$. Usually, values for $\hat{x}[k]$ and $P[k]$ are required synchronously in time rather than at the instants of an events, e.g., as input to a monitoring system or a time synchronous controller. To that extent, let us define $\mathbb{T}_e \subset \mathbb{R}_+$ and $\mathbb{T}_s \subset \mathbb{R}_+$ as the set of time instants that correspond to all events and synchronous instants, respectively. If $\tau_s \in \mathbb{R}_+$ denotes the sampling time of \mathbb{T}_s , then

$$\mathbb{T}_e := \{t_e \in \mathbb{R}^n \mid e \in \mathbb{Z}_+\} \quad \text{and} \quad \mathbb{T}_s := \{s\tau_s \mid s \in \mathbb{Z}_+\}.$$

A definition of the event instants t_e is given in the next section. Notice that it could happen that an event instant coincides with a synchronous instant. Therefore $\mathbb{T}_e \cap \mathbb{T}_s$ might be non-empty. Further, let $\mathbb{T} := \mathbb{T}_e \cup \mathbb{T}_s$ denote the set of all sample instants, i.e., both event and synchronous ones.

The open issue for state estimation, as it is proposed in Problem 1.3.2, is providing *stable* estimation results for any event sampling strategy. Such an estimator allows the sensor node to adopt different event sampling strategies depending on, for example, the expected lifetime of its battery. Moreover, a computationally efficient algorithm of the event based state-estimator (EBSE) is sought for, to attain an applicable solution. Before addressing related work, let us recall that stability refers to a non-divergent error-covariance.

Definition 2.1.1 The EBSE is said to be *stable*, if $\lim_{k \rightarrow \infty} \lambda_q(P[k]) \leq c$ holds for all $q \in \mathbb{Z}_{[1,n]}$ and some constant $c < \infty$.

The main challenge in event based estimation is that the instant of a next event is unknown and could take a long time. Still, estimation results are expected to become available synchronously in time. Existing solutions related to asynchronous estimation perform a prediction of the state at these synchronous time instants, as no measurement is received, see, e.g., (Mallick et al., 2001). It was shown in (Sinopoli et al., 2004) that this would lead to a diverging error-covariance $P[k]$ for long inter-event time periods. However, since triggering a new event depends on the sensor value, note that *not* receiving a new measurement sample at the synchronous instants \mathbb{T}_s still induces some knowledge on $y(t)$, for all $t \in \mathbb{T}_s$. This knowledge can then be used to perform an update on the estimated state at these synchronous instants and thereby, curtail the runaway error-covariance. Although an existing solution that employs this idea was proposed in (Nguyen and Suh, 2007), the considered estimator is especially designed for the event sampling strategy Send-on-Delta and scalar measurements $y \in \mathbb{R}$. See Figure 1.8 for an illustration of Send-on-Delta. Moreover, the asymptotic analysis to prove stability of the estimator is omitted.

Therefore, this chapter presents an EBSE, suitable for any type of event sampling strategy, by starting with a generalized mathematical formulation of sampling. Based on this formulation the EBSE is derived in three steps, followed by an asymptotic analysis of its error-covariance $P[k]$. The EBSE is then compared to alternative estimators, which reduce communication requirements as well, in an illustrative example of object tracking.

2.2 A mathematical formulation of sampling

To present a mathematical description of (event) sampling, let us start with some examples of sampling strategies. The one that is currently well known is sampling synchronously in time, though more recently, asynchronous approaches have emerged as an alternative to time-synchronous sampling. Event based sampling is such an asynchronous approach, for which some examples are Send-on-Delta and Integral sampling, as proposed in (Heemels et al., 1999; Åström and Bernhardsson, 2002; Miskowicz, 2006, 2007), see Figure 1.8. Both strategies assume scalar measurements and define that triggering the next event sample $e \in \mathbb{Z}_+$ depends on $|y(t) - y(t_{e-1})|$ at the current time $t > t_{e-1}$. An extension from scalar to vector measurements is found in (Donkers, 2012), which employs $\|y(t) - y(t_{e-1})\|_2^2 \leq \sigma \|y(t)\|_2^2 + \Delta$ to trigger the next event, for some weight $\sigma \in \mathbb{R}_+$ and threshold $\Delta \in \mathbb{R}_+$.

The above examples indicate that triggering a sample $e \in \mathbb{Z}_+$ can be based on time and measurement values. Therefore, the proposed mathematical formulation of sampling starts by introducing the set $\mathcal{H}[e] \subseteq \mathbb{R}^{m+1}$ in the

time-measurement-space as all allowable values that $y(t)$ may take in between t_{e-1} and t_e . Further, $\begin{pmatrix} y(t_{e-1}) \\ t_{e-1} \end{pmatrix} \in \text{int}(\mathcal{H}[e])$ will ensure that $t_e > t_{e-1}$, i.e., that two different events are not simultaneously triggered. Then, generating the next event instant t_e , given t_{e-1} , yields

$$t_e := \inf \left\{ t \in \mathbb{R}_+ \mid t > t_{e-1} \text{ and } \begin{pmatrix} y(t) \\ t \end{pmatrix} \notin \mathcal{H}[e] \right\}, \quad (2.2)$$

subject to, $\begin{pmatrix} y(t_{e-1}) \\ t_{e-1} \end{pmatrix} \in \text{int}(\mathcal{H}[e])$.

Since $\mathcal{H}[e]$ is used to trigger the event instant t_e , note that $\mathcal{H}[e]$ should be known at t_{e-1} , or at a previous time instant. Next, let us define $\mathcal{H}[e|t] \subset \mathbb{R}^m$ as a section of $\mathcal{H}[e]$ in the *measurement-space* for a fixed time t , i.e.,

$$\mathcal{H}[e|t] := \left\{ y \in \mathbb{R}^m \mid \begin{pmatrix} y \\ t \end{pmatrix} \in \mathcal{H}[e] \right\}.$$

An example of $\mathcal{H}[e]$ and $\mathcal{H}[e|t]$ is depicted in Figure 2.2. The figure shows that the following proposition can be derived for event sampling strategies, for which similar results were obtained in (Lehmann and Lunze, 2010), i.e.,

Proposition 2.2.1 *Let $y(t)$ be sampled according to (2.2), for all $e \in \mathbb{Z}_+$. Then $y(t) \in \mathcal{H}[e|t]$ holds for any time instant $t \in [t_{e-1}, t_e)$.*

Proposition 2.2.1 formalizes the inherent measurement knowledge of event sampling. Moreover, it will be used by the developed EBSE, which will be presented next, to perform an update not only at the instants of an event but also synchronously in time when no new measurement is received.

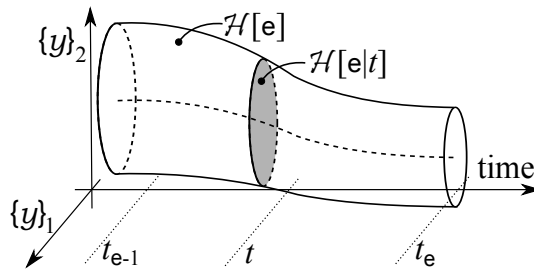


Figure 2.2: An example of $\mathcal{H}[e] \subseteq \mathbb{R}^3$ in the time-measurement-space and a corresponding subset $\mathcal{H}[e|t]$ defining a set in the measurement-space at time $t \geq t_{e-1}$, where $\{y\}_q$ denotes the q -th element of $y \in \mathbb{R}^2$.

Examples of sampling

- Sampling on time refers to strategies where $\mathcal{H}[\mathbf{e}]$ is time dependent. An example is sampling synchronously in time, i.e., Riemann sampling, for which $\mathcal{H}[\mathbf{e}] = \left\{ \begin{pmatrix} y \\ t \end{pmatrix} \in \mathbb{R}^{m+1} \mid t - t_{\mathbf{e}-1} < \tau_s \right\}$ and some $\tau_s \in \mathbb{R}_+$. As $\mathcal{H}[\mathbf{e}]$ is unbounded in the measurement-domain, $y(t)$ can have any value in between two consecutive sample instants, i.e., $\mathcal{H}[\mathbf{e}|t] = \mathbb{R}^m$.
- Sampling on sensor value refers to strategies where $\mathcal{H}[\mathbf{e}]$ depends on measured values. An example is Send-on-Delta, for which $\mathcal{H}[\mathbf{e}] = \left\{ \begin{pmatrix} y \\ t \end{pmatrix} \in \mathbb{R}^{m+1} \mid |y - y(t_{\mathbf{e}-1})| < \Delta \right\}$ and some $\Delta \in \mathbb{R}_{>0}$ (see Figure 1.8). As $\mathcal{H}[\mathbf{e}]$ is unbounded in the time-domain, it can happen that no event will be triggered after some time $t_{\mathbf{e}-1} \in \mathbb{R}_+$.

2.3 An event based state-estimator

Let $\mathbf{k} \in \mathbb{Z}_+$ be defined as the total amount of samples (event and synchronous) until a current time $t > t_{\mathbf{k}}$ with $t_{\mathbf{k}} \in \mathbb{T}$. Further, let $\tau_{\mathbf{k}} := t_{\mathbf{k}} - t_{\mathbf{k}-1}$ denote the corresponding sampling interval depending on \mathbf{k} . Then the EBSE calculates an estimate of $x[\mathbf{k}] := x(t_{\mathbf{k}})$ based on the information that becomes available from the implied measurements $y[q] := y(t_q)$, for all instants $\{t_q \in \mathbb{T} \mid t_q \leq t_{\mathbf{k}}\}$. Note that depending on the type of sample instant, i.e., event or not, this information is either the received measurement $y(t_q)$ in case t_q is an event instant, or it is the inherent knowledge that $y(t_q)$ is bounded by a particular set $H[\mathbf{e}|t_q]$ in case t_q is not an event instant (see Proposition 2.2.1). To exploit both types of measurement information in the EBSE, let us introduce a (bounded) Borel set $\mathcal{Y}[q] \in \mathbb{R}^m$ at each sample instant $t_q \in \mathbb{T}$, such that $y[q] \in \mathcal{Y}[q]$ holds for all $q \leq \mathbf{k}$. Then, employing this concept for state estimation, implies that the PDF computed by the EBSE, yields

$$p\left(x[\mathbf{k}] \mid y[0] \in \mathcal{Y}[0], y[1] \in \mathcal{Y}[1], \dots, y[\mathbf{k}] \in \mathcal{Y}[\mathbf{k}]\right), \quad \forall \mathbf{k} \in \mathbb{Z}_+, \quad (2.3)$$

$$\mathcal{Y}[q] := \begin{cases} H[\mathbf{e}|t_q] & \text{if } t_q \notin \mathbb{T}_{\mathbf{e}}, \{t_{\mathbf{e}-1}, t_{\mathbf{e}} \in \mathbb{T}_{\mathbf{e}} \mid t_{\mathbf{e}-1} < t_q < t_{\mathbf{e}}\}, \\ \{y(t_q)\} & \text{if } t_q \in \mathbb{T}_{\mathbf{e}}. \end{cases} \quad \forall q \in \mathbb{Z}_{[0, \mathbf{k}]}$$

For brevity, the bounded values $y[0] \in \mathcal{Y}[0]$ until $y[\mathbf{k}] \in \mathcal{Y}[\mathbf{k}]$ are denoted as $\mathcal{Y}[0:\mathbf{k}]$, due to which (2.3) becomes $p(x[\mathbf{k}] \mid \mathcal{Y}[0:\mathbf{k}])$. Then Bayes'-rule, as presented in (Mardia et al., 1979), further gives that the desired PDF satisfies

$$p(x[\mathbf{k}] \mid \mathcal{Y}[0:\mathbf{k}]) = \frac{p(x[\mathbf{k}] \mid \mathcal{Y}[0:\mathbf{k}-1]) p(\mathcal{Y}[\mathbf{k}] \mid x[\mathbf{k}])}{\int_{\mathbb{R}^n} p(x[\mathbf{k}] \mid \mathcal{Y}[0:\mathbf{k}-1]) p(\mathcal{Y}[\mathbf{k}] \mid x[\mathbf{k}]) dx[\mathbf{k}]}. \quad (2.4)$$

The prediction $p(x[\mathbf{k}] \mid \mathcal{Y}[0:\mathbf{k}-1])$ in (2.4) is obtained from the preceding estimate according to the results of (Montgomery and Runger, 2007), i.e.,

$$p(x[\mathbf{k}] \mid \mathcal{Y}[0:\mathbf{k}-1]) = \int_{\mathbb{R}^n} p(x[\mathbf{k}] \mid x[\mathbf{k}-1]) p(x[\mathbf{k}-1] \mid \mathcal{Y}[0:\mathbf{k}-1]) dx[\mathbf{k}-1]. \quad (2.5)$$

In brief, the developed EBSE computes $p(x[k]|\mathcal{Y}[0:k])$ of (2.4) at each sample instant $t_k \in \mathbb{T}$ in three steps. Therein, a sum of Gaussians approach is employed to limit processing demand of the EBSE.

1. Formulate the likelihood $p(\mathcal{Y}[k]|x[k])$ as a sum of N Gaussians;
2. Calculate the updated $p(x[k]|\mathcal{Y}[0:k])$ of (2.4) as a sum of N Gaussians;
3. Approximate the resulting $p(x[k]|\mathcal{Y}[0:k])$ as a single Gaussian.

Step 3 is crucial for attaining computational tractability and makes an asymptotic analysis of the EBSE possible, as it will be shown in Section 2.4. Nonetheless, let us start with a detailed derivation of each step in the next three sections, which will make use of the following result.

Proposition 2.3.1 *Let the Gaussians $G(v, Tz, Z)$ and $G(z, \mu, \Sigma)$ be given for some $z, \mu \in \mathbb{R}^n$, $v \in \mathbb{R}^m$, $\Sigma \in \mathbb{R}^{n \times n}$, $T \in \mathbb{R}^{m \times n}$, $Z \in \mathbb{R}^{m \times m}$ and let $\Theta := (\Sigma^{-1} + T^\top Z^{-1}T)^{-1}$, $\theta := \Theta(\Sigma^{-1}\mu + T^\top Z^{-1}v)$. Then it holds that*

$$\int_{\mathbb{R}^n} G(z, \mu, \Sigma) G(v, Tz, Z) dz = G(v, T\mu, T\Sigma T^\top + Z) \quad \text{and} \quad (2.6)$$

$$G(z, \mu, \Sigma) G(v, Tz, Z) = G(z, \theta, \Theta) G(v, T\mu, T\Sigma T^\top + Z). \quad (2.7)$$

See (Gales and Airey, 2006) for a proof of the first claim above, which further yields the second claim.

2.3.1 Step 1: likelihood formulation

This section gives a unified formula of the likelihood $p(\mathcal{Y}[k]|x[k])$ valid for all instants $t_k \in \mathbb{T}$. To that extent, the measurement information $y[k] \in \mathcal{Y}[k]$ is regarded as a quantized measurement. As such, the results of (Mahler, 2011), in which quantized measurements are modeled as a the uniform distribution $p(y[k] \in \mathcal{Y}[k])$ for all $y[k] \in \mathcal{Y}[k]$, indicate that the likelihood corresponding to this quantized measurement, yields

$$p(\mathcal{Y}[k]|x[k]) = \int_{\mathbb{R}^m} p(y[k]|x[k]) p(y[k] \in \mathcal{Y}[k]) dy[k]. \quad (2.8)$$

The first PDF in the above integral, i.e., $p(y[k]|x[k])$, is directly obtained from the process model of (2.1), i.e.,

$$p(y[k]|x[k]) = G(y[k], Cx[k], V). \quad (2.9)$$

To find an expression for the second PDF, i.e., $p(y[k] \in \mathcal{Y}[k])$, let us assume that at t_{k-1} a total amount of $e-1$ events was triggered. Further, recall that $\Pi_{\mathcal{P}}(y)$ is defined as a uniform distribution of $y \in \mathcal{P}$, i.e., the distribution has a constant value within the set $\mathcal{P} \subset \mathbb{R}^m$ and zero outside \mathcal{P} .

Therefore, $\Pi_{\mathcal{Y}[k]}(y[k])$ is the desired uniform distribution of $p(y[k] \in \mathcal{Y}[k])$, which can further be rewritten by substituting $\mathcal{Y}[k]$ of (2.3), as follows:

$$p(y[k] \in \mathcal{Y}[k]) := \begin{cases} \Pi_{\mathcal{H}[e|t_k]}(y[k]) & \text{if } t_k \in \mathbb{T}_s \setminus \mathbb{T}_e, \\ \delta(y[k] - y(t_e)) & \text{if } t_k = t_e \text{ and } t_e \in \mathbb{T}_e. \end{cases} \quad (2.10)$$

For brevity $\Pi_{\mathcal{H}[e|t_k]}(y[k])$ is denoted as $\Pi_{\mathcal{H}}(y[k])$.

The PDFs of (2.9) and (2.10) facilitate in deriving a unified expression of the likelihood $p(\mathcal{Y}[k]|x[k])$ valid for any sampling instant $t_k \in \mathbb{T}$. Let us start this derivation for the case that $t_k \in \mathbb{T}_e$ is an *event instant*. Then, substituting the corresponding PDFs of (2.9) and (2.10) into (2.8), gives that

$$p(\mathcal{Y}[k]|x[k]) = G(y(t_e), Cx[k], V), \quad \text{if } t_k \in \mathbb{T}_e. \quad (2.11)$$

In case $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$ is a *synchronous instant*, please note that the likelihood $p(\mathcal{Y}[k]|x[k])$ is not necessarily Gaussian, as $p(y[k] \in \mathcal{Y}[k]) = \Pi_{\mathcal{H}}(y[k])$ is a uniform distribution depending the event sampling strategy. To have a unified likelihood expression, independent of the event sampling approach, $\Pi_{\mathcal{H}}(y[k])$ is approximated as a summation of N Gaussians, see (Sorenson and Alspach, 1971) for more details. The mean of the q -th Gaussian in this approximation is denoted as $\hat{y}_q[k] \in \mathbb{R}^m$, for all $q \in \mathbb{Z}_{[1, N]}$, and is obtained by equidistant sampling of $\mathcal{H}[e|t_k]$. This supports that each of the N Gaussians has an equivalent covariance matrix, denoted with $U[k] \in \mathbb{R}^{m \times m}$, due to which the approximation is in line with the following characterization:

$$p(y[k] \in \mathcal{Y}[k]) \approx \sum_{q=1}^N \frac{1}{N} G(y[k], \hat{y}_q[k], U[k]), \quad t_k \in \mathbb{T}_s \setminus \mathbb{T}_e. \quad (2.12)$$

Substituting the approximated $p(y[k] \in \mathcal{Y}[k])$ of (2.12) and $p(y[k]|x[k])$ of (2.9) into (2.8), gives for a synchronous instant $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$ that

$$p(\mathcal{Y}[k]|x[k]) \approx \sum_{i=1}^N \frac{1}{N} \int_{\mathbb{R}^m} G(y[k], Cx[k], V) G(y[k], \hat{y}_q[k], U[k]) dy[k].$$

An explicit solution of such an integral was already obtained in Proposition 2.3.1. Hence, substituting $z = y[k]$, $\mu = Cx[k]$, $\Sigma = V$, $v = \hat{y}_q[k]$, $T = I_m$ and $Z = U[k]$ into this proposition, while using the fact that $G(v, Tz, Z) = G(Tz, v, Z)$, induces the following likelihood at synchronous time instants:

$$p(\mathcal{Y}[k]|x[k]) \approx \sum_{q=1}^N \frac{1}{N} G(\hat{y}_q[k], Cx[k], V + U[k]), \quad \forall t_k \in \mathbb{T}_s \setminus \mathbb{T}_e. \quad (2.13)$$

A unified expression for the likelihood $p(\mathcal{Y}[k]|x[k])$ of (2.8) can thus be

established as the combination of (2.11) and (2.13), i.e.,

$$p(\mathcal{Y}[\mathbf{k}]|x[\mathbf{k}]) \approx \sum_{q=1}^N \frac{1}{N} G(\hat{y}_q[\mathbf{k}], Cx[\mathbf{k}], R[\mathbf{k}]), \quad \forall t_{\mathbf{k}} \in \mathbb{T} \quad (2.14)$$

where, $R[\mathbf{k}] := V + U[\mathbf{k}]$.

At an event instant $t_{\mathbf{k}} \in \mathbb{T}_e$ the EBSE receives a new measurement $y(t_e)$ and the variables of (2.14) become $N = 1$, $\hat{y}^1[\mathbf{k}] = y(t_e)$ and $U[\mathbf{k}] = 0_{m \times m}$. At a synchronous instant $t_{\mathbf{k}} \in \mathbb{T}_s \setminus \mathbb{T}_e$ the variables depend on the approximation of $\Pi_{\mathcal{H}}(y[\mathbf{k}])$ and thus on the employed event sampling strategy. Let us first present a small example of such an approximation for the event sampling strategy Send-on-Delta. After that, the explanation of the EBSE is continued with step 2 based on the likelihood of (2.14).

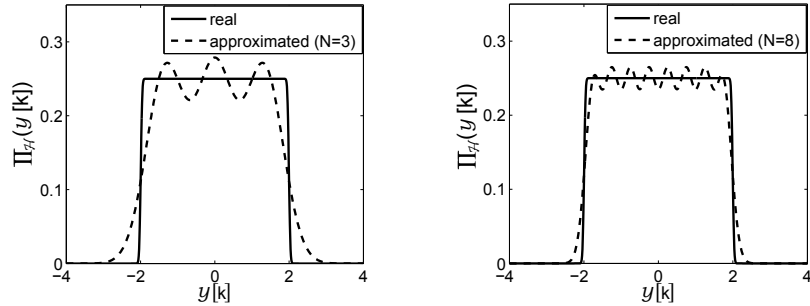
Example of approximating $\Pi_{\mathcal{H}}(y[\mathbf{k}])$ for Send-on-Delta

Let us assume that the EBSE receives new measurements according to the event sampling strategy Send-on-Delta. Then an approximation of $\Pi_{\mathcal{H}}(y[\mathbf{k}])$ in (2.12) $t_{\mathbf{k}} \in \mathbb{T}_s \setminus \mathbb{T}_e$, which was established for some $N \in \mathbb{R}_{>0}$ by trial-and-error, is given with

$$\hat{y}_q[\mathbf{k}] = y(t_{e-1}) - \left(\frac{N - 2(q-1) - 1}{2N} \right) 2\Delta, \quad \forall q \in \mathbb{Z}_{[1,N]} \quad \text{and}$$

$$U[\mathbf{k}] = \left(\frac{2\Delta}{N} \right)^2 \left(0.25 - 0.05e^{-\frac{4(N-1)}{15}} - 0.08e^{-\frac{4(N-1)}{180}} \right).$$

Figure 2.2 illustrates such an approximation of $\Pi_{\mathcal{H}}(y[\mathbf{k}])$ into $\sum_{q=1}^N \frac{1}{N} G(y[\mathbf{k}], \hat{y}_q[\mathbf{k}], U[\mathbf{k}])$ according to the above formulas.



(a) Approximation by 3 Gaussians (b) Approximation by 8 Gaussians

Figure 2.3: Illustration for approximating $\Pi_{\mathcal{H}}(y[\mathbf{k}])$ into a summation of Gaussians, when Send-on-Delta is employed as event sampling strategy, i.e., $\mathcal{H}[e] := \{y \in \mathbb{R} | |y - y(t_{e-1})| < \Delta\}$, for which $y(t_{e-1}) = 0$ and $\Delta = 2$.

2.3.2 Step 2: state estimation

Step 2 of the EBSE first determines the product $p(x[k]|\mathcal{Y}[0:k-1])p(\mathcal{Y}[k]|x[k])$, which is then used to establish the updated estimate $p(x[k]|\mathcal{Y}[0:k])$ of (2.4).

The product $p(x[k]|\mathcal{Y}[0:k-1])p(\mathcal{Y}[k]|x[k])$ is a multiplication of the predicted PDF $p(x[k]|\mathcal{Y}[0:k-1])$ with the likelihood computed by step 1 of the EBSE in (2.14). As such, let us focus on finding a solution for the predicted PDF $p(x[k]|\mathcal{Y}[0:k-1])$. To that extent, recall that an expression of this PDF was obtained in (2.5) as the integral of $p(x[k]|x[k-1])p(x[k-1]|\mathcal{Y}[0:k-1])$ over $x[k-1]$. This latter product is a multiplication of two new PDFs, for which the first PDF $p(x[k]|x[k-1])$ directly results from the process model in (2.1a), i.e., $p(x[k]|x[k-1]) = G(x[k], A(\tau_k)x[k-1], Q(\tau_k))$. The other PDF in this latter product denotes the estimation results of the EBSE at t_{k-1} , i.e., $p(x[k-1]|\mathcal{Y}[0:k-1]) = G(x[k-1], \hat{x}[k-1], P[k-1])$. Hence, substituting these two Gaussian PDFs into the integral of (2.5), gives that

$$\begin{aligned} p(x[k]|\mathcal{Y}[0:k-1]) \\ = \int_{\mathbb{R}^n} G(x[k-1], \hat{x}[k-1], P[k-1]) G(x[k], A(\tau_k)x[k-1], Q(\tau_k)) dx[k-1]. \end{aligned}$$

An explicit solution for the above expression of the predicted PDF is obtained by employing Proposition 2.3.1. Hence, substituting $z = x[k-1]$, $\mu = \hat{x}[k-1]$, $\Sigma = P[k-1]$, $v = x[k]$, $T = A(\tau_k)$ and $Z = Q(\tau_k)$ into this proposition, implies that the predicted PDF, yields

$$\begin{aligned} p(x[k]|\mathcal{Y}[0:k-1]) &= G(x[k], \hat{x}[k^-], P[k^-]), \\ \text{where } \hat{x}[k^-] &:= A(\tau_k)\hat{x}[k-1], \\ P[k^-] &:= A(\tau_k)P[k-1]A^\top(\tau_k) + Q(\tau_k). \end{aligned} \tag{2.15}$$

The notion k^- emphasizes the predictive character of $\hat{x}[k^-]$ and $P[k^-]$ as the estimation results of $x[k]$.

Now that both $p(x[k]|\mathcal{Y}[0:k-1])$ of (2.15) and $p(\mathcal{Y}[k]|x[k])$ of (2.14) are available, an expression of their product is derived, i.e.,

$$p(x[k]|\mathcal{Y}[0:k-1])p(\mathcal{Y}[k]|x[k]) \approx \sum_{q=1}^N \frac{1}{N} G(x[k], \hat{x}[k^-], P[k^-]) G(\hat{y}_q[k], Cx[k], R[k]).$$

The solution to a product of two Gaussians is presented in Proposition 2.3.1. Hence, substituting $z = x[k]$, $\mu = \hat{x}[k^-]$, $\Sigma = P[k^-]$, $v = \hat{y}_q[k]$, $T = C$ and

$Z = R[k]$ into this proposition results in

$$p(x[k]|\mathcal{Y}[0:k-1])p(\mathcal{Y}[k]|x[k]) \approx \sum_{q=1}^N \frac{1}{N} \omega_q[k] G(x[k], \hat{\theta}_q[k], \Theta[k]), \quad (2.16a)$$

$$\begin{aligned} \text{where } \Theta[k] &:= (P^{-1}[k^-] + C^\top R^{-1}[k]C)^{-1}, \\ \hat{\theta}_q[k] &:= \Theta[k] (P^{-1}[k^-] \hat{x}[k^-] + C^\top R^{-1}[k] \hat{y}_q[k]), \\ \omega_q[k] &:= G(\hat{y}_q[k], C \hat{x}[k^-], CP[k^-]C^\top + R[k]). \end{aligned} \quad (2.16b)$$

The above expressions of $\hat{\theta}_q[k]$ and $\Theta[k]$ are similar to the update formulas of the Kalman filter in information form, as presented in (1.5). As such, each individual $G(x[k], \hat{\theta}_q[k], \Theta[k])$, for all $q \in \mathbb{Z}_{[1,N]}$, can be regarded as Gaussian PDF that represents an updated estimate of $x[k]$ based on the *implied* measurement $\hat{y}_q[k] \in \mathcal{Y}[k]$. Moreover, the product in (2.16a) of the predicted PDF with the likelihood will be used next, to find the updated estimation results of the EBSE, i.e., $p(x[k]|\mathcal{Y}[0:k])$ of (2.4). Note that this updated PDF has the expression of (2.16a) in its nominator, while its denominator is the integral of (2.16a) over $x[k]$. As this integral equals $\sum_{q=1}^N \frac{1}{N} \omega_q[k]$, the explicit formula of $p(x[k]|\mathcal{Y}[0:k])$ computed by the EBSE, yields

$$p(x[k]|\mathcal{Y}[0:k]) \approx \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} G(x[k], \theta_q[k], \Theta[k]). \quad (2.17)$$

The PDF in (2.17) is a summation of N Gaussians that represents the updated estimation results of the EBSE and thus completes step 2. The goal of step 3 is to approximate this PDF into a single Gaussian and thereby, prevent an exponential increase in the computational requirements of the EBSE. If this step would not be done, and the number of Gaussians that determine $p(\mathcal{Y}[k]|x[k])$ is denoted with $N[k]$, then $p(x[k]|\mathcal{Y}[0:k])$ of (2.17) would be described with $\prod_{q \in \mathbb{Z}_{[1,k]}} N[q]$ Gaussians.

2.3.3 Step 3: state approximation

Step 3 of the EBSE approximates $p(x[k]|\mathcal{Y}[0:k])$ of (2.17) from a summation of N Gaussians into the single Gaussian $p(x[k]|\mathcal{Y}[0:k]) \approx G(x[k], \hat{x}[k], P[k])$, where $\hat{x}[k] \in \mathbb{R}^n$ and $P[k] \in \mathbb{R}^{n \times n}$. This is done, such that the mean and error-covariance of the two PDFs are equivalent, for which the results of (Kotecha and Djurić, 2003) are used.

The expressions of $\hat{x}[k]$ and $P[k]$ that were obtained from this approximation are presented in the complete algorithm of the EBSE. Therein, the formulation routine corresponding to step 1 is denoted as “Likelihood(\cdot, \cdot, \cdot)”. This routine establishes $R[k]$ and $\hat{y}_q[k]$, for all $q \in \mathbb{Z}_{[1,N]}$, that characterize the sum of Gaussians as presented in (2.14). Also, notice that $\omega_q[k] \in \mathbb{R}_+$ is

by definition a scalar weight.

Algorithm of the EBSE

Prediction

$$\hat{x}[k^-] = A(\tau_k)\hat{x}[k-1];$$

$$P[k^-] = A(\tau_k)P[k-1]A^\top(\tau_k) + Q(\tau_k);$$

Measurement update

$$(\hat{y}_1[k], \dots, \hat{y}_N[k], R[k]) = \text{Likelihood}(\mathcal{H}[e|t_k], V), \quad (2.14);$$

$$\Theta[k] = (P^{-1}[k^-] + C^\top R^{-1}[k]C)^{-1};$$

for all $q \in \mathbb{Z}_{[1, N]}$, do:

$$\hat{\theta}_q[k] = \Theta[k] (P^{-1}[k^-]\hat{x}[k^-] + C^\top R^{-1}[k]\hat{y}_q[k]);$$

$$\omega_q[k] = G(\hat{y}_q[k], C\hat{x}[k^-], CP[k^-]C^\top + R[k]);$$

end

Approximation

$$\hat{x}[k] = \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} \hat{\theta}_q[k];$$

$$P[k] = \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} \left(\Theta[k] + (\hat{x}[k] - \hat{\theta}_q[k]) (\hat{x}[k] - \hat{\theta}_q[k])^\top \right).$$

The above EBSE algorithm supports any sampling strategies that is in line with (2.2). Let us continue with stability analysis of the EBSE.

2.4 Asymptotic analysis

This section presents an asymptotic analysis of the error-covariance $P[k]$, as computed in the above EBSE algorithm, where $P[\infty] := \lim_{k \rightarrow \infty} P[k]$. For clarity of the analysis, some preliminary definitions are presented first.

2.4.1 Preliminaries

Let $\mathbf{a}(k) := \{a \in \mathbb{Z}_{[0, k]} | t_k - t_{k-a} \leq \tau_s \text{ and } t_{k-a} \in \mathbb{T}_s\}$ denote the number of sample instants in between the current instant $t_k \in \mathbb{T}$ and the first preceding synchronous instant $t_{k-\mathbf{a}(k)} \in \mathbb{T}_s$. Further, as the singular value $\sigma_{\max}(A_\tau)$ and eigenvalue $\lambda_{\max}(Q_\tau)$ are continuous in τ , let us define the constants

$$\alpha(\tau_s) := \sup_{\tau \in [0, \tau_s]} \sigma_{\max}(A(\tau)), \quad \beta(\tau_s) := \sup_{\tau \in [0, \tau_s]} \sqrt{\lambda_{\max}(Q(\tau))}.$$

Next, the vectors $\hat{y}^-[\mathbf{k}] \in \mathbb{R}^l$ and $\hat{y}^+[\mathbf{k}] \in \mathbb{R}^l$ are obtained via the extreme elements within $\hat{y}_q[\mathbf{k}]$, for all $q \in \mathbb{Z}_{[1,N]}$, as follows:

$$\{\hat{y}^-[\mathbf{k}]\}_r := \min_{\forall q \in \mathbb{Z}_{[1,N]}} \{\hat{y}_q[\mathbf{k}]\}_r \text{ and } \{\hat{y}^+[\mathbf{k}]\}_r := \max_{\forall q \in \mathbb{Z}_{[1,N]}} \{\hat{y}_q[\mathbf{k}]\}_r, \quad \forall r \in \mathbb{Z}_{[1,m]}.$$

Note that both $\hat{y}^-[\mathbf{k}]$ and $\hat{y}^+[\mathbf{k}]$ can be determined from the set $\mathcal{H}[\mathbf{e}|t_{\mathbf{k}}]$ that is already available, see also the illustration in Figure 2.4. Based on $\hat{y}^-[\mathbf{k}]$ and $\hat{y}^+[\mathbf{k}]$, let the weight $\varsigma[\mathbf{k}] \in \mathbb{R}_+$ be defined as follows:

$$\varsigma[\mathbf{k}] := \begin{cases} (\hat{y}^-[\mathbf{k}] - \hat{y}^+[\mathbf{k}])^\top (\hat{y}^-[\mathbf{k}] - \hat{y}^+[\mathbf{k}]) \lambda_{\min}^{-1}(R[\mathbf{k}]) + 1 & \text{if } t_{\mathbf{k}} \in \mathbb{T}_s \setminus \mathbb{T}_e \\ 1 & \text{if } t_{\mathbf{k}} \in \mathbb{T}_e. \end{cases}$$

Proposition 2.4.1 *Let a bounded set $\mathcal{H}[\mathbf{e}|t_{\mathbf{k}}]$ be given for all $\mathbf{k} \in \mathbb{Z}_+$. Then, there exist a matrix $R_{\max} \in \mathbb{R}^{m \times m}$ and a weight $\varsigma_{\max} \in \mathbb{R}_{>0}$, such that $R[\mathbf{k}] \preceq R_{\max}$ and $\varsigma[\mathbf{k}] \leq \varsigma_{\max}$ hold for all $\mathbf{k} \in \mathbb{Z}_+$.*

Proposition 2.4.1 follows from the fact that a bounded $\mathcal{H}[\mathbf{e}|t_{\mathbf{k}}]$ allows $\Pi_{\mathcal{H}}(y[\mathbf{k}])$ to be approximated with a limited amount of N Gaussians and a bounded $U[\mathbf{k}]$, i.e., R_{\max} existst. Furthermore, a bounded $\mathcal{H}[\mathbf{e}|t_{\mathbf{k}}]$ ensures a bounded Euclidean distance $\|\hat{y}^-[\mathbf{k}] - \hat{y}^+[\mathbf{k}]\|_2$, due to which there exists a $\varsigma_{\max} \geq \varsigma[\mathbf{k}]$.

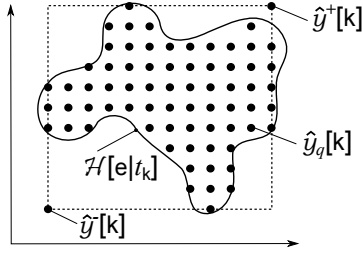


Figure 2.4: An illustration of $\hat{y}^-[\mathbf{k}]$ and $\hat{y}^+[\mathbf{k}]$ for a particular $\mathcal{H}[\mathbf{e}|t_{\mathbf{k}}]$.

2.4.2 Asymptotic analysis of the error-covariance

The asymptotic analysis makes use of two Riccati difference equations in information form apart from the EBSE. The employed equations are:

- The update of $P[\mathbf{k}]$ according to the proposed EBSE;
- A *weighted* Riccati difference equation (wRDE), which performs an update of its covariance matrix $\Sigma[\mathbf{k}] \in \mathbb{R}^{n \times n}$ at all synchronous instants $t[\mathbf{k}] \in \mathbb{T}_s$ by employing R_{\max} and ς_{\max} (see Proposition 2.4.1). Since $t_{\mathbf{k}-a(\mathbf{k})}$ denotes the preceding synchronous instant, the update yields

$$\begin{aligned} \Sigma[\mathbf{k}^-] &:= A(\tau_s)\Sigma[\mathbf{k}-a(\mathbf{k})]A^\top(\tau_s) + Q(\tau_s) \text{ and} \\ \Sigma[\mathbf{k}] &:= \varsigma_{\max} (\Sigma^{-1}[\mathbf{k}^-] + C^\top R_{\max}^{-1} C)^{-1}, \quad \forall t_{\mathbf{k}} \in \mathbb{T}_s; \end{aligned} \quad (2.18)$$

- A hybrid Riccati difference equation (hRDE), which performs a similar update as the wRDE at synchronous sample instants $t_k \in \mathbb{T}_s$, although a prediction is employed at unique event instant $t_k \in \mathbb{T}_e \setminus \mathbb{T}_s$. Hence, the covariance matrix of the hRDE, denoted with $\bar{\Sigma}[k] \in \mathbb{R}^{n \times n}$, is updated as follows:

$$\begin{aligned} \bar{\Sigma}[k^-] &:= A(\tau_k)\bar{\Sigma}[k-1]A^\top(\tau_k) + Q(\tau_k) \text{ and} \\ \bar{\Sigma}[k] &:= \begin{cases} \varsigma_{\max}(\bar{\Sigma}^{-1}[k^-] + C^\top R_{\max}^{-1}C)^{-1} & \text{if } t_k \in \mathbb{T}_s, \\ \bar{\Sigma}[k^-] & \text{if } t_k \in \mathbb{T}_e \setminus \mathbb{T}_s. \end{cases} \end{aligned} \quad (2.19)$$

The asymptotic analysis proves that $\lambda_{\max}(P[\infty])$ is bounded by starting with three lemmas that relate $P[k]$ of the EBSE to $\Sigma[k]$ of the wRDE. The first lemma provides a property of $P[k]$.

Lemma 2.4.2 *The proposed EBSE satisfies $P[k] \preceq \varsigma[k]\Theta[k]$, for all $t_k \in \mathbb{T}$.*

See Appendix B.1 for the proof. In the next lemma, the EBSE is compared with the hRDE. Both methods perform an update of the covariance, although for the hRDE they are limited to synchronous sample instants. Moreover, if the hRDE performs an update, its covariance is scaled according to ς_{\max} such that it “encloses” the one of the EBSE.

Lemma 2.4.3 *Consider the proposed EBSE and the hRDE of (2.19) and let $P[0] = \bar{\Sigma}[0]$. Then, $P[k] \preceq \bar{\Sigma}[k]$ holds for all $t_k \in \mathbb{T}$.*

See Appendix B.2 for the proof. Notice that Lemma 2.4.3 is useful because it relates P of the EBSE, which is an approximated covariance of a sum of Gaussians, to the hRDE. However, the hRDE has a hybrid prediction-update strategy. As such, determining an asymptotic bound on $\bar{\Sigma}[k]$ is a challenging task. The wRDE on the other hand performs a constant update for which determining an asymptotic bound is tackled. Therefore the third lemma relates the hRDE with the wRDE.

Lemma 2.4.4 *Consider the hRDE of (2.19) and the wRDE of (2.18). At synchronous instants $t_k \in \mathbb{T}_s$ it holds that $\lambda_{\max}(\bar{\Sigma}[k]) = \lambda_{\max}(\Sigma[k])$, while $\lambda_{\max}(\bar{\Sigma}[k]) \leq \alpha^2(\tau_s)\lambda_{\max}(\Sigma[k-\mathbf{a}(k)]) + \beta^2(\tau_s)$ holds at the events $t_k \in \mathbb{T}_e \setminus \mathbb{T}_s$.*

See the Appendix B.3 for the proof. Although Lemma 2.4.3 guarantees that $P[k] \preceq \bar{\Sigma}[k]$ holds, Lemma 2.4.4 shows that $\bar{\Sigma}[k]$ is asymptotically bounded if and only if $\lim_{k \rightarrow \infty} \Sigma[k]$ exists. To that extent, the asymptotic properties of the standard Riccati difference equation (RDE), as they were established in (Chan et al., 1984), are employed. More specifically, the conditions were presented under which a solution of the discrete-time Algebraic Riccati equation exists. Since the standard RDE is similar to the wRDE when substituting $\varsigma_{\max} = 1$, the asymptotic solution for the covariance matrix of the wRDE is obtained *mutatis mutandis* from the results presented in (Chan et al., 1984).

As such, let us define $\bar{A} := \sqrt{\varsigma_{\max}} A(\tau_s) (I_n - \Phi C^\top (C \Phi C^\top + R_{\max})^{-1} C)$ and $\bar{\Phi} := \varsigma_{\max} A(\tau_s) \Sigma[\infty] A^\top(\tau_s) + Q(\tau_s)$, for some $\Sigma[\infty] \in \mathbb{R}^{n \times n}$.

Proposition 2.4.5 *Let $(A(\tau_s), C)$ be an observable pair. Then there exists a stabilizing solution $\Sigma[\infty] := \lim_{k \rightarrow \infty} \Sigma[k]$ to the wRDE of (2.18) that is unique and independent of $\Sigma[0]$, if $\lambda_q(\bar{A}) \leq 1$ for all $q \in \mathbb{Z}_{[1, n]}$.*

Now, combining Lemma 2.4.3, Lemma 2.4.4 and Proposition 2.4.5, while observing that $\alpha^2(\tau_s) \geq 1$ and $\beta^2(\tau_s) \geq 1$, directly proves the upper bound on all eigenvalues of $P[k]$ of the EBSE.

Theorem 2.4.6 *Let the premise of Proposition 2.4.1 and Proposition 2.4.5 be satisfied. Then the EBSE results in a stable estimate, i.e., $\lambda_{\max}(P[\infty]) \leq \alpha^2(\tau_s) \lambda_{\max}(\Sigma[\infty]) + \beta^2(\tau_s)$ exists.*

Notice that a simple, yet effective idea of updating the EBSE at its synchronous time instants with a bounded measurement set turns out to be sufficient for the derivation of an asymptotic bound on its error-covariance. This desirable property is otherwise known to be very difficult to attain for event based estimators in general. Most currently available analyses are limited to synchronous estimators, such as the Kalman filter, with an extension to package loss in (Sinopoli et al., 2004). Furthermore, the developed results of this section trivially applies to the state-estimator that is proposed in (Nguyen and Suh, 2007) as a particular case of the EBSE, i.e., the approximation of (2.14) is determined with $N = 1$ for all $k \in \mathbb{Z}_+$ and Send-on-Delta is employed as the event sampling strategy. The theoretical analysis of the proposed EBSE is verified in an illustrative example of object tracking.

2.5 Illustrative example

The effectiveness of the developed EBSE is illustrated in terms of estimation error and computational tractability for a 1D object tracking system.

Process The considered model in line with (2.1) is a double integrator, i.e.,

$$\begin{aligned} x(t) &= \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix} x(t-\tau) + \begin{pmatrix} \frac{1}{2}\tau^2 \\ \tau \end{pmatrix} a(t-\tau) \\ y(t) &= (1 \quad 0) x(t) + v(t). \end{aligned}$$

The state vector $x(t)$ combines the object's position and speed. Further, $a(t)$ denotes the object's acceleration, while only the position is measured in $y(t)$. Acceleration is assumed unknown. Therefore, the model of (2.1) is characterized with a process-noise $w(t, \tau) := (\frac{1}{2}\tau^2) a(t)$. As $|a(t)| \leq 0.5$, for all $t \in \mathbb{R}_+$, a suitable covariance in line with (Curry, 1970) is $\text{cov}(a(t)) = 0.02$ and thus $Q(\tau) = (\frac{1}{2}\tau^2) 0.02 (\frac{1}{2}\tau^2 \quad \tau)$.

The sampling time is $\tau_s = 0.1$ seconds and the measurement-noise covariance is $V = 0.1 \cdot 10^{-3}$. The object's true position, speed and acceleration are depicted in Figure 2.5.

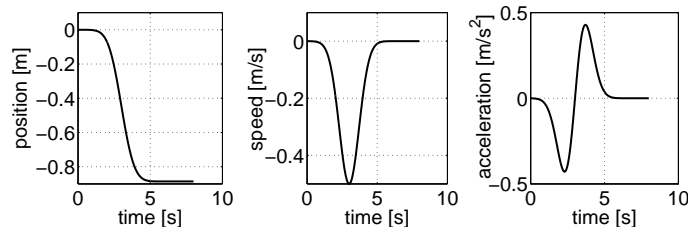


Figure 2.5: The position, speed and acceleration of the tracked object.

Three different state-estimators are compared. The first two are the proposed EBSE and the asynchronous Kalman filter (AKF) of (Mallick et al., 2001). For those estimators, the sensor employs the event sampling strategy Send-on-Delta to trigger $y(t_e)$, i.e., $\mathcal{H}[e] = \left\{ \begin{pmatrix} y \\ t \end{pmatrix} \in \mathbb{R}^{m+1} \mid |y - y(t_{e-1})| < \Delta \right\}$ and $\Delta = 0.1$. Values of $R[k]$ and $\hat{y}_q[k]$ at a synchronous sample instant $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$, for all $q \in \mathbb{Z}_{[1, N]}$, are computed by the EBSE according to the approximation example of Send-on-Delta in Section 2.3.1 and $N = 5$. In contrast to the EBSE, the AKF performs an update at the event instants $t_k \in \mathbb{T}_e$ only, i.e., when a new measurement is received. At the remaining synchronous instants $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$, the AKF performs a prediction by applying the formula of (2.15). Notice that the AKF can be viewed as an estimator with intermittent observations, for which the analysis in (Sinopoli et al., 2004; Mo and Sinopoli, 2008) already showed that its error-covariance will have a diverging behavior. The third estimator is the time-synchronous quantized Kalman filter (QKF) introduced in (Curry, 1970). The QKF performs the Kalman filtering algorithm of Section 1.2 on quantized measurements of $y(t)$, for all $t \in \mathbb{T}_s$, in which the quantization level is equal to 0.1. Quantization of the measurement value reduces communication requirements as well and can thus be considered as an alternative to the EBSE.

Figure 2.6 depicts the squared estimation error of the three estimators. Therein, the EBSE and AKF computed their estimation results based on 9 event measurement samples, while the QKF received 80 quantized measurement samples. The figure shows that the QKF estimates the object's position with the least error. However, its error in speed is worse compared to the EBSE. Furthermore, the plot of the AKF clearly shows that prediction of the state gives an exponential growth in estimation error when the time between the event sample instants increases, e.g., $t > 4$ [s]. This behavior is not present in the results of the EBSE that shows a converging behavior, due to which the EBSE has stable estimation results.

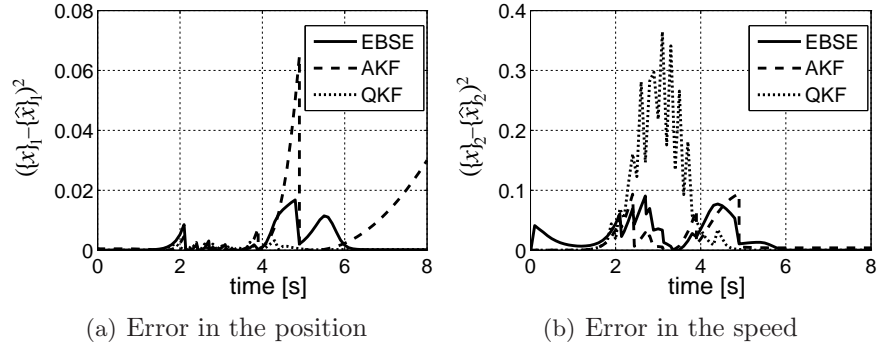


Figure 2.6: The squared estimation error of the two state elements for each employed estimator, i.e., $(\{x(t_k)\}_1 - \{\hat{x}(t_k)\}_1)^2$ and $(\{x(t_k)\}_2 - \{\hat{x}(t_k)\}_2)^2$.

Another important aspect of the three estimators is their total amount of required processing time. The EBSE algorithm indicates that its “measurement update” employs the same set of equations as the corresponding part of the Kalman filter, for each of the N Gaussians. Therefore, a rule of thumb is that the EBSE will require N times the amount of processing time of a Kalman filtering algorithm. Since the QKF performs such an algorithm, with a particular model on the measurement-noise, the EBSE of this illustrative example will roughly cost about $N = 5$ times more processing time than the QKF. After running all three algorithms in Matlab on an Intel[®] Pentium[®] processor of 1.86 [GHz] with 504 [MB] of RAM the following performance results were obtained. The AKF estimated $x[k]$, for all $t_k \in \mathbb{T}_e$, and predicted $x[k]$, for all $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$, in a total time of 0.016 seconds, whereas the QKF estimated $x[k]$, for all $t_k \in \mathbb{T}_s$, in 0.022 seconds of processing time. The EBSE required 0.094 seconds ($\approx 5 \times 0.022 = 0.11$) to estimate $x[k]$ for all $t_k \in \mathbb{T}$.

Therefore, although the EBSE requires most processing time, it is still computationally comparable to the AKF and QKF. Furthermore, the EBSE provides an estimation error similar to that of the QKF but with significantly less data transmission. However, one cannot presume that event sampling will always yield fewer measurement samples, as it depends on the application at hand and the employed sampling strategy. Nonetheless, event sampling is a more efficient approach for acquiring relevant measurements. Moreover, the EBSE is currently the only stable estimator that is not specialized for one event sampling strategy only.

2.6 A discussion on package loss

An issue for the considered set-up of the EBSE in Figure 2.1 is package loss, which is a typical drawback for any data connection. One of the main con-

sequences of package loss is that it may cause unstable estimation results, as it was shown in an asymptotic analysis of the Kalman filter presented in (Sinopoli et al., 2004). When a package is lost, the Kalman filter must perform a prediction on its estimation results, as there is no measurement value for updating the estimated state. Predictions lead to a diverging error of the estimator, which was analyzed in (Sinopoli et al., 2004). The analysis showed that there exists an upper bound on the probability of package loss, after which the error-covariance of the Kalman filter will diverge and thus be unstable. It is worth mentioning that the EBSE algorithm can still guarantee stable estimation results in the presence of package loss, when a particular event sampling strategy is employed. Such a strategy is presented in this discussion, after which the object tracking example of the previous section is employed to assess the estimation results.

The explanation of the proposed event sampling strategy starts with the event triggering condition, as it was established in (2.2), i.e.,

$$t_e = \inf \left\{ t \in \mathbb{R}_+ \mid t > t_{e-1} \text{ and } \begin{pmatrix} y(t) \\ t \end{pmatrix} \notin \mathcal{H}[e] \right\},$$

given that $\begin{pmatrix} y(t_{e-1}) \\ t_{e-1} \end{pmatrix} \in \text{int}(\mathcal{H}[e])$. The main idea of the proposed sampling strategy is to extend the Send-on-Delta approach towards multiple dimensions. As such, let us define the triggering set $\mathcal{H}[e] := \left\{ \begin{pmatrix} y \\ t \end{pmatrix} \mid \|y - y[e-1]\|_2 \leq \Delta \right\}$, due to which its corresponding section $\mathcal{H}[e|t]$ in the measurement-space at any time t is a ball with radius Δ , centered at the previously sampled measurement $y[e-1]$, i.e., $\mathcal{H}[e|t] := \{y \in \mathbb{R}^m \mid \|y - y[e-1]\|_2 \leq \Delta\}$ for all $t \in (t_{e-1}, t_e)$. In addition, let us define that the sensor node not only exchanges $y[e]$ but the number e of the corresponding event as well. This number can then be used by the EBSE for detecting when previously exchanged samples were lost.

As an example, let us assume that the EBSE receives a new event sampled measurement $y[e]$, while the event received by the EBSE before t_e was $y[e-c]$, for some $c \in \mathbb{Z}_+$. As such, the $c - 1$ measurement samples prior to t_e were not received by the EBSE, though at t_e the estimator will detect that they were lost. This further implies that the estimated state at any time-synchronous sample instant $t \in (t_{e-c}, t_e)$ was updated by the EBSE with an incorrect set $\mathcal{H}[e-c|t]$, as $y(t) \notin \mathcal{H}[e-c|t]$ but $y(t) \in \mathcal{H}[e-q|t]$, for some $q \in \mathbb{Z}_{[1,c]}$. Nonetheless, since package loss was detected, the EBSE can correct this mistake. To that extent, note that the triggering sets which correspond to the lost event instants satisfy $\mathcal{H}[e-q|t] \subseteq \tilde{\mathcal{H}}[e-c|t]$, for all $q \in \mathbb{Z}_{[1,c]}$ and $\tilde{\mathcal{H}}[e-c|t] := \{y \in \mathbb{R}^m \mid \|y - y[e-c]\|_2 \leq c \cdot \Delta\}$, see also Figure 2.7. Therefore, the ‘‘incorrect’’ estimation history after t_{e-c} can be re-calculated by using the correct triggering set $\tilde{\mathcal{H}}[e-c|t]$, i.e., $y(t) \in \tilde{\mathcal{H}}[e-c|t]$ holds for all $t \in (t_{e-c}, t_e)$. The importance of this correction is that stable estimation results are then attained, since the EBSE still performs a state update with a bounded set $\tilde{\mathcal{H}}[e-c|t]$.

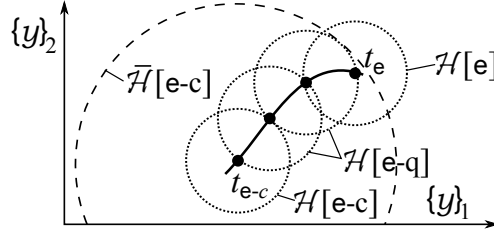


Figure 2.7: A correction of the event triggering sets $\mathcal{H}[e-q]$, where $q \in \mathbb{Z}_{[1,2]}$, in case there are two event instants missed in between t_{e-c} and t_e due to package loss, i.e., $c = 3$.

The next step in this discussion is to assess the above event sampling strategy with the EBSE. To that extent, let us consider the object tracking example of the previous section, for which the vehicle's position, speed and acceleration are presented in Figure 2.8.

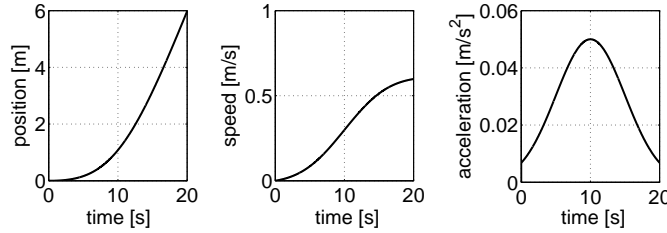


Figure 2.8: The position, speed and acceleration of the tracked object in presence of package loss.

The simulation compares the estimation results of the EBSE with the time-synchronous QKF introduced in Section 2.5. Both adopt a sampling time of $\tau_s = 0.1$ seconds. The estimators receive new measurement samples from a sensor node with a 50% probability of package loss. To that extent, the EBSE employs the above event sampling strategy, with $\Delta = 0.1$ meter, along with the procedure to correct its estimation history when it detects that packages were lost. Note that the QKF cannot perform a correction and will thus predict its estimated state when a package is lost. Figure 2.9(a) depicts the squared estimation error of both the EBSE and QKF, while the trace of their corresponding error-covariance is shown in Figure 2.9(b). Before analyzing the figures let us first point out that, since the vehicle drives 6[m] in 20 seconds, the EBSE receives 60 measurement samples and the QKF receives 200 quantized measurements. Further, Figure 2.9(a) indicates that there is a bias on the QKF estimation error. This bias is mainly a result from a bad estimate of the vehicle's speed rather than a result of lost packages.

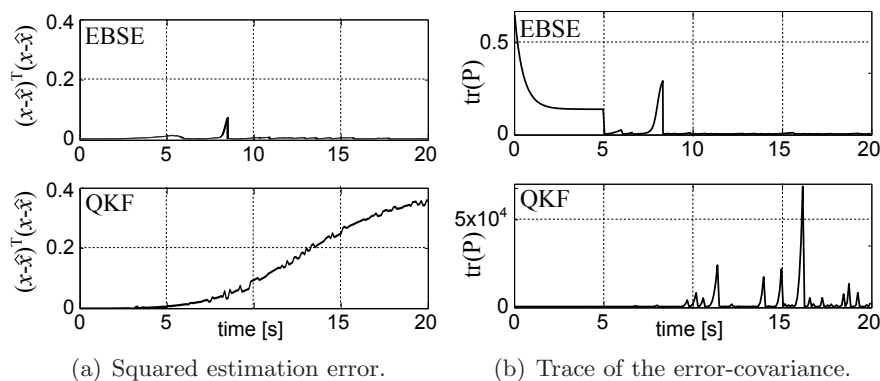


Figure 2.9: Results of the EBSE and QKF with 50% package loss.

Based on the above figures, note that the estimation error of both the EBSE as well as the QKF is bounded and that their main difference is illustrated in Figure 2.9(b) of the error-covariance. This figure shows that the error-covariance of the QKF is unbounded, which was expected from the asymptotic analysis of the Kalman filter with package loss presented in (Mo and Sinopoli, 2008). The error-covariance matrix of the EBSE on the other hand, remains bounded even in situations with severe package loss (around 8[s] for the EBSE). As such, the EBSE establishes stable estimation results, when the above proposed event sampling strategy is employed.

2.7 Conclusions

This chapter presented an event based state-estimator (EBSE) suitable for any event sampling strategy. The distinguishing feature of the proposed EBSE is that *updated* estimation results are computed at the two different types of sample instants, i.e., at event instants, when a measurement value is received, and at synchronous time instants, when no measurement is received. In the latter, case the update is based on inherent knowledge that the measured value lies within a bounded set used to define the event. Furthermore, a proof of an asymptotic bound on the largest (positive) eigenvalue of the error-covariance was presented. Hence the EBSE is a stable estimator, even in the situation that no new measurement is received anymore. This property of the EBSE was demonstrated in an illustrative example, together with an indication of its computational tractability and a discussion for coping with missed measurement samples due to package loss. Flexibility of the EBSE to utilize measurement samples from any type of event sample strategy was attained via the proposed mathematical formulation of event sampling. This analysis is continued in the next chapter, which addresses the applicability of the EBSE for control.

Event based state estimation in a control loop

3.1 A feedback control set-up 3.2 Matched sampling 3.3 Event based estimation with control actions	3.4 Integration with a robust MPC 3.5 Illustrative control example 3.6 Conclusions
--	--

The event based state-estimator (EBSE) obtained in the previous chapter is a valuable step for exploiting the information of event sampled measurements. Nonetheless, in realistic applications a state-estimator is often performed prior to a control algorithm. Therefore, this chapter studies an event based control set-up, in which the stable estimation results of the EBSE are used as input to a time-synchronous control algorithm. Note that the EBSE supports any event sampling strategy. Therefore, a similar property is expected for the proposed event based controller, which further implies that stability of the feedback control system is decoupled from the event triggering criteria. Besides analyzing this expectation, the chapter presents a novel event sampling strategy to obtain a more constant control performance. On top of that, a stabilizing control algorithm is proposed by modifying an existing robust MPC approach, such that the disturbance rejection is optimized depending on estimation errors. A justification of the design is given, along with illustrative control examples for various event sampling strategies.

3.1 A feedback control set-up

Event based control has emerged recently as a viable alternative to classical, time-synchronous control, with many relevant applications in networked systems. For networked systems that have a control objective, event based control offers a straightforward solution to cope with the limitations in communication. This is because event based control aims to reduce the exchange of measurement data by employing an event sampling strategy. A recent overview of pros and cons of this control approach is found in (Åström, 2008). The set-up for event based control, which is described next, is similar the that of the event based state-estimator in Chapter 2.

Communication The sensor node employs an event sampling strategy to obtain a measurement $y(t) \in \mathbb{R}^m$ at the instants of an event. The e -th event instant is denoted as $t_e \in \mathbb{R}_+$ and $\mathbb{T}_e : \cup_{e \in \mathbb{Z}_+} \{t_e\}$ represents the set of all event instants. Then, for a given set $\mathcal{H}[e] \subset \mathbb{R}^{m+1}$ in *time-measurement*-space, the next event instant is characterized by $t_e := \inf \{t \in \mathbb{R}_+ \mid t > t_{e-1} \text{ and } \binom{y(t)}{t} \notin \mathcal{H}[e]\}$. See (2.2) for more details.

Process Let us consider the controlled, linear process of (1.4), i.e.,

$$x(t) = A(\tau)x(t-\tau) + B(\tau)u(t-\tau) + w(t, \tau), \quad (3.1a)$$

$$y(t) = Cx(t) + Du(t) + v(t), \quad (3.1b)$$

in which $x \in \mathbb{R}^n$ denotes the state vector. Both the process-noise $w \in \mathbb{R}^n$ and measurement-noise $v \in \mathbb{R}^m$ are characterized by Gaussian PDFs, for some $Q(\tau) \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{m \times m}$, i.e.,

$$p(w(t, \tau)) := G(w(t, \tau), 0, Q(\tau)) \quad \text{and} \quad p(v(t)) := G(v(t), 0, V).$$

The goal is to control the process modeled in (3.1) from the event sampled measurements $y(t_e)$. Subsequent studies on control that are based on the event sampling method ‘‘Send-on-Delta’’ were presented in (Brockett and Liberzon, 2000; Kofman and Braslavsky, 2006; Heemels et al., 2008; Henningsson et al., 2008; Dimarogonas and Johansson, 2009). The conclusion that can be drawn from these works is that, when measurements are sent only at event instants, it is difficult to guarantee (practical) stability of the control system. The natural solution that emerged for solving this problem was to include the controller in the event triggering process. Various alternatives are presented in (Brockett and Liberzon, 2000; Tabuada, 2007; Wang and Lemmon, 2009; Lehmann and Lunze, 2009; Dimarogonas and Johansson, 2009) and the references therein. The general procedure within this framework is to define a specific criterion for triggering events as a function of the state vector. This function can either be related to guarantee control stability, see, e.g., (Wang and Lemmon, 2009), or to improve disturbance rejection, see, e.g., (Lehmann and Lunze, 2009). One of the concerns regarding existing event based controllers is the fact that they are designed for a specific type of event sampling method or, that the sampling method is designed specifically for the controller. This implies that both functionalities of the system, i.e., event sampling and control, depend heavily on each other and changing one requires a re-design of the other to guarantee the same properties for the control system. Furthermore, except for the recent controller presented in (Donkers, 2012), existing solution on event based control assume that the entire state is measured, i.e., $y(t) = x(t) + v(t)$.

To obtain new insights for addressing the above concerns of existing event based control methods, this chapter presents a feasibility study of a control system that includes the event based state-estimator (EBSE) of Chapter 2. A schematic set-up of such an event based controller is depicted in Figure 3.1, for which the following control advantages are expected:

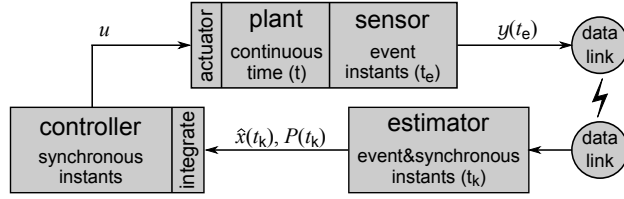


Figure 3.1: Schematic set-up for integrating the EBSE in a feedback loop.

- The control set-up supports any event sampling strategy;
- Control stability is decoupled from the event triggering criteria, as the EBSE already yields a bounded error-covariance;
- The EBSE enables an integration of event sampled measurements with (the widely available) time-synchronous controllers.

Let us give a detailed description of Figure 3.1, before continuing with the outline of this chapter. The sensor samples new measurements at the instant of an event that are collected in the set $\mathbb{T}_e = \cup_{e \in \mathbb{Z}_+} \{t_e\}$, while the controller runs synchronously in time, for which the set of synchronous sample instants $\mathbb{T}_s := \{s\tau_s \mid s \in \mathbb{Z}_+\}$ is introduced with $\tau_s \in \mathbb{R}_{>0}$ as the sampling time. The EBSE computes an estimate of the state $x(t_k)$ at both event and synchronous instants $t_k \in \mathbb{T}$, where $\mathbb{T} := \mathbb{T}_e \cup \mathbb{T}_s$. This estimate has a Gaussian distribution and is thus characterized by some mean $\hat{x}(t_k) \in \mathbb{R}^n$ and error-covariance $P(t_k) \in \mathbb{R}^{n \times n}$. The two topics that complete the proposed event based control set-up are the employed event sampling strategy and the integration of the EBSE with a suitable control algorithm.

Let us start with an event sampling strategy. The illustrative case-study presented in Section 2.5 indicates that the estimation error of the EBSE can change rapidly when Send-on-Delta is employed for obtaining event sampled measurements. Similar variations on the estimation results are likely to appear with other existing event sampling strategies. Therefore, a novel event sampling method is proposed in the next section, which reduces these rapid variations by sending relevant measurements that are matched to the EBSE.

The remaining sections address the integration of estimation and control. Firstly, the EBSE is extended, so that control actions can be included. Secondly, the employed control algorithm is presented that involves a modification of the robust MPC of (Lazar and Heemels, 2008) into a controller that can deal with estimation errors. Thirdly, an integration procedure is developed, in which the error-covariance of the EBSE is transformed into a deterministic measure of the estimation error. This measure can then be used by the modified robust MPC for computing a stabilizing control action and moreover, for optimizing disturbance rejections of the feedback loop.

3.2 Matched sampling

This section presents the event sampling method *Matched sampling*. The goal of Matched sampling is to reduce variations on the estimation performance by matching the relevance of $y(t)$ to the estimation results. To that end, a formal characterization of relevant measurements is proposed, next, followed by the corresponding event triggering criterion. This is then used to analyze stability of the EBSE with Matched sampling by studying the resulting set $\mathcal{H}[e]$. Furthermore, a comparison with Send-on-Delta demonstrates that the EBSE yields a more constant estimation result in favor of Matched sampling.

3.2.1 Event triggering criteria

Matched sampling triggers a new event instant $t_e = t$ if $y(t)$ is “relevant” for the estimator. To find a suitable characterization of a “relevant” $y(t)$, let us define $y(t_{0:e-1}) := \{y(t_q) | t_q \in \mathbb{T}_e, t_q < t\}$ as all $e-1$ event sampled measurements until time t . Further, let $p(x(t_{e-1}) | y(t_{0:e-1}))$ denote the EBSE result at t_{e-1} . Then, the sensor node determines the following PDFs:

- $p(x(t) | y(t_{0:e-1}), y(t))$ represents the *updated* PDF of x at a current time $t > t_{e-1}$ in case $y(t)$ is shared with the state-estimator.
- $p(x(t) | y(t_{0:e-1}))$ represents the *predicted* PDF of x at a current time $t > t_{e-1}$ in case $y(t)$ is *not* shared with the state-estimator.

In information theory, the relevance of an updated PDF compared to a predicted PDF is often determined via the *Kullback-Leibler divergence*.

The Kullback-Leibler divergence $d(p_1(x) || p_2(x)) \in \mathbb{R}_+$ of two PDFs $p_1(x)$ and $p_2(x)$ is a *non-symmetric* measure for the difference of $p_2(x)$ relative to $p_1(x)$. Therein, $p_1(x)$ is considered to be the *true* or *updated* PDF of a random vector $x \in \mathbb{R}^n$, while $p_2(x)$ is a *model* or *prediction* of $p_1(x)$. The divergence is also known as the uncertainty reduction on x that is achieved if $p_1(x)$ is replaced by $p_2(x)$. The definition of the Kullback-Leibler divergence, as it is found in (Cover and Thomas, 1991), yields

$$d(p_1(x) || p_2(x)) := \int_{\mathbb{R}^n} p_1(x) \log \frac{p_1(x)}{p_2(x)} dx. \quad (3.2)$$

An illustration of $p_1(x) \log \frac{p_1(x)}{p_2(x)}$, for three examples of $p_2(x)$, is depicted in Figure 3.2. Although the Kullback-Leibler divergence is often intuited as a metric or distance, it is not a true metric, for example, as it is not symmetric. Also, it is not the same as a divergence in calculus. For that reason the Kullback-Leibler divergence was originally introduced in (Kullback and Leibler, 1951) as a *directed* divergence from one distribution to another.

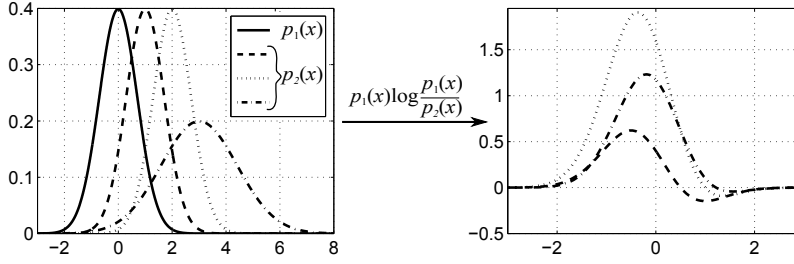


Figure 3.2: The function $p_1(x) \log \frac{p_1(x)}{p_2(x)}$ that corresponds to the Kullback-Leibler divergence $d(p_1(x)||p_2(x))$, for three different $p_2(x)$ and $x \in \mathbb{R}$. The value of $d(p_1(x)||p_2(x))$ is equivalent to the integral of this function.

Note that, in line with the Kullback-Leibler divergence, the sensor node computes both the updated PDF $p(x(t)|y(t_{0:e-1}), y(t))$ and the predicted PDF $p(x(t)|y(t_{0:e-1}))$. As such, the corresponding Kullback-Leibler divergence can be regarded as a measure for the relevance of $y(t)$, i.e., $y(t)$ is more relevant for the estimator when the value of this divergence increases. Therefore, along the results of (Marck et al., 2008) for triggering new measurements in event based classification, the event triggering criteria of Matched sampling is a level-crossing criterion of the Kullback-Leibler divergence, for some $\Delta_{KL} \in \mathbb{R}_+$, i.e.,

$$t_e := \inf \{t \in \mathbb{R}_+ \mid t > t_{e-1} \text{ and } d(p_1(x(t))||p_2(x(t))) > \Delta_{KL}\}, \quad (3.3a)$$

$$\text{where } p_1(x(t)) := p(x(t)|y(t_{0:e-1}), y(t)),$$

$$p_2(x(t)) := p(x(t)|y(t_{0:e-1})). \quad (3.3b)$$

A characterization of $p_1(x(t))$ and $p_2(x(t))$ in (3.3b) is derived by assuming that the sensor node has access to the estimation results of the EBSE at t_{e-1} , i.e., $p(x(t_{e-1})|y(t_{0:e-1})) = G(x(t_{e-1}), \hat{x}(t_{e-1}), P(t_{e-1}))$. This PDF is then used by the sensor node to compute $p_1(x(t))$ and $p_2(x(t))$ according to the standard asynchronous Kalman filter of (Mallick et al., 2001), for some $\tau_e := t - t_{e-1}$, so that the computational power required from a sensor node is limited, i.e.,

$$p_1(x(t)) := G(x(t), \hat{x}_1(t), P(t)), \quad p_2(x(t)) := G(x(t), \hat{x}_2(t), P_2(t)), \quad (3.4a)$$

$$\text{where } P_2(t) := A(\tau_e)P(t_{e-1})A^\top(\tau_e) + Q(\tau_e),$$

$$\hat{x}_2(t) := A(\tau_e)\hat{x}(t_{e-1}),$$

$$P_1(t) := (P_2^{-1}(t) + C^\top V^{-1}C)^{-1}, \quad (3.4b)$$

$$\hat{x}_1(t) := P_1(t)(P_2^{-1}(t)\hat{x}_2(t) + C^\top V^{-1}y(t)).$$

The above expressions are a direct result from the fact that $p_2(x(t))$ is a prediction of $p(x(t_{e-1})|y(t_{0:e-1}))$ towards $t > t_{e-1}$ based on the process model (3.1), while $p_1(x(t))$ yields an update of $p(x(t_{e-1})|y(t_{0:e-1}))$ with the current measurement $y(t)$. The results of (Majda et al., 2002) on the Kullback-Leibler

divergence of Gaussian PDFs further gives that $d(p_1(x)||p_2(x))$ of (3.2) can be separated in a *dispersion*-term, denoted with $\alpha \in \mathbb{R}$, and *signal*-term, i.e.,

$$\begin{aligned} d(p_1(x)||p_2(x)) &= \alpha(t) + \frac{1}{2}(\hat{x}_1(t) - \hat{x}_2(t))^\top P_2^{-1}(t)(\hat{x}_1(t) - \hat{x}_2(t)), \\ \alpha(t) &:= \frac{1}{2} \left(\log |P_2(t)| |P_1(t)|^{-1} + \text{tr} (P_2^{-1}(t)P_1(t)) - n \right). \end{aligned} \quad (3.5)$$

The variable n in (3.5) is the vector size of x . Note that $\alpha(t)$ depends on the difference of the updated error-covariance $P_1(t)$ with the predicted $P_2(t)$. Hence, the bigger this difference, the larger $\alpha(t)$ will be and the sooner $d(p_1(x)||p_2(x)) > \Delta_{KL}$ will trigger a new event. A similar reasoning holds for the signal-term depending on the prediction error, i.e., larger values of $(\hat{x}_1(t) - \hat{x}_2(t))^\top (\hat{x}_1(t) - \hat{x}_2(t))$ will trigger new events. This completes the event triggering condition of Matched sampling. Next, let us assess the proposed sampling strategy in combination with the EBSE of Chapter 2.

3.2.2 EBSE with Matched sampling

The EBSE performs an update at each sample instant $t \in \mathbb{T}$ (for brevity, this paragraph will use t to denote the EBSE instants instead of t_k). At synchronous instants, this update is based on a *bounded* set $\mathcal{H}[e|t]$, such that $y(t) \in \mathcal{H}[e|t]$, see Proposition 2.4.1. Otherwise, stability¹ of the EBSE cannot be guaranteed. Therefore, the analysis of Matched sampling in combination with the EBSE focuses on a proof that $\mathcal{H}[e|t]$ is a bounded set. This proof is given after deriving an expression of $\mathcal{H}[e|t]$ for Matched sampling.

A derivation of the set $\mathcal{H}[e|t]$ starts from $d(p_1(x)||p_2(x))$ of (3.5), as this divergence characterizes the event triggering criterion of Matched sampling. The derivation makes use of the following result, which is obtained from the update equations of $\hat{x}_1(t)$ in (3.4b), i.e.,

$$\begin{aligned} \hat{x}_1(t) - \hat{x}_2(t) &= P_1(t)(P_2^{-1}(t)\hat{x}_2(t) + C^\top V^{-1}y(t)) - \hat{x}_2(t) \\ &= P_1(t) \left(C^\top V^{-1}y(t) - (P_1^{-1}(t) - P_2^{-1}(t))\hat{x}_2(t) \right) \\ &= P_1(t)C^\top V^{-1}(y(t) - C\hat{x}_2(t)). \end{aligned}$$

Substituting this result into the Kullback-Leibler divergence of (3.5), yields

$$\begin{aligned} d(p_1(x)||p_2(x)) &= \alpha(t) + \frac{1}{2}(y(t) - C\hat{x}_2(t))^\top \Upsilon(t)(y(t) - C\hat{x}_2(t)), \\ \text{where } \Upsilon(t) &:= V^{-1}CP_1(t)P_2^{-1}(t)P_1(t)C^\top V^{-1}. \end{aligned} \quad (3.6)$$

In the situation that no new measurement was received at time t , it must hold that $d(p_1(x)||p_2(x)) \leq \Delta_{KL}$, see (3.3), due to which

$$(y(t) - C\hat{x}_2(t))^\top \Upsilon(t)(y(t) - C\hat{x}_2(t)) \leq 2(\Delta_{KL} - \alpha(t)). \quad (3.7)$$

¹Stability of the EBSE means that $\lambda_{\max}(P(t))$ is asymptotically bounded.

The inequality of (3.7) is similar to the characterization of an ellipsoidal sub-level-set, i.e., $\mathcal{E}_{\mu,\Sigma} := \{y \in \mathbb{R}^m \mid (y - \mu)^\top \Sigma^{-1} (y - \mu) \leq 1\}$, for some $\mu \in \mathbb{R}^m$ and $\Sigma \in \mathbb{R}^{m \times m}$ (see Section 1.5). Hence, even though $y(t)$ is not available to the EBSE, the inequality in (3.7) guarantees that $y(t) \in \mathcal{H}[e|t]$, when

$$\mathcal{H}[e|t] := \mathcal{E}_{C\hat{x}_2(t), \Phi^{-1}(t)} \quad \text{and} \quad \Phi(t) := \frac{1}{2}(\Delta_{KL} - \alpha(t))^{-1} \Upsilon(t). \quad (3.8)$$

Note that the EBSE can obtain $\mathcal{H}[e|t]$ of (3.8) via the values of $\hat{x}(t_{e-1})$, $P(t_{e-1})$, C and V . Only the threshold Δ_{KL} must be shared by the sensor node.

Based on the availability of this threshold, let us analyze the stability of the EBSE in combination with Matched sampling, which is guaranteed when $\mathcal{H}[e|t]$ is bounded for all $t \in \mathbb{T}$. The results of the next two lemmas prove this property for Matched sampling, by observing that $\mathcal{H}[e|t]$ of (3.8) is bounded for all $t \in \mathbb{T}$, if $\Phi \succ 0$ holds for all $t \in \mathbb{T}$. Or similarly,

- $0 < \Delta_{KL} - \alpha(t) < \infty$ holds for all $t \in \mathbb{T}$ and
- $\Upsilon(t) \succ 0$ holds for all $q \in \mathbb{Z}_{[1,n]}$ and all $t \in \mathbb{T}$.

Lemma 3.2.1 *Let $\Delta_{KL} \in \mathbb{R}_+$ be given and let $\alpha(t)$ satisfy (3.5). Then, $0 < \Delta_{KL} - \alpha(t) \leq \Delta_{KL} + \frac{1}{2}n$ holds for all $t \in \mathbb{T}$.*

Lemma 3.2.2 *Let $C \in \mathbb{R}^{m \times n}$ of (3.1) be such that $\text{rank}(C) = m$ and let $\Upsilon(t)$ satisfy (3.6). Then, $\Upsilon(t) \succ 0$ holds for all $t \in \mathbb{T}$.*

The proofs of Lemma 3.2.1 and Lemma 3.2.2 are found in Appendix C.1 and Appendix C.2, respectively. Other conditions for attaining bounded eigenvalues of the error-covariance $P(t)$ are not addressed here, since they do not depend on the event sampling strategy. As such, the EBSE in combination with Matched sampling satisfies the conditions for stable estimation results from a sampling point of view. This aspect is demonstrated in the next illustrative example, after the following remark on the premise of Lemma 3.2.2.

Remark 3.2.3 Lemma 3.2.2 requires that $\text{rank}(C) = m$ is met by the measurement matrix $C \in \mathbb{R}^{m \times n}$. Not satisfying this condition means that multiple sensors are measuring the same q -th state-element $\{x\}_q$. To circumvent the issue that $\text{rank}(C) \neq m$, one can first fuse the independent measurements via standard sensor fusion methods, see, e.g., (Bar-Shalom and Campo, 1986). The resulting ‘‘measurement matrix’’ $C_f \in \mathbb{R}^{m_f \times n}$ that corresponds to the fused measurement does satisfy $\text{rank}(C_f) = m_f$.

3.2.3 An illustrative comparison

The effectiveness of Matched sampling (MS) is compared to Send-on-Delta (SoD) in the object tracking case-study of Section 2.5. Two simulations

are analyzed, in which the sensor node either performs MS or SoD. The triggering condition of MS is characterized by (3.3) and $\Delta_{KL} = 1.5$, while SoD of Section 1.3.1 employs $\Delta = 0.5$. Details on the considered process model are found in Section 2.5.

The EBSE performs an update on the estimation results at each sample instant $t_k \in \mathbb{T}$. At the instants of an event $t_k \in \mathbb{T}_e$, a new measurement is received by the EBSE, while at the synchronous sample instants $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$ no measurement is received and the EBSE performs an update based on the fact that $y(t_k) \in \mathcal{H}[e|t_k]$. To that extent, the EBSE determines an *implied* measurement value $\hat{y}^1(t_k) \in \mathbb{R}^m$ with a certain covariance $R(t_k) \in \mathbb{R}^{m \times m}$, see also the EBSE algorithm presented in Section 2.3.3. Suitable values for both sampling strategies are the following:

$$\begin{aligned} \text{MS: } \hat{y}^1(t_k) &= C\hat{x}_2(t_k), & R &= \frac{1}{2}(\Delta_{KL} - \alpha(t_k))\Upsilon(t_k) + V; \\ \text{SoD: } \hat{y}^1(t_k) &= y(t_{e-1}), & R &= \frac{1}{4}\Delta^2 + V. \end{aligned}$$

Figure 3.3 illustrates the event instants of the two different event sampling strategies in the plot of the estimated position computed by the EBSE. Notice that SoD triggered 54 event instants, which for MS were reduced to 38 instants. Among others, these numbers depend on process dynamics.

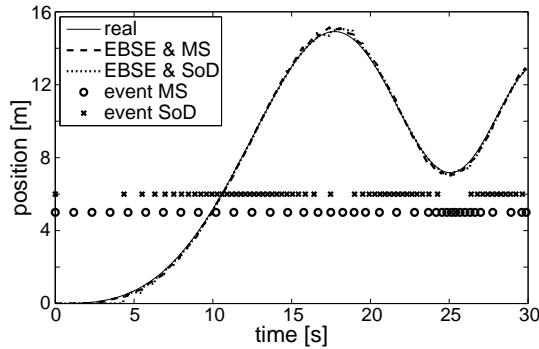


Figure 3.3: The event sample instants and estimated positions of EBSE in combination with Matched sampling (MS) and with Send-on-Delta (SoD).

The above figure further indicates that the main difference between MS and SoD is *when* an event is triggered. For SoD most sample instants takes place when the object has a high but constant speed. For MS most samples takes place when the unknown acceleration is large, which is considered to be of more relevance to the EBSE. Similar results are also noticed in Figure 3.4, which depicts the modeled and true estimation error of the EBSE for both MS and SoD, i.e. $\text{tr}(P)$ and $(x - \hat{x})^\top(x - \hat{x})$, respectively.

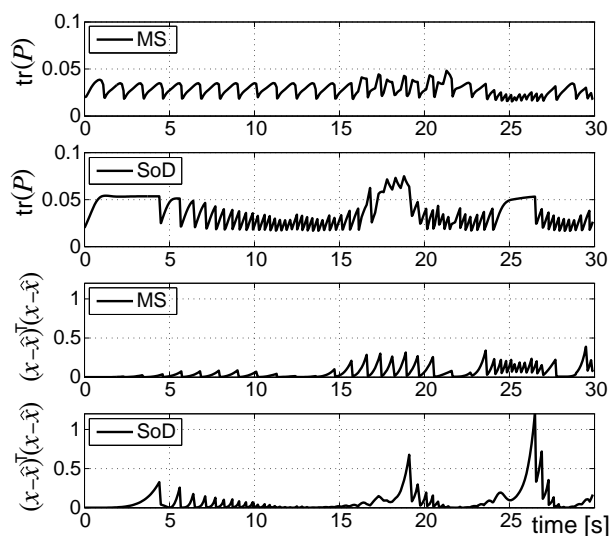


Figure 3.4: The estimation results of the Matched sampling (MS) versus Send-on-Delta (SoD) on the modeled estimation error, i.e., $\text{tr}(P)$, and the real estimation error, i.e., $(x - \hat{x})^T(x - \hat{x})$.

A first conclusion that can be drawn from Figure 3.4 is that, *on average*, the estimation results are comparable for both sampling strategies. However, the modeled and true estimation error with MS are more alike compared to SoD. Also, the variations of $\text{tr}(P)$ and $(x - \hat{x})^T(x - \hat{x})$ differ. The results with MS indicate that their values almost have a repetitive behavior, i.e., with little variations and a predictive character. For the EBSE combined with SoD, the amplitude variations are more severe and shows a rapid impulsive behavior mixed with steady periods. Hence, when MS is employed as event sampling strategy, the EBSE shows less variation in its performance measures, i.e., the estimation error and error-covariance. This without comprising on communication requirements by inducing more event instants.

The next step is to close the loop with a controller, for which the EBSE of Chapter 2 is extended to include control actions.

3.3 Event based estimation with control actions

The event sampled measurements are exploited by the EBSE to compute an estimate of x . However, the current control value $u(t_k)$ is additional information that can be used by the EBSE as well. Since including $u(t_k)$ involves a slight modification of the EBSE algorithm presented in Section 2.3.3, let

us briefly address this extension. For clarity of expression, the notation of Chapter 2 is adopted, i.e., $x[k]$ is used to denote $x(t_k)$ and so on.

The extended EBSE algorithm is summarized below. Therein, the event sampling strategy is used to formulate *implied* measurement information that is represented by a “sensor value” $\hat{y}_q[k] \in \mathbb{R}^m$, for all $q \in \mathbb{Z}_{[1,N]}$, and a corresponding “noise covariance” $R[k] \in \mathbb{R}^{m \times m}$ via the Likelihood(\cdot, \cdot, \cdot) routine. Also, note that $\omega_q[k] \in \mathbb{R}_+$ is by definition a scalar weight.

Algorithm of the EBSE

Prediction

$$\hat{x}[k^-] = A(\tau_k)\hat{x}[k-1] + B(\tau_k)u[k-1];$$

$$P[k^-] = A(\tau_k)P[k-1]A^\top(\tau_k) + Q(\tau_k);$$

Measurement update

$$(\hat{y}^1[k], \dots, \hat{y}^N[k], R[k]) = \text{Likelihood}(\mathcal{H}[e|t_k], V), \quad (2.14);$$

$$\Theta[k] = (P^{-1}[k^-] + C^\top R^{-1}[k]C)^{-1};$$

for all $q \in \mathbb{Z}_{[1,N]}$, do:

$$\hat{\theta}_q[k] = \Theta[k](P^{-1}[k^-]\hat{x}[k^-] + C^\top R^{-1}[k](\hat{y}_q[k] - Du[k]));$$

$$\omega_q[k] = G(\hat{y}_q[k] - Du[k], C\hat{x}[k^-], CP[k^-]C^\top + R[k]);$$

end

Approximation

$$\hat{x}[k] = \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} \hat{\theta}_q[k];$$

$$P[k] = \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} (\Theta[k] + (\hat{x}[k] - \hat{\theta}_q[k])(\hat{x}[k] - \hat{\theta}_q[k])^\top).$$

The above EBSE algorithm is similar to the one of Chapter 2 added with the terms $B(\tau_k)u[k-1]$ and $Du[k]$. Notice that these additional terms are present in both $\hat{\theta}_q[k]$ and $\hat{x}[k]$. As a result, the effect of the control action $u[k]$ cancels out in $\hat{x}[k] - \hat{\theta}_q[k]$ and thus in the update of $P[k]$ as well. This further implies that stability of the EBSE with control actions is obtained *mutatis mutandis* from the proof of Theorem 2.4.6, for which $\alpha(\tau_s) := \sup_{\tau \in [0, \tau_s]} \sigma_{\max}(A(\tau))$ and $\beta(\tau_s) := \sup_{\tau \in [0, \tau_s]} \sqrt{\lambda_{\max}(Q(\tau))}$.

Theorem 3.3.1 *Let Assumption 2.4.1 and the premise of Proposition 2.4.5 be satisfied. Then the EBSE with control actions results in a stable estimate, i.e., $\lambda_{\max}(P[\infty]) \leq \alpha^2(\tau_s)\lambda_{\max}(\Sigma[\infty]) + \beta^2(\tau_s)$ holds and $\Sigma[\infty]$ exists.*

The result of Theorem 3.3.1 guarantees that $P[k]$ is asymptotically bounded.

This property is used by the robust MPC to calculate a stabilizing control action while tuning the disturbance rejection.

3.4 Integration with a robust MPC

The time-synchronous controller of the proposed event based control set-up of Figure 3.1 will be based on the robust MPC approach presented in (Lazar and Heemels, 2008). Originally, the approach was introduced to achieve input-to-state stability (ISS) with respect to additive disturbances, see Definition 1.5.5. In this section, the robust MPC is modified, so that its ISS-property holds for estimation errors, instead. Moreover, the MPC algorithm offers a possibility to enhance disturbance rejection in case the estimation error is small. This is one of the main reasons to select this controller, as it can act during operation on the time-varying estimation results of the EBSE. To that end, an integration procedure is developed that exploits the error-covariance $P(t_k)$ of the EBSE and turns it into a deterministic measure of the estimation error. Further, an illustrative control example shows convincing results of the proposed feedback set-up.

3.4.1 A robust MPC algorithm

The controller receives the estimates $\hat{x}(t_k)$ and $P(t_k)$ at each synchronous instant $t_k \in \mathbb{T}_s$, according to the specified sampling time τ_s . These variables are used by the controller to determine a stabilizing control action $u(t_k)$ depending on the estimated state $\hat{x}(t_k)$ and estimation error $\varpi(t_k) := \hat{x}(t_k) - x(t_k)$. While estimators tend to track the *current* state $x(t_k)$, controllers aim to optimize the *next* state $x(t_k + \tau_s)$. As such, a typical representation of the model in (3.1) for control purposes, yields

$$x(t_k + \tau_s) := Ax(t_k) + Bu(t_k) + \varpi(t_k), \quad \forall t_k \in \mathbb{T}_s, \quad (3.9)$$

where $A := A(\tau_s)$ and $B := B(\tau_s)$. As the robust MPC of (Lazar and Heemels, 2008) is a deterministic controller, x , u and ϖ are characterized by a bounded region rather than a PDF. Hence, let us assume that $x(t_k) \in \mathbb{X} \subseteq \mathbb{R}^n$, $\hat{x}(t_k) \in \mathbb{X}$, $u(t_k) \in \mathbb{U} \subseteq \mathbb{R}^l$ and $\varpi(t_k) \in \mathbb{W}(t_k) \subset \mathbb{R}^n$, such that $0 \in \text{int}(\mathbb{X})$, $0 \in \text{int}(\mathbb{U})$ and $0 \in \text{int}(\mathbb{W}(t_k))$, for all $t_k \in \mathbb{T}_s$. Furthermore, $\mathbb{W}(t_k)$ is a symmetric set, i.e., $\varpi \in \mathbb{W}$ implies that $-\varpi \in \mathbb{W}$.

The time-dependent set $\mathbb{W}(t_k)$ is characterized by a known polytope (closed and bounded polyhedron) that defines the deterministic bounds of the estimation error $\varpi(t_k)$. An efficient procedure to determine $\mathbb{W}(t_k)$ from $P(t_k)$ is the topic of Section 3.4.2. Let us continue here by presenting the modified robust MPC. For simplicity and clarity of exposition, the instant t_k is omitted throughout a part of this section, i.e. \hat{x} , x , \mathbb{W} , etc. will denote $\hat{x}(t_k)$, $x(t_k)$, $\mathbb{W}(t_k)$ and so on. The goal is to design a control algorithm that finds

a control action $u(\hat{x}) \in \mathbb{U}$, such that for all $\varpi \in \mathbb{W}$ the following holds

$$V(Ax+Bu(\hat{x})) - V(x) + a\|x\|^c - \vartheta(\|\varpi\|) \leq 0. \quad (3.10)$$

Let us explain each element of the above inequality.

- $\vartheta : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a continuous, strictly increasing function and $\vartheta(0) = 0$.
- $a\|x\|^c$, for some $a, c \in \mathbb{R}_{>0}$, is a power function of $\|x\|$, i.e., a continuous, strictly increasing function on $\|x\| \in \mathbb{R}_+$ that is 0 if $\|x\| = 0$.
- $V(\cdot)$, in this entire section, is not the measurement-noise covariance. To respect the notation in control theory $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$ denotes a Lyapunov function, satisfying $a_1\|x\|^c \leq V(x) \leq a_2\|x\|^c$ for some $a_1, a_2 \in \mathbb{R}_{>0}$.

It was already proven in (Kellett and Teel, 2005) that satisfying (3.10), for some Lyapunov function $V(\cdot)$, would guarantee ISS of the corresponding closed-loop system (3.9). The ISS-property is derived by substituting $\beta(\|x[0]\|, k) = a_1^{-1}(2(1 - a_2^{-1}a)^k a_2)^{-c}\|x[0]\|$ into Definition 1.5.5 of ISS together with the following *ISS-gain*:

$$\gamma(\|\varpi\|) = (2a)^c a_1^{-1} a_2^{-c} (\vartheta(\|\varpi\|))^{-c}. \quad (3.11)$$

The robust MPC proposed in (Lazar and Heemels, 2008) optimizes this ISS-gain at each sample instant by *minimization* of $\vartheta(\|\varpi\|)$ and thereby, improves the disturbance rejection of the feedback loop. In what follows, the results of (Lazar and Heemels, 2008) are extended to estimation errors that act as a disturbance on the input of the controller.

Finite dimensional problem

The first step of the robust MPC approach is to substitute the inequality of (3.10), which should hold for an infinite number of values $\varpi \in \mathbb{W}$, by an inequality for a finite number of specific values that characterize \mathbb{W} .

To that extent, let us denote ϖ_j , for all $j \in \mathbb{Z}_{[1,N]}$, as the vertices of $\mathbb{W} \subset \mathbb{R}^n$, each having some optimization variable $\varsigma_j \in \mathbb{R}_+$. Further, let us define the matrix $W \in \mathbb{R}^{n \times N}$ and the row-vector $\zeta \in \mathbb{R}^N$ as follows:

$$W := (\varpi_1 \quad \varpi_2 \quad \cdots \quad \varpi_N) \quad \text{and} \quad \zeta := (\varsigma_1 \quad \varsigma_2 \quad \cdots \quad \varsigma_N). \quad (3.12)$$

The set $\mathbb{W} \subset \mathbb{R}^n$ is divided into M unique subsets $\mathcal{S}_i \subset \mathbb{R}^n$ called simplices, for all $i \in \mathbb{Z}_{[1,M]}$, i.e., $\cup_{i \in \mathbb{Z}_{[1,M]}} \mathcal{S}_i = \mathbb{W}$ and $\text{int}(\mathcal{S}_i) \cap \text{int}(\mathcal{S}_r) = \emptyset$ for $i \neq r$. See also Figure 3.5 for a graphical illustration. Notice that each simplex \mathcal{S}_i is equal to the convex hull of l particular vertices ϖ_j and the origin, i.e., $\mathcal{S}_i := \text{Co}\{0, \varpi_{q(i,1)}, \dots, \varpi_{q(i,l)}\}$, for some $\varpi_{q(i,r)} \in \{\varpi_1, \dots, \varpi_N\}$ and $r \in \mathbb{Z}_{[1,l]}$. Further, let us introduce a unique ‘‘vertices selection’’ matrix $T_i \in \mathbb{R}^{N \times n}$ with a corresponding simplex $\mathcal{S}_i = \text{Co}\{0, \varpi_{q(i,1)}, \dots, \varpi_{q(i,l)}\}$ as follows:

$$T_i := \{T \in \mathbb{R}^{N \times n} \mid \mathcal{S}_i, WT = (\varpi_{q(i,1)}, \dots, \varpi_{q(i,l)})\},$$

where, $\{T\}_{qr} \in \{0, 1\}$, $\forall q \in \mathbb{Z}_{[1,N]}, \forall r \in \mathbb{Z}_{[1,n]}$.

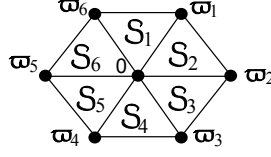


Figure 3.5: An example of the set \mathbb{W} with its corresponding vertices ϖ_j and simplices \mathcal{S}_i , for all $j \in \mathbb{Z}_{[1,N]}$ and $i \in \mathbb{Z}_{[1,M]}$.

Note that the matrix WT_i is invertible for all $i \in \mathbb{Z}_{[1,M]}$. In case \mathcal{S}_3 of Figure 3.5 is taken as an example, then $\mathcal{S}_3 = \text{Co}\{0, \varpi_{q(3,1)}, \varpi_{q(3,2)}\}$, $q(3,1) = 2$ and $q(3,2) = 3$, due to which $T_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}^\top$.

The definitions of W , ζ and T_i , for all $i \in \mathbb{Z}_{[1,M]}$, are used to derive a finite number of inequalities that can be used as a substitute for (3.10). Although this derivation assumes that x and \hat{x} are known, the dependency on x is removed afterwards. Hence, consider the following finite set of inequalities,

$$\begin{aligned} V(A\hat{x} + Bu(\hat{x})) - V(x) + a\|x\|^c &\leq 0, \\ V(A(\hat{x} - \varpi_j) + Bu(\hat{x})) - V(x) + a\|x\|^c - \varsigma_j &\leq 0, \quad \forall j \in \mathbb{Z}_{[1,N]}. \end{aligned} \quad (3.13)$$

Assumption 3.4.1 $V(\cdot)$ is a continuous, convex Lyapunov function. This includes quadratic functions, i.e., $V(x) = x^\top P_v x$ with $P_v \succ 0$, and functions based on norms, i.e., $V(x) = \|P_v x\|$ with P_v a full-column rank matrix.

Lemma 3.4.2 Let Assumption 3.4.1 be satisfied and let \hat{x} and x be given. If there exist a $u(\hat{x})$ and ϖ_j , for all $j \in \mathbb{Z}_{[1,N]}$, such that (3.13) holds, then (3.10) holds for the same $u(\hat{x})$, with $\vartheta(\|\varpi\|) := \eta\|\varpi\|$ and

$$\eta := \max_{i=1, \dots, M} \{\|\zeta T_i (WT_i)^{-1}\|\}. \quad (3.14)$$

The proof of this lemma is found in Section C.3.

To remove the dependency of (3.13) on x and thereby, formulate a finite dimensional optimization problem attaining closed-loop ISS, let us re-institute the time notion of t_k . Further, note that $x(t_k) \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$ holds and that $\hat{x}(t_k)$ and $\mathbb{W}(t_k)$ are available to the controller. Then, removing this dependency implies that $V(x)$ and $a\|x\|^c$ of (3.13) are substituted with variables that depend on $\hat{x}(t_k)$ and $\mathbb{W}(t_k)$. To that extent, let us define

$$V_{\min}(t_k) := \min_{x \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)} V(x), \quad (3.15)$$

$$a_{\max}(t_k) := \max_{x \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)} a\|x\|^c. \quad (3.16)$$

Next, let us define a cost-function $J: \mathbb{R}^N \rightarrow \mathbb{R}_+$, depending on $\zeta \in \mathbb{R}^N$, such that it satisfies $a_3\|\zeta\|^c \leq J(\zeta) \leq a_4\|\zeta\|^c$, for some $a_3, a_4, c \in \mathbb{R}_{>0}$. Hence,

$J(\zeta)$ is in between two continuous, strictly increasing functions of $\|\zeta\|$. It was mentioned at (3.11) that optimization of the ISS-gain for disturbance rejection is attained by minimization of $\vartheta(\|\varpi\|)$. The results of Lemma 3.4.2 further imply that this is similar to minimizing $\varsigma_1 \cdots \varsigma_N$ of ζ via $J(\zeta)$. As such, a finite dimensional optimization problem of the robust MPC, yields

$$u(t_k) := \arg_{u \in \mathbb{U}} \min J(\zeta), \quad \forall \varsigma_j \in \mathbb{R}_+ \text{ and } \forall j \in \mathbb{Z}_{[1,N]}, \quad (3.17a)$$

subject to

$$Az + Bu \in \mathbb{X}, \quad \forall z \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k), \quad (3.17b)$$

$$V(A\hat{x}(t_k) + Bu) - V_{\min}(t_k) + a_{\max}(t_k) \leq 0, \quad (3.17c)$$

$$V(A(\hat{x}(t_k) - \varpi_j(t_k)) + Bu) - V_{\min}(t_k) + a_{\max}(t_k) - \varsigma_j \leq 0. \quad (3.17d)$$

In the above formulation, (3.17a-b) characterize the optimization of the ISS-gain for disturbance rejection. The inequalities (3.17c-d) are a substitute of (3.13), and thus of (3.10), such that ISS with respect to the estimation error $\varpi(t_k)$ is guaranteed from the available variables. A formal proof of ISS is present next, for which the two sets $\mathcal{U} \subset \mathbb{R}^l$ and $\mathcal{X} \subset \mathbb{R}^n$ are defined as

$$\begin{aligned} \mathcal{U}(\hat{x}(t_k)) &:= \{u \in \mathbb{R}^l \mid (3.17b-c-d) \text{ holds } \forall \varsigma_j \in \mathbb{R}_+ \text{ and } \forall j \in \mathbb{Z}_{[1,N]}\}, \\ \mathcal{X}(t_k + \tau_s) &:= \{Ax(t_k) + Bu \mid u \in \mathcal{U}(\hat{x}(t_k))\}. \end{aligned}$$

Then $u \in \mathcal{U}(\hat{x}(t_k))$ implies that $u(t_k) = u$ is a stabilizing control action when solving (3.17), i.e., by neglecting the minimization problem of (3.17a). Further, $x(t_k + \tau_s) \in \mathcal{X}(t_k + \tau_s)$ refers to the difference inclusion corresponding to system (3.9) for each feasible control action within $\mathcal{U}(\hat{x}(t_k))$.

Theorem 3.4.3 *Let a Lyapunov function V , satisfying Assumption 3.4.1, and a cost-function J be given. Further, let a bounded set $\overline{\mathbb{W}}$ be defined, such that $\mathbb{W}(t_k) \subseteq \overline{\mathbb{W}}$, for all t_k , and suppose that (3.17) is feasible for all $\hat{x}(t_k) \in \mathbb{X} \oplus \overline{\mathbb{W}}$ and all t_k . Then the difference inclusion $x(t_k + \tau_s) \in \mathcal{X}(t_k + \tau_s)$, for all $k \in \mathbb{Z}_+$, is ISS in \mathbb{X} for inputs in $\overline{\mathbb{W}}$.*

The proof of this theorem is found in Section C.4. Notice that this theorem evaluated the worst case scenario of ς_j for proving ISS, which corresponds to the worst case evaluation of the set $\mathbb{W}(t_k)$, for all $k \in \mathbb{Z}_+$. Furthermore, the proof was presented by taking all feasible control actions $\mathcal{U}(t_k)$ into account while analyzing ISS. In reality, the proposed robust MPC provides the freedom to select a control action $u(t_k)$ that optimizes the ISS-gain of the closed-loop system by minimizing the variables of ζ via the cost-function J , i.e., (3.17a). A brief discussion on how to solve (3.17) is given next.

Implementation of robust MPC via linear programming

This section indicates certain ingredients that allow the implementation of (3.17) via linear programming, for which our attention is restricted to Lya-

Lyapunov functions that are defined by the infinity norm, i.e.,

$$V(x) = \|P_v x\|_\infty, \quad (3.18)$$

where $P_v \in \mathbb{R}^{p \times n}$ is a full-column rank matrix. Note that (3.18) satisfies the criteria of a Lyapunov function, i.e., $a_1 \|x\|^c \leq V(x) \leq a_2 \|x\|^c$, for $a_1 := \frac{\sigma_{\min}(P_v)}{\sqrt{p}}$, $a_2 := \|P_v\|_\infty$ and $c = 1$.

An implementation to compute $a_{\max}(t_k)$ of (3.16) is derived first, for which $a\|x\|^c$ is assumed to be characterized by $c = 1$ and some $a \in \mathbb{R}_{>0}$. Then, $a_{\max}(t_k)$ is directly obtained via the vertices of $\mathbb{W}(t_k)$, i.e.,

$$a_{\max}(t_k) := \max_{j \in \mathbb{Z}_{[1, N]}} (a \|\hat{x}(t_k) + \varpi_j(t_k)\|).$$

Based on the above definitions, let us continue with an implementation of (3.17). Notice that it is sufficient to impose (3.17b) only for the vertices of $\{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$, i.e., for $\cup_{j \in \mathbb{Z}_{[1, N]}} \{\varpi_j(t_k) + \hat{x}(t_k)\}$. Further, V_{\min} and a_{\max} are scalar-valued functions and ς_j are scalar-gains, for all $j \in \mathbb{Z}_{[1, N]}$. Hence, feasible computational requirements for solving (3.17) are attained by rewriting (3.17c-d) into scalar inequalities. To that extent, let us use the fact that satisfying $\|z\|_\infty \leq \alpha$, for some $z \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}_+$, is guaranteed if the same inequality holds for the absolute value of each q -th element of z , i.e., $\pm\{z\}_q \leq \alpha$. Applying this property to (3.17) implies that to satisfy (3.17c-d) it is necessary and sufficient to require that

$$\begin{aligned} \pm \{P_v(A\hat{x}(t_k) + Bu(t_k))\}_q - V_{\min}(t_k) + a_{\max}(t_k) &\leq 0, \\ \pm \{P_v(A(\hat{x}(t_k) - \varpi_j(t_k)) + Bu(t_k))\}_q - V_{\min}(t_k) + a_{\max}(t_k) - \varsigma_j &\leq 0, \end{aligned}$$

for all $q \in \mathbb{Z}_{[1, p]}$ and $j \in \mathbb{Z}_{[1, N]}$. For the remaining part of (3.17), i.e., (3.17a), let us choose an infinity-norm based cost function as well, i.e.,

$$\begin{aligned} J(x(t_k), u, \zeta) &:= \|P_J(A(\hat{x}(t_k) - \varpi_j(t_k)) + Bu)\|_\infty \\ &+ \|Q_J(\hat{x}(t_k) - \varpi_j(t_k))\|_\infty + \|R_J\|_\infty + \sum_{j \in \mathbb{Z}_{[1, N]}} \|\kappa_j \varsigma_j\|_\infty, \end{aligned} \quad (3.19)$$

for some suitable full-column rank matrices P_J , Q_J and R_J and weights $\kappa_j \in \mathbb{R}_+$. Then, minimization of the cost J as characterized in (3.17a), while taking the implications of (3.17b-c-d) into account, results in

$$u(t_k) := \arg_{u \in \mathbb{U}} \min \left(\epsilon_1 + \epsilon_2 + \sum_{j \in \mathbb{Z}_{[1, N]}} \kappa_j \varsigma_j \right), \quad \forall \varsigma_j, \epsilon_1, \epsilon_2 \in \mathbb{R}_+, \forall j \in \mathbb{Z}_{[1, N]},$$

subject to

$$\begin{aligned} A(\hat{x}(t_k) + \varpi_j(t_k)) + Bu &\in \mathbb{X}, \quad \forall j \in \mathbb{Z}_{[1, N]} \\ \pm \{P_J(A(\hat{x}(t_k) - \varpi_{j_1}(t_k)) + Bu)\}_q + \|Q_J(\hat{x}(t_k) - \varpi_{j_2}(t_k))\|_\infty &\leq \epsilon_1, \\ \pm \{R_J u\}_r &\leq \epsilon_2, \end{aligned}$$

for all $(j_1, j_2) \in \mathbb{Z}_{[1,N]} \times \mathbb{Z}_{[1,N]}$. Moreover, the latter two inequalities need to be satisfied for each q -th row of P_J and r -th row of the matrix R_J . The only thing left for implementing is to compute $V_{\min}(t_k)$. Using the same reasoning as above, it can be shown that $V_{\min}(t_k)$ of (3.15) can be formulated as a linear program. As such, implementation of the control problem (3.17) amounts to solving 2 linear programs and calculating the maximum over a finite set of real numbers, which can be performed efficiently.

3.4.2 Integration of the EBSE and robust MPC

The routine of the event based control set-up in Figure 3.1 that is still open, is the integration procedure of the EBSE with the robust MPC. Note that $\hat{x}(t_k)$ computed by the EBSE can directly be used by the robust MPC. However, the EBSE models an unbiased estimation error via the error-covariance $P(t_k) = \text{cov}(\varpi(t_k))$, while the robust MPC requires a deterministic set $\varpi(t_k) \in \mathbb{W}(t_k)$ off all possible the estimation errors. Hence, integration of the EBSE and robust MPC involves a transformation of the stochastic representation $p(\varpi(t_k)) = G(\varpi(t_k), 0, P(t_k))$ into the deterministic $\varpi(t_k) \in \mathbb{W}(t_k)$. Since all variables of this section have the same instant t_k , let us omit this time notation throughout this section, i.e., $G(\varpi(t_k), 0, P(t_k))$ becomes $G(\varpi, 0, P)$.

The integration procedure consists of two step. A representation of $p(\varpi) = G(\varpi, 0, P)$ in ellipsoidal sub-level-sets is derived first, after which \mathbb{W} is defined as an over-approximation of one of these ellipsoidal sets, such that $\varpi \in \mathbb{W}$ has a high probability.

Step 1

The ellipsoidal sub-level-sets that correspond to the Gaussian $G(\varpi, 0, P)$ were already defined via $\mathcal{E}_{0,cP} \subset \mathbb{R}^n$, for some $c \in \mathbb{R}_+$, as follows

$$\mathcal{E}_{0,cP} := \left\{ \varpi \in \mathbb{R}^n \mid \varpi^\top P^{-1} \varpi \leq c \right\}. \quad (3.20)$$

Applying some statistics, one can calculate the probability $\Pr(\cdot)$ that the estimation error ϖ is within the set $\mathcal{E}_{0,cP}$. Examples of this probability are $\Pr(\varpi \in \mathcal{E}_{0,1P}) \approx 0.68$, $\Pr(\varpi \in \mathcal{E}_{0,4P}) \approx 0.95$, and $\Pr(\varpi \in \mathcal{E}_{0,9P}) \approx 0.997$.

Step 2

The estimation error set \mathbb{W} is defined as a polytopic over-approximation of $\mathcal{E}_{0,cP}$, for a certain $c \in \mathbb{R}_+$. There is no optimal method to calculate \mathbb{W} , as it amounts to the ancient problem of “squaring the circle”. See for example the results in (Alessio et al., 2007) and the references therein. Instead, a trade-off is made between the size of \mathbb{W} on the one hand, indicating the worst case estimation error, and computational complexity of obtaining \mathbb{W} on the other hand. An illustrative example is depicted in Figure 3.6 of a tight and a fast over-approximation, where $\mathcal{E}_{0,4P}$.

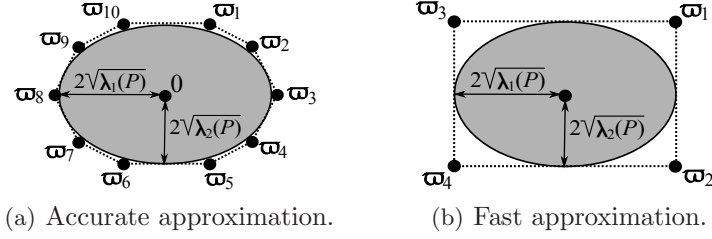


Figure 3.6: Two over-approximations of $\mathcal{E}_{0,4P}$ by \mathbb{W} .

The goal of this chapter is to have the least computational complexity. Therefore, an extension of the fast approximation to a cubic-set \mathbb{W} with length $d \in \mathbb{R}_{>0}$ is employed. This means that the vertices of \mathbb{W} are explicitly defined as all possible realizations on $\{\varpi_j\}_q \in \{-d, d\}$, for all $q \in \mathbb{Z}_{[1,n]}$ and $j \in \mathbb{Z}_{[1,N]}$. The next result presents the transformation from the Gaussian PDF $p(\varpi) = G(\varpi, 0, P)$ into the over-approximation \mathbb{W} , i.e., $\mathcal{E}_{0,cP} \subseteq \mathbb{W}$, where $c \in \mathbb{R}_+$ is a design parameter to characterize estimation errors, i.e.,

$$\mathbb{W} := \{\varpi \mid \|\varpi\|_\infty \leq d\} \quad \text{with} \quad d := \sqrt{c\lambda_{\max}(P)}. \quad (3.21)$$

Lemma 3.4.4 *Let $p(\varpi) = G(\varpi, 0, P)$ be given for some $P \in \mathbb{R}^{n \times n}$ and let \mathbb{W} satisfy (3.21). Then, for any $c \in \mathbb{R}_+$, it holds that $\mathcal{E}_{0,cP} \subset \mathbb{W}$.*

Proof: Let us consider $\mathcal{E}_{0,cP}$, which by definition yields $\varpi^\top P^{-1} \varpi \leq c$. As such, $\lambda_{\min}(P^{-1}) \|\varpi\|_\infty^2 \leq \lambda_{\min}(P^{-1}) \|\varpi\|_2^2 \leq \varpi^\top P^{-1} \varpi \leq c$. This also means that $\|\varpi\|_\infty \leq \sqrt{c(\lambda_{\min}(P^{-1}))^{-1}}$ holds. Then, applying the fact that $\lambda_{\min}(P^{-1}) = (\lambda_{\max}(P))^{-1}$ gives that $\|\varpi\|_\infty \leq \sqrt{c\lambda_{\max}(P)}$ and thus $\varpi \in \mathbb{W}$ is satisfied for all $\varpi \in \mathcal{E}_{0,cP}$, which completes the proof. ■

Lemma 3.4.4 indicates that the probability of $\varpi \in \mathbb{W}$ has a lower bound at $\Pr(\varpi \in \mathbb{W}) \geq \Pr(\varpi \in \mathcal{E}_{0,cP})$. Therefore, choosing $c = 9$ implies that $\varpi(t_k) \in \mathbb{W}(t_k)$ has at least a probability of $\Pr(\varpi \in \mathcal{E}_{0,9P}) \approx 0.997$.

Remark 3.4.5 The hypothesis of Theorem 3.4.3, i.e., there exists a constant \overline{W} such that $\mathbb{W}(t_k) \subseteq \overline{W}$ holds for all t_k , is satisfied. Let us explain this statement. Note that Theorem 3.3.1 established the property of a bounded error-covariance $P(t_k)$, for all $t_k \in \mathbb{T}$, due to which $\lambda_{\max}(P(t_k))$ is also bounded for all $t_k \in \mathbb{T}$. Then, by the definition of $\mathbb{W}(t_k)$, it holds that $\mathbb{W}(t_k) \subseteq \overline{W}$ for all t_k and $\overline{W} := \left\{ \varpi \in \mathbb{R}^n \mid \|\varpi\|_\infty \leq \sup_{t_k} \sqrt{c\lambda_{\max}(P(t_k))} \right\}$. □

This completes the overall design of the feedback loop, as it is depicted in Figure 3.1. The only aspect that should be addressed carefully is that the EBSE is a stochastic estimator, while the robust MPC algorithm is a deterministic controller. As such, since $P(t_k) = \text{cov}(\varpi(t_k))$, one does not

have a guarantee that $\varpi(t_k) \in \mathbb{W}(t_k)$ and thus that the real state is bounded. Instead, a deterministic set $\varpi(t_k) \in \mathbb{W}(t_k)$ can only be given with a certain probability. If $\varpi(t_k) \notin \mathbb{W}(t_k)$, for some $t_k \in \mathbb{R}_+$, inherent ISS is *still* attained as long as $x \in \mathbb{X}$, which is shown in the next control example.

3.5 Illustrative control example

The effectiveness of the developed EBSE and robust MPC set-up is demonstrated in a 1D position control case study. To that end, two simulations are analyzed, one in which the sensor node performs Matched sampling (MS) and one where Send-on-Delta (SoD) is employed for sampling the measurements.

Communication Measurements are sampled according to one of the employed event sampling strategies. The triggering condition of MS in (3.3) is based on the Kullback-Leibler divergence and $\Delta_{KL} = 2.8$, while SoD of Section 1.3.1 employs a level-crossing criteria on $|y(t) - y(t_{e-1})|$ with the threshold $\Delta = 0.1$. Notice that the control action $u(t)$ is not available to the sensor. Therefore, the value of the Kullback-Leibler divergence $d(p_1(x)||p_2(x))$ is computed by the sensor node according to (3.4) and $Q(\tau_e) = \begin{pmatrix} 0.5\tau_e^2 & \\ & \tau_e \end{pmatrix} 0.1$. This latter covariance is larger than the one that corresponds to $cov(w(t, \tau))$, which will be shown next, as it should model the additional uncertainty that is caused by an unknown control action u at the sensor node.

Process The process model is a controlled double integrator, i.e.,

$$\begin{aligned} x(t + \tau) &= \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix} x(t) + \begin{pmatrix} \frac{1}{2}\tau^2 \\ \tau \end{pmatrix} u(t) + w(t, \tau), \\ y(t) &= (1 \quad 0) x(t) + v(t). \end{aligned} \quad (3.22)$$

The state vector $x(t)$ combines to the object's position and speed, while only the position is measured in $y(t)$. The control input $u(t)$ is defined as the object's acceleration and both the state and control inputs are subject to the constraints $x(t) \in \mathbb{X} = [-5, 5] \times [-5, 5]$ and $u(t) \in \mathbb{U} = [-2, 2]$. The process-noise and measurement-noise are characterized as $p(w(t, \tau)) = G(w(t, \tau), 0, 3\tau \cdot 10^{-6} I_2)$ and $p(v(t)) = G(v(t), 0, 1 \cdot 10^{-4})$. Further, the sampling time of the robust MPC is $\tau_s = 0.7$ seconds.

The EBSE performs an update on the estimated state at each sample instant $t_k \in \mathbb{T}$. At the instants of an event $t_k \in \mathbb{T}_e$ a new measurement is received by the EBSE for the update. However, at the synchronous sample instants $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$ no measurement is received and the update is based on the fact that $y(t_k) \in \mathcal{H}[e|t_k]$. To that extent, the EBSE determines an *implied* measurement value $\hat{y}^1 \in \mathbb{R}^m$ with a certain covariance $R \in \mathbb{R}^{m \times m}$, see also the EBSE algorithm presented in Section 3.3. Suitable values for

both sampling strategies are the following:

$$\begin{aligned} \text{MS: } \hat{y}^1(t_k) &= C\hat{x}_2(t_k), \quad R = \frac{1}{2}(\Delta_{KL} - \alpha(t_k))\Upsilon(t_k) + V; \\ \text{SoD: } \hat{y}^1(t_k) &= y(t_{e-1}), \quad R = \frac{1}{4}\Delta^2 + V. \end{aligned}$$

Next, let us design the parameters of the robust MPC. A technique of (Lazar and Heemels, 2008) was used to compute the weight $P_V \in \mathbb{R}^{2 \times 2}$ of the Lyapunov function $V(x) = \|P_V x\|_\infty$, for $a\|x\|^c := 0.01\|x\|$, yielding $P_V = \begin{pmatrix} 2.7429 & 0.7121 \\ 0.1989 & 4.0173 \end{pmatrix}$. Following the integration procedure of Section 3.4.2, the set $\mathbb{W}(t_k) \subset \mathbb{R}^2$ will have 4 vertices. As such, the robust MPC covers 4 optimization variables $\zeta_j(t_k)$, for all $j \in \mathbb{Z}_{[1,4]}$, that each correspond to a vertex ϖ_j of $\mathbb{W}(t_k)$. The cost function $J(x(t_k), u(t_k), \zeta(t_k))$ of (3.19) is characterized by $P_J = 0.4I_4$, $Q_J = 0.2I_2$, $R_J = 0.1$ and $\Gamma_j = 4$, for all $j \in \mathbb{Z}_{[1,4]}$. The resulting linear program has 11 optimization variables and 108 constraints. In this implementation, also $V_{\min}(t_k)$ of (3.15) is calculated by solving a linear program with 3 optimization variables and 5 constraints.

The simulation scenario involves the system response in case $x(t_0) = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ with the origin as reference. The initial state estimates of the EBSE are chosen as $\hat{x}(t_0) = \begin{pmatrix} 3.5 \\ 1.2 \end{pmatrix}$ and $P(t_0) = I_2$. The evolution of the *true* state is depicted in Figure 3.7, while Figure 3.8 presents the control action. Figure 3.9 illustrates the absolute estimation error per element, i.e., $|\{\varpi(t_k)\}_1|$ and $|\{\varpi(t_k)\}_2|$, versus the corresponding bound as modeled by the length of the cubic set $\mathbb{W}(t_k)$ characterized by Lemma 3.4.4, i.e., $d = 3\lambda_{\max}^{\frac{1}{2}}(P(t_k))$.

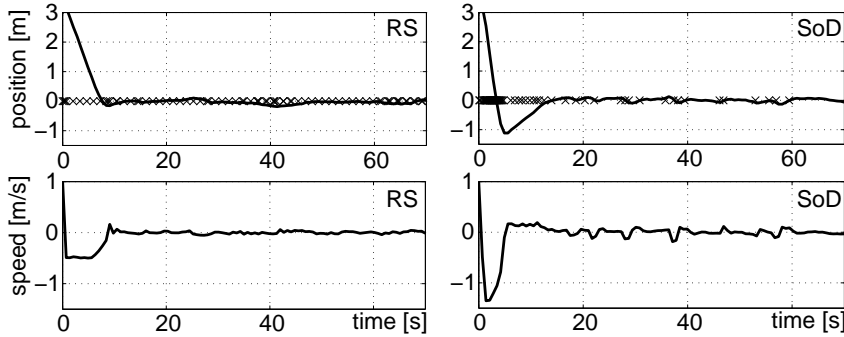


Figure 3.7: Evolution of the true position, i.e., $\{x\}_1$ and speed, i.e., $\{x\}_2$. During the simulation MS triggered 76 event instants and SoD triggered 71. Their instants are denoted with the symbols “x” in the position plot

Similar as to the estimation case of Section 3.2.3, Figure 3.7 indicates that MS triggers new event instants at a faster rate when the state shows unexpected behavior. For SoD the number of events increases when state values change fast, e.g., during the first 5 seconds. In both sampling strategies the controlled state-dynamics are stable. Further, Figure 3.8 indicates

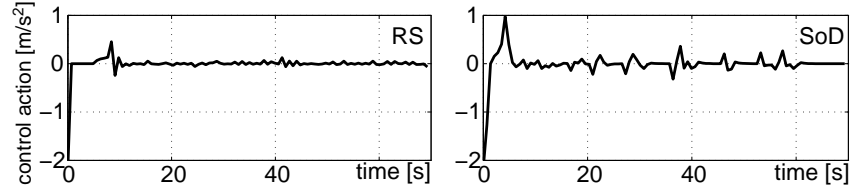


Figure 3.8: Evolution of the the control input $u(t_k)$ for MS and SoD.

that the control actions with SoD have higher impulses. This means that, as expected, the variations in the control performance with SoD are larger compared to the relatively constant control performance with MS. A similar observation is noticed in Figure 3.9 on the true and modeled estimation-error.

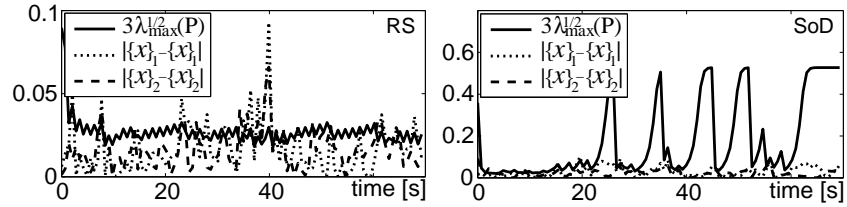


Figure 3.9: Evolution of the true estimation error $w(t_k)$ compared to the bounds that were derived from the error-covariance $P(t_k)$.

The estimation error, as depicted in Figure 3.9, indicates that most of the sample instants satisfy $w(t_k) \in \mathbb{W}(t_k)$. Nonetheless, sometimes these bounds are violated, i.e., $w(t_k) \notin \mathbb{W}(t_k)$, though the robust MPC still attains a stable closed-loop system. This is due to the fact that $x \in \mathbb{X}$ is satisfied. The figure also confirms that MS yields a more constant estimation performance of the EBSE compared SoD. Especially at approximately 23, 33, 41, 49 and 60 seconds the error-covariance $P(t_k)$ with SoD has high values.

This illustrative example shows that the considered event based control set-up, as it is depicted in Figure 2.1, is a viable alternative to existing control solutions. Moreover, the proposed method does not impose requirements on the employed event sampling strategy, nor on an ability of the set-up to trigger new events. Instead, stability of the feedback loop for any event sampling strategy is attained by introducing the EBSE.

3.6 Conclusions

This chapter discussed the integration of the event based state estimator (EBSE) in a feedback loop. The resulting event based controller is suitable

for any event sampling strategy and supports a time-synchronous control algorithm based on an estimated value of the state. Furthermore, stability the feedback control system is decoupled from the event triggering criteria, as the EBSE guarantees a bounded error-covariance at each sample instant. Based on this error-covariance, explicit polytopic bounds on the estimation error could be established that were then used in a robust MPC algorithm. It was proven that the resulting MPC controller achieves ISS to the estimation error and, moreover, it optimizes the closed-loop trajectory-dependent ISS gain. A justification of the proposed control set-up was given by addressing several integration aspects of the *stochastic* EBSE and *deterministic* MPC. A theoretical proof of closed-loop properties is a topic of future research, although simulations with two different event sampling strategies provide convincing and promising evidence of the potential of the proposed method. Additional to the integration, a novel event sampling strategy was proposed to obtain a more constant control performance by reducing variations on the estimation error. This is done by exchanging only those measurements that are of relevance to the EBSE, while still satisfying the conditions for attaining a stable EBSE.

State fusion with unknown correlations: Ellipsoidal intersection

4.1 Fusion in sensor networks 4.2 The state fusion objective 4.3 Fusion strategy and related work	4.4 Ellipsoidal intersection 4.5 Illustrative example 4.6 Conclusions
---	---

The next three chapters cover distributed state estimation. Therein, each node i typically computes a local estimate of the global state x based on its own measurement y_i and on the data shared by neighboring nodes, see also Figure 4.1. Similar as to the real state x , let this local estimate be introduced as a random vector $x_i \in \mathbb{R}^n$ that is characterized by a Gaussian distribution.

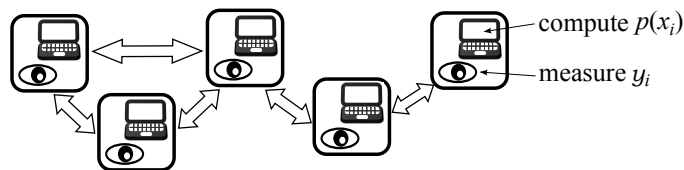


Figure 4.1: A typical set-up for distributed state estimation, where each node computes a local estimates of the global state.

This chapter in particular is concerned with a state fusion method for nodes that share local estimates. More specifically, the problem focuses on merging two prior estimates with an unknown correlation into a single fused estimate. Existing solutions lead to a conservative fusion result, as chosen parameterizations concentrate on the fusion formulas and deal with correlations afterwards. Yet, some properties of their correlation can be derived from the prior estimates beforehand. To that extent, a novel parametrization of these prior estimates is proposed, so that an explicit characterization of the correlation is obtained before deriving the fusion formulas. Then, maximizing correlation guarantees that the fusion result is based on independent parts of the prior estimates and, simultaneously, addresses the fact that correlation is unknown. A fusion example of the proposed method demonstrates a reduction in the uncertainty after fusion compared to current alternatives.

4.1 Fusion in sensor networks

Before continuing with the fusion objective, let us first point out the importance of state fusion in distributed estimation. Distributed solutions for state estimation are beneficial in networked systems when a centralized implementation for processing the produced measurements is impractical. Mostly, this is related to communication restriction of the network topology, or to the required computational power of a centralized node. To solve this issue, the estimation algorithm is distributed among the nodes, in line with the network topology. Some examples are the distributed estimation algorithms proposed in (Durant-Whyte et al., 1990; Franken and Hupper, 2005; Alriksson and Rantzer, 2006; Olfati-Saber, 2007; Khan and Moura, 2007). Therein, each node computes a local estimate of the global state, for example, by employing the Kalman filter on local measurements. As a result, the sensor network has as many local state estimates as there are nodes in the network. However, it is unlikely that these estimation results will be equal, since each node has access to a different set of local measurements. To that extent, neighboring nodes can share local estimates as input to a state fusion method. Such a method is able to synchronize and improve the estimation results by fusing multiple local estimates into a single fused estimate. The advantage of fusion over synchronization is that the resulting fused estimate remains *consistent*, i.e., the error-covariance that corresponds to a local estimate is a correct model of its estimation error.

Definition 4.1.1 Let $x_i, x \in \mathbb{R}^n$ denote two random vectors, such that x_i represents an estimate of x that is characterized by the Gaussian PDF $p(x_i) = G(x_i, \hat{x}_i, P_i)$. Then, x_i is said to be *consistent* if $\mathbb{E}[(x - x_i)(x - x_i)^\top] \preceq P_i$.

Definition 4.1.2 Two random vectors $u, v \in \mathbb{R}^n$ are said to be *independent* if they satisfy $cov(u, v) = 0_{n \times n}$. Otherwise, u and v are *correlated*.

For state fusion, it is important to observe that nodes share data, due to which the local estimates throughout the network will be correlated. This means that the state fusion method proposed in (Bar-Shalom and Campo, 1986) could be employed, as it merges two estimates given that their correlation is available. However, assuming that the correlation of local estimates is available is too restrictive for many large-scale or ad-hoc networked systems, since it amounts to keeping track of all the estimates that are shared between nodes. Therefore, this chapter presents a novel state fusion method, labeled as ellipsoidal intersection (EI), that can cope with unknown correlations. To that extent, the fusion objective is formulated next, followed by related works. After that, a derivation of the proposed fusion algorithm is presented, which is then compared to a popular fusion alternative for demonstrating the benefits of EI.

4.2 The state fusion objective

Let us consider two random vectors $x_i, x_j \in \mathbb{R}^n$, representing two prior estimates of a state $x \in \mathbb{R}^n$. Both estimates are characterized by a Gaussian distribution, for some $\hat{x}_i, \hat{x}_j \in \mathbb{R}^n$ and $P_i, P_j \in \mathbb{R}^{n \times n}$, i.e.,

$$p(x_i) = G(x_i, \hat{x}_i, P_i) \quad \text{and} \quad p(x_j) = G(x_j, \hat{x}_j, P_j). \quad (4.1)$$

The goal is to merge the prior estimates x_i and x_j into a single *fused estimate* x_f , when correlation is unknown. Similar as to the prior estimation, let the fusion result be Gaussian distributed, for some $\hat{x}_f \in \mathbb{R}^n$ and $P_f \in \mathbb{R}^{n \times n}$, i.e.,

$$p(x_f) = G(x_f, \hat{x}_f, P_f). \quad (4.2)$$

In case the correlation $\text{cov}(x_i, x_j)$ is available and both estimate are unbiased and consistent, then a fusion result that is commonly known as the Best Linear Unbiased Estimate¹ was derived in (Bar-Shalom and Campo, 1986).

Theorem 4.2.1 *Let x_i and x_j of (4.1) represent two unbiased and consistent estimates of x , for which $P_{ij} := \text{cov}(x_i, x_j)$ is given. Then, a fusion result x_f of x_i and x_j conform to (4.2) that is characterized by*

$$\begin{aligned} P_f &= P_i - (P_i - P_{ij})(P_i + P_j - P_{ij} + P_{ij}^\top)^{-1}(P_i - P_{ij}^\top) \quad \text{and} \\ \hat{x}_f &= \hat{x}_i + (P_i - P_{ij})(P_i + P_j - P_{ij} + P_{ij}^\top)^{-1}(\hat{x}_j - \hat{x}_i), \end{aligned} \quad (4.3)$$

is the consistent fused estimate with minimal $\text{tr}(P_f)$.

Since correlation is unknown, the fusion formulas in (4.3) cannot be used and x_f has to be determined via an alternative set of formulas. Irrespective of this alternative expression, note that x_f should meet an intuitive fusion property: the estimation error of x_f is at least equivalent or less than the estimation error of x_i and of x_j , as prior information is merged. A characterization of this property was presented in (Benaskeur, 2002), which uses the notion that the covariance of a consistent estimate models its estimation error.

Property 4.2.2 After fusion of the prior estimates x_i and x_j conform to (4.1), the fused estimate x_f conform to (4.2) is such that

$$P_f \preceq P_i \quad \text{and} \quad P_f \preceq P_j.$$

Therefore, the objective is to develop a fusion method that satisfies the above property and establishes a fusion algorithm that has a low computational complexity. Before a detailed derivation of EI is presented, let us state the proposed fusion strategy in relation to existing work on state fusion, next.

¹Linear denotes that x_f is Gaussian and unbiased denotes $\mathbb{E}[x - x_f] = 0$.

4.3 Fusion strategy and related work

Current fusion methods that cope with unknown correlations define a parametrization of the fusion formulas, i.e., of \hat{x}_f and P_f , before they deal with correlations. An early and still widely used parametrization was introduced as *covariance intersection* (CI) in (Julier and Uhlmann, 1997b). In case the fused estimate of CI is denoted as $p(x_f) = G(x_f, \hat{x}_f^{\text{CI}}, P_f^{\text{CI}})$, then the corresponding fusion formulas, for some $\omega \in \mathbb{R}_{[0,1]}$, yield

$$\begin{aligned} P_f^{\text{CI}} &:= (\omega P_i^{-1} + (1 - \omega) P_j^{-1})^{-1}, \\ \hat{x}_f^{\text{CI}} &:= P_f^{\text{CI}} (\omega P_i^{-1} \hat{x}_i + (1 - \omega) P_j^{-1} \hat{x}_j), \end{aligned} \quad (4.4)$$

The popularity of CI led to various approaches for determining ω , see, e.g., (Hanebeck et al., 2001; Chen et al., 2002; Franken and Hupper, 2005). However, a fulfillment of Property 4.2.2 by CI, when $P_i \neq P_j$, is restricted to two apparent cases: $\omega = 0$ and $P_j \preceq P_i$, or $\omega = 1$ and $P_i \preceq P_j$. For any other case neither $P_f^{\text{CI}} \preceq P_i$ nor $P_f^{\text{CI}} \preceq P_j$ holds, which is illustrated in Figure 4.2. More recent studies showed that $\mathcal{E}_{0,P_f} \subseteq \mathcal{E}_{0,P_i} \cap \mathcal{E}_{0,P_j}$ is sufficient to meet Property 4.2.2. This set-relation was applied in (Benaskeur, 2002) to find P_f . However, the expression of \hat{x}_f is derived by assuming independent x_i and x_j , which is contradictory to the problem formulation. Another fusion method along the same line was proposed in (Zhuo and Li, 2008). Therein, an iterative algorithm from (Kurzhanskiy and Varaiya, 2006) is employed to determine \mathcal{E}_{0,P_f} as the largest ellipsoid contained in $\mathcal{E}_{0,P_i} \cap \mathcal{E}_{0,P_j}$. While certain aspects of the procedure of (Zhuo and Li, 2008) require further clarification, the iterative nature of this solution already restricts its applicability.

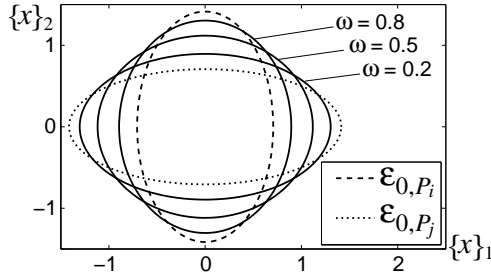


Figure 4.2: An example of P_f^{CI} , for three values of ω , in terms of their sub-level-sets \mathcal{E}_{0,P_i} , \mathcal{E}_{0,P_j} and $\mathcal{E}_{0,P_f^{\text{CI}}}$ (solid lines). Notice that $\mathcal{E}_{0,P_f^{\text{CI}}} \not\subseteq \mathcal{E}_{0,P_i} \cap \mathcal{E}_{0,P_j}$ is a consequence of $P_{i_f}^{\text{extscci}} \not\preceq P_i$ and $P_{i_f}^{\text{extscci}} \not\preceq P_j$, see also Figure 1.9.

Overall, the issues of existing fusion methods are a consequence of parameterizing x_f and deal with correlations afterwards, e.g., CI in (4.4). Hence, existing parameterizations account for a restricted representation of correlation based on the fused estimate. Instead, the EI fusion method proposes a

novel parametrization of the prior estimates, by which an explicit characterization of the unknown correlation is established first. This parametrization involves the introduction of three new estimates that each represents an *independent* part of the prior estimates x_i and x_j . Algebraic fusion formulas are then derived by merging these newly introduced independent estimates according to the fusion result of Theorem 4.2.1. After that, the characterization of $\text{cov}(x_i, x_j)$ is used to find the maximum correlation and thereby, guarantee that these new estimates are indeed independent. An upper bound on this maximum is found along the inequalities that satisfy Property 4.2.2.

4.4 Ellipsoidal intersection

The explanation of EI as a state fusion method will make use of the next result, which follows from Theorem 4.2.1. To that extent, let us introduce $\Omega : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ as a function of two independent random vectors $x_1, x_2 \in \mathbb{R}^n$ that are Gaussian distributed, i.e., $p(x_1) = G(x_1, \hat{x}_1, P_1)$ and $p(x_2) = G(x_2, \hat{x}_2, P_2)$, such that

$$\Omega(x_1, x_2) := (P_1^{-1} + P_2^{-1})^{-1} (P_1^{-1}x_1 + P_2^{-1}x_2). \quad (4.5)$$

Corollary 4.4.1 *Let x_1 and x_2 represent two unbiased and consistent estimates of x that are independent. Then, $x_3 := \Omega(x_1, x_2)$ is a Gaussian distributed random vector characterized by*

$$P_3 := (P_1^{-1} + P_2^{-1})^{-1}, \quad \hat{x}_3 := P_3 (P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2), \quad (4.6)$$

such that x_3 is the consistent fused estimate with minimal $\text{tr}(P_3)$.

The proof of Corollary 4.4.1 is presented in Appendix D.1. It is worth to point out that this proof makes use of the results in Theorem 4.2.1, due to which the fusion formulas in (4.6) attain the Best Linear Unbiased Estimate.

4.4.1 Characterization of the unknown correlation

This section derives an explicit characterization of the unknown correlation, after which the EI fusion formulas are derived. Characterizing $\text{cov}(x_i, x_j)$ is based on a parametrization of the prior estimates. To that extent, let us propose that each prior estimate x_i and x_j represents a fusion result determined by the function $\Omega(\cdot)$ and Corollary 4.4.1. Then, by introducing $x_{ij}, x_{ii}, x_{jj} \in \mathbb{R}^n$ as Gaussian distributed random vectors, for some $\gamma, \theta_i, \theta_j \in \mathbb{R}^n$ and $\Gamma, \Theta_i, \Theta_j \in \mathbb{R}^{n \times n}$, the proposed parametrization, yields

$$\begin{aligned} x_i &= \Omega(x_{ij}, x_{ii}) \quad \text{and} \quad x_j = \Omega(x_{ij}, x_{jj}), \\ \text{with } p(x_{ii}) &:= G(x_{ii}, \theta_i, \Theta_i), \quad p(x_{jj}) := G(x_{jj}, \theta_j, \Theta_j), \\ p(x_{ij}) &:= G(x_{ij}, \gamma, \Gamma). \end{aligned} \quad (4.7)$$

Fulfilling the premise of Corollary 4.4.1 further implies the following:

Assumption 4.4.2 x_{ii} , x_{ij} and x_{jj} are pair-wise independent.

Note that the proposed parametrization in (4.7) and Assumption 4.4.2 involves three newly introduced estimates x_{ii} , x_{ij} and x_{jj} that each corresponds to an independent part of the prior estimates x_i and x_j . A fulfillment of Assumption 4.4.2 is the topic of Section 4.4.2, since this section continues with a description of the correlation followed by the fusion formulas.

The explicit characterization of $cov(x_i, x_j)$, for independent x_{ii} , x_{ij} and x_{jj} , is obtained by substituting x_i and x_j of (4.7) into $cov(x_i, x_j)$, i.e.,

$$\begin{aligned} cov(x_i, x_j) &:= \mathbb{E}[x_i x_j^\top] - \mathbb{E}[x_i] \mathbb{E}[x_j^\top] \\ &= \mathbb{E}[P_i(\Gamma^{-1} x_{ij} + \Theta_i^{-1} x_{ii})(x_{ij}^\top \Gamma^{-1} + x_{jj}^\top \Theta_j^{-1}) P_j] \\ &\quad - P_i(\Gamma^{-1} \gamma + \Theta_i^{-1} \theta_i)(\gamma^\top \Gamma^{-1} + \theta_j^\top \Theta_j^{-1}) P_j \\ &= P_i \Gamma^{-1} P_j. \end{aligned} \quad (4.8)$$

Further, the fact that x_{ii} , x_{ij} and x_{jj} are pair-wise independent, while $x_i = \Omega(x_{ij}, x_{ii})$ and $x_j = \Omega(x_{ij}, x_{jj})$, implies that fusion of x_i and x_j is now equivalent to fusing x_{ij} , x_{ii} and x_{jj} . Fusion of multiple prior estimate is commonly conducted recursively and since x_{ii} , x_{ij} and x_{jj} are independent, they can be fused via the fusion function $\Omega(\cdot)$ and Corollary 4.4.1.

Proposition 4.4.3 *Let the prior estimates x_i and x_j of (4.1) be given, such that they satisfy (4.7) and Assumption 4.4.2. Then, the fused estimate x_f of (4.2) is equivalent to $x_f = \Omega(x_{ij}, x_{ii}, x_{jj})$, or similarly*

$$\begin{aligned} x_f &= \Omega(x_i, x_{jj}), \\ \text{where, } P_f &= (P_i^{-1} + \Theta_j^{-1})^{-1}, \quad \hat{x}_f = P_f (P_i^{-1} \hat{x}_i + \Theta_j^{-1} \theta_j). \end{aligned} \quad (4.9)$$

A derivation of the above proposition is found in Appendix D.2. Basically, the fusion result of (4.9) indicates that the prior estimate x_i is fused with the independent part x_{jj} of the other prior estimate x_j . Note that the alternative fusion formulas are thus obtained by substituting $x_j = \Omega(x_{ij}, x_{jj})$, i.e., $\Theta_j^{-1} = P_j^{-1} - \Gamma^{-1}$ and $\Theta_j^{-1} \theta_j = P_j^{-1} \hat{x}_j - \Gamma^{-1} \gamma$, into (4.9), due to which

$$\begin{aligned} P_f &= (P_i^{-1} + P_j^{-1} - \Gamma^{-1})^{-1}, \\ \hat{x}_f &= P_f (P_i^{-1} \hat{x}_i + P_j^{-1} \hat{x}_j - \Gamma^{-1} \gamma). \end{aligned} \quad (4.10)$$

The unknown variables Γ and γ of (4.10) correspond to the unknown correlation $cov(x_i, x_j)$. A characterization of $cov(x_i, x_j)$ was derived in (4.8). This characterization can thus be used to find explicit expressions of the *mutual covariance* Γ and the *mutual mean* γ . The main objective when determining Γ will be to attain Assumption 4.4.2, for which the correlation in (4.8) should be maximized. Therefore, let us first explain what is meant with maximizing

the correlation of prior estimates, before deriving an explicit formula of the mutual covariance Γ .

Maximizing a correlation matrix

This intermezzo presents a method to find the “maximum” correlation of the two prior estimates x_i and x_j . It makes use of an alternative definition for correlation proposed in (Edwards, 1979).

Definition 4.4.4 Two random vectors $x_i, x_j \in \mathbb{R}^n$ are said to be *correlated* if there exists a matrix $H \in \mathbb{R}^{n \times n}$ and a Gaussian distributed random vector $y \in \mathbb{R}^n$, such that $x_i = Hx_j + y$ holds and x_j is independent to y . Moreover, x_i and x_j are said to be *fully correlated* when $x_i = Hx_j$, i.e., $p(y) = G(y, 0, 0_{n \times n})$.

The following characterization is derived from Definition 4.4.4:

$$P_i \equiv \text{cov}(x_i) = \text{cov}(Hx_j + y) = HP_jH^\top + \text{cov}(y), \quad (4.11)$$

$$P_{ij} := \text{cov}(x_i, x_j) = \text{cov}(Hx_j + y, x_j) = HP_j. \quad (4.12)$$

Substituting (4.12) into (4.11), yields

$$P_i = P_{ij}P_j^{-1}P_{ij}^\top + \text{cov}(y). \quad (4.13)$$

By (4.13), note that $P_i \succeq P_{ij}P_j^{-1}P_{ij}^\top$ holds when x_i and x_j are correlated, as $x_i = Hx_j + y$ and $\text{cov}(y) \succ 0$, which turns into $P_i = P_{ij}P_j^{-1}P_{ij}^\top$ if x_i and x_j are fully correlated. Hence, maximizing the correlation of x_i and x_j means to reduce the effect of y on x_i and thus to find the “smallest” matrix $\text{cov}(y) = P_i - P_{ij}P_j^{-1}P_{ij}^\top \succ 0$. Since, for any $A, B \succeq 0$, we have that $\text{tr}(A) \leq \text{tr}(B)$ if $A \preceq B$ (Fact 8.12.25 of (Bernstein, 2005)), the desired “maximum” correlation P_{ij}^{\max} can be found as follows:

$$P_{ij}^{\max} := \arg \min_{P_{ij} \in \mathbb{R}^{n \times n}} \text{tr}(P_i - P_{ij}P_j^{-1}P_{ij}^\top),$$

$$\text{subject to } P_i \succeq P_{ij}P_j^{-1}P_{ij}^\top.$$

Note that $\min \text{tr}(P_i - P_{ij}P_j^{-1}P_{ij}^\top) = \max \text{tr}(P_{ij}P_j^{-1}P_{ij}^\top)$. Further, the fact that $P_{ij}P_j^{-1}P_{ij}^\top \succeq 0$, implies that $\text{tr}(P_{ij}P_j^{-1}P_{ij}^\top) = \log |P_{ij}P_j^{-1}P_{ij}^\top| = 2 \log |P_{ij}| - \log |P_j|$. As such, the above characterization of P_{ij}^{\max} is equivalently found via

$$P_{ij}^{\max} := \arg \max_{P_{ij} \in \mathbb{R}^{n \times n}} \log |P_{ij}|, \quad (4.14)$$

$$\text{subject to } P_i \succeq P_{ij}P_j^{-1}P_{ij}^\top.$$

4.4.2 An explicit formula of the mutual covariance

This section defines the *mutual covariance* Γ as the result of a minimization problem. Then, an explicit solution to this problem is derived, which yields a fusion result for P_f that meets Property 4.2.2.

The objective when determining the value of Γ is to attain Assumption 4.4.2. Since x_{ii} and x_{ij} are already independent, as well as x_{jj} and x_{ij} , note that an independent x_{ii} and x_{jj} implies that any correlation of x_i and x_j should be represented by x_{ij} only. A characterization that corresponds to such a correlation was derived in (4.8) as $\text{cov}(x_i, x_j) = P_i \Gamma^{-1} P_j$. Therefore, “maximizing” $\text{cov}(x_i, x_j)$ of (4.8) fulfills the desired property that any (unknown) correlation is solely described by x_{ij} , due to which x_i and x_{jj} in (4.9) are independent. A solution for finding this maximum correlation is obtained with the optimization problem of (4.14), i.e., substituting $P_{ij} = P_i \Gamma^{-1} P_j$. The condition of (4.14) then becomes $P_i \succeq \Gamma P_j^{-1} \Gamma$. However, note that $x_j = \Omega(x_{ij}, x_{jj})$ of (4.7), yields $\Theta_j^{-1} = P_j^{-1} - \Gamma^{-1} \succ 0$ and thus $\Gamma \succeq P_j$. Similarly, $\Gamma \succeq P_i$ must hold as well. A fulfillment of $\Gamma \succeq P_j$ and $\Gamma \succeq P_i$ automatically satisfies $P_i \succeq \Gamma P_j^{-1} \Gamma$ as well. Hence, $P_i \succeq \Gamma P_j^{-1} \Gamma$ can be removed from the optimization problem when $\Gamma \succeq P_j$ and $\Gamma \succeq P_i$ are included. In addition, it is worth to point out that substituting $P_{ij} = P_i \Gamma^{-1} P_j$ into $\max \log |P_{ij}|$ of (4.14) results in $\max \log |P_i \Gamma^{-1} P_j| = \min \log |\Gamma|$, as $|P_i \Gamma^{-1} P_j| = |P_i| |\Gamma|^{-1} |P_j| > 0$. Hence, Γ is obtained as the unique solution of a convex optimization problem, i.e.,

$$\begin{aligned} \Gamma := \arg \min_{\Upsilon \in \mathbb{R}^{n \times n}} \log |\Upsilon|, \\ \text{subject to } \Upsilon \succeq P_i, \Upsilon \succeq P_j. \end{aligned} \quad (4.15)$$

It was shown in (Boyd and Vandenberghe, 2004) that the ellipsoid $\mathcal{E}_{0,\Gamma}$, i.e., characterized by Γ of (4.15), is the *Löwner-John* ellipsoid of $\mathcal{E}_{0,P_i} \cup \mathcal{E}_{0,P_j}$. This means that $\mathcal{E}_{0,\Gamma}$ is the smallest ellipsoid that contains $\mathcal{E}_{0,P_i} \cup \mathcal{E}_{0,P_j}$. Figure 4.3 illustrates an example of such a *Löwner-John* ellipsoid $\mathcal{E}_{0,\Gamma}$ that corresponds to $P_i = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$ and $P_j = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & 2 \end{pmatrix}$.

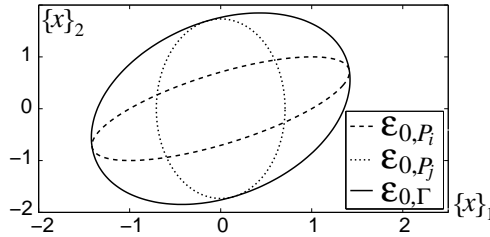


Figure 4.3: An illustration of $\mathcal{E}_{0,\Gamma}$ calculated conform to (4.15).

An explicit solution of Γ that is equal to the result of (4.15) is provided along the properties of the next two lemmas. The first property is that

the optimization problem of (4.15) is invariant to linear transformations. To that extent, let $T \in \mathbb{R}^{n \times n}$ denote a *nonsingular* matrix, due to which $T^{-\top} := (T^\top)^{-1} = (T^{-1})^\top$, see Proposition 2.6.8 of (Bernstein, 2005). This nonsingular transformation matrix T is then used to define the following transformed covariances, i.e.,

$$\hat{P}_i = T^{-1}P_iT^{-\top} \quad \text{and} \quad \hat{P}_j = T^{-1}P_jT^{-\top}, \quad (4.16)$$

along with the optimization problem

$$\begin{aligned} \hat{\Gamma} &:= \arg \min_{\hat{\Upsilon} \in \mathbb{R}^{n \times n}} \log |\hat{\Upsilon}|, \\ &\text{subject to} \quad \hat{\Upsilon} \succeq \hat{P}_i, \hat{\Upsilon} \succeq \hat{P}_j. \end{aligned} \quad (4.17)$$

Lemma 4.4.5 *Let $P_i, P_j \succ 0$ be given and let $\hat{P}_i, \hat{P}_j, \hat{\Gamma}$ satisfy (4.16) and (4.17), for some nonsingular $T \in \mathbb{R}^{n \times n}$. Then $\Gamma = T\hat{\Gamma}T^\top$.*

The second property is an explicit result of Γ for a diagonal P_i and P_j . To that end, let the diagonal matrix $D_\Gamma \in \mathbb{R}^{n \times n}$, for some diagonal $D_j \in \mathbb{R}^{n \times n}$, be defined with the following diagonal elements:

$$\{D_\Gamma\}_{qq} := \max\{1, \{D_j\}_{qq}\}, \quad \forall q \in \mathbb{Z}_{[1,n]}. \quad (4.18)$$

Lemma 4.4.6 *Let $P_i = I_n$ and $P_j = D_j$, be given for some diagonal matrix $D_j \succ 0$. Then $\Gamma = D_\Gamma$.*

Lemma 4.4.5 and Lemma 4.4.6 are proven in Appendix D.3 and Appendix D.4, respectively. They are instrumental in the proof of the next result, which derives the desired algebraic expression of Γ that substitutes (4.15). This is then followed by the proof that EI satisfies Property 4.2.2. To that extent, let us introduce the diagonal matrices $D_i, D_j \in \mathbb{R}^{n \times n}$ and rotational matrices $S_i, S_j \in \mathbb{R}^{n \times n}$ via the eigenvalue decompositions²

$$P_i = S_i D_i S_i^{-1}, \quad D_i^{-\frac{1}{2}} S_i^{-1} P_j S_i D_i^{-\frac{1}{2}} = S_j D_j S_j^{-1}, \quad (4.19a)$$

$$\text{and let } T := S_i D_i^{\frac{1}{2}} S_j. \quad (4.19b)$$

Theorem 4.4.7 *For any $P_i, P_j \succ 0$, the optimal covariance Γ defined by (4.15) is equal to $TD_\Gamma T^\top$.*

Theorem 4.4.7 is proven in Appendix D.5. The proof starts with a transformation of P_i, P_j and Γ according to (4.17) and (4.19b), i.e., $\hat{P}_i = T^{-1}P_iT^{-\top}$, $\hat{P}_j = T^{-1}P_jT^{-\top}$ and $\hat{\Gamma} = T^{-1}\Gamma T^{-\top}$. The transformed covariances \hat{P}_i and \hat{P}_j then satisfy the premise of Lemma 4.4.6, due to which $\hat{\Gamma}$ satisfies (4.18).

² $A = SDS^{-1}$ denotes the eigenvalue decomposition of $A \in \mathbb{R}^{n \times n}$, where S and D are characterized by the eigenvectors and eigenvalues of A , respectively.

Theorem 4.4.8 *Let x_i and x_j of (4.1) be fused according to (4.10). Then, $P_f \preceq P_i$ and $P_f \preceq P_j$.*

Proof: Since $P_i \preceq \Gamma$ and $P_j \preceq \Gamma$ (see (4.15)), $P_j^{-1} \succeq \Gamma^{-1}$ holds and thus $P_j^{-1} - \Gamma^{-1} \succeq 0$. Adding P_i^{-1} on both sides results in $P_i^{-1} + P_j^{-1} - \Gamma^{-1} \succeq P_i^{-1}$. Equation (4.10) shows that the left hand side of this inequality is P_f^{-1} , i.e., $P_f^{-1} \succeq P_i^{-1}$ and thus $P_f \preceq P_i$. The property $P_f \preceq P_j$ is proven similarly. ■

Let us continue by deriving an explicit formula of the mutual mean γ .

4.4.3 An explicit formula for the mutual mean

The *mutual mean* γ is the expected value of the random vector x_{ij} . The most essential criterion for γ is that the relation between the first moment $\gamma := \mathbb{E}[x_{ij}]$ and the second moment $\Gamma := \mathbb{E}(x_{ij}x_{ij}^\top) - \gamma\gamma^\top$ is maintained. Otherwise, this first and second moment of x_{ij} have conflicting results, due to which x_{ij} will not be a *consistent* estimate. Therefore, the derivation of γ should be analogous to the steps for deriving Γ in the proof of Theorem 4.4.7 but then translated from the second into the first moment. This translation is described in the next two steps.

1. Transformation The derivation of Γ in the proof of Theorem 4.4.7 started with the transformations $\hat{P}_i = T^{-1}P_iT^{-\top}$, $\hat{P}_j = T^{-1}P_jT^{-\top}$ and $\hat{\Gamma} = T^{-1}\Gamma T^{-\top}$. Similar transformations in the first moment, yield

$$\mu_i := T^{-1}\hat{x}_i, \quad \mu_j := T^{-1}\hat{x}_j, \quad \mu_\gamma = T^{-1}\gamma. \quad (4.20)$$

2. Characterization These transformations resulted in $\hat{P}_i = I_n$, $\hat{P}_j = D_j$ and $\hat{\Gamma} = D_\Gamma$, which after substitution in (4.18) gave the optimal solution for $\hat{\Gamma}$ via $\{\hat{\Gamma}\}_{qq} = \max\{\{\hat{P}_i\}_{qq}, \{\hat{P}_j\}_{qq}\}$, for all $q \in \mathbb{Z}_{[1,n]}$. A similar characterization in the first moment implies that the elements of μ_γ are defined, for all $q \in \mathbb{Z}_{[1,n]}$, as follows:

$$\{\mu_\gamma\}_q := \begin{cases} \{\mu_i\}_q & \text{if } \{\hat{P}_j\}_{qq} < \{\hat{P}_i\}_{qq}, \\ \{\mu_j\}_q & \text{if } \{\hat{P}_j\}_{qq} > \{\hat{P}_i\}_{qq}, \\ 0.5(\{\mu_i\}_q + \{\mu_j\}_q) & \text{if } \{\hat{P}_j\}_{qq} = \{\hat{P}_i\}_{qq}, \end{cases} \quad (4.21a)$$

$$\text{with } \hat{P}_i := I_n, \quad \hat{P}_j := D_j \quad \text{and} \quad \hat{\Gamma} = D_\Gamma. \quad (4.21b)$$

The above matrix D_j is equal to the one that was obtained in (4.19a). Notice that, by (4.20), the mutual mean can be calculated as $\gamma = T\mu_\gamma$. However, $\{\mu_\gamma\}_q$ is sensitive to numerical errors in the eigenvalue decompositions (or singular value decompositions) that are necessary to determine T of (4.19b). Especially, when $\{\hat{P}_i\}_{qq} \approx \{\hat{P}_j\}_{qq}$, i.e., $\{D_j\}_{qq} \approx 1$. To solve this issue, let us derive a numerically robust expression for $\gamma = T\mu_\gamma$ by approximating μ_γ

of (4.21). To that extent, consider a situation that $\{D_j\}_{qq} \neq 1$ holds for all $q \in \mathbb{Z}_{[1,n]}$. Then $\{\mu_\gamma\}_q$ in (4.21) is equivalent to

$$\{\mu_\gamma\}_q = \left((\{D_j\}_{qq} - \{D_\Gamma\}_{qq}) + (1 - \{D_\Gamma\}_{qq}) \right)^{-1} \times \left((\{D_j\}_{qq} - \{D_\Gamma\}_{qq})\{\mu_i\}_{qq} + (1 - \{D_\Gamma\}_{qq})\{\mu_j\}_q \right). \quad (4.22)$$

To prove that (4.22) is equivalent to (4.21a), one can substitute the two cases when $\{D_j\}_{qq} \neq 1$, while using $\{D_\Gamma\} = \max\{1, \{D_j\}\}$ of (4.18), i.e., $\{D_\Gamma\}_{qq} = 1$, if $\{D_j\}_{qq} < 1$, and $\{D_\Gamma\}_{qq} = \{D_j\}_{qq}$, if $\{D_j\}_{qq} > 1$, for all $q \in \mathbb{Z}_{[1,n]}$. Furthermore, since all matrices are diagonal, one can rewrite (4.22) as follows:

$$\mu_\gamma = \left((D_j - D_\Gamma) + (I_n - D_\Gamma) \right)^{-1} \left((D_j - D_\Gamma)\mu_i + (I_n - D_\Gamma)\mu_j \right).$$

Next, let us take the third case of (4.21) into account as well, i.e., $\{\mu_\gamma\}_q = 0.5(\{\mu_i\}_{qq} + \{\mu_j\}_q)$ if $\{D_j\}_{qq} \approx 1$. Then, by extending the above expression of μ_γ , a suitable approximation, for some small scalar value $\epsilon \in \mathbb{R}_{>0}$, yields

$$\mu_\gamma \approx (W_i + W_j)^{-1} (W_i\mu_i + W_j\mu_j), \quad (4.23a)$$

$$W_i := D_j^{-1} - D_\Gamma^{-1} + \varsigma I_n, \quad W_j := I_n - D_\Gamma^{-1} + \varsigma I_n, \quad (4.23b)$$

$$\varsigma := \begin{cases} 0 & \text{if } |\{D_j\}_{qq} - 1| \geq 10\epsilon, \forall q \in \mathbb{Z}_{[1,n]}, \\ \epsilon & \text{otherwise.} \end{cases} \quad (4.23c)$$

Note that the diagonal matrices D_j and D_Γ of (4.23) are inverted, whereas this was not the case in prior expressions of μ_γ . The inversion does not change the result of μ_γ but was done to simplify the derivation of an algebraic formula for γ . Such a formula is now found by substituting μ_γ of (4.23a) into $\gamma = T\mu_\gamma$, i.e., $\gamma \approx T(W_i + W_j)^{-1} (W_i\mu_i + W_j\mu_j)$. Using $T^{-1}T = I_n$, this is further rewritten into

$$\begin{aligned} \gamma &\approx T(W_i + W_j)^{-1} T^\top T^{-\top} (W_i T^{-1} T \mu_i + W_j T^{-1} T \mu_j) \\ &= (T^{-\top} W_i T^{-1} + T^{-\top} W_j T^{-1})^{-1} \times \\ &\quad (T^{-\top} W_i T^{-1} T \mu_i + T^{-\top} W_j T^{-1} T \mu_j). \end{aligned} \quad (4.24)$$

Note that $T^{-\top} W_i T^{-1} = P_j^{-1} - \Gamma^{-1} + \varsigma P_i^{-1}$ and $T^{-\top} W_j T^{-1} = P_i^{-1} - \Gamma^{-1} + \varsigma P_i^{-1}$, where ς is the approximation parameter defined in (4.23). When simplifying these transformations into $T^{-\top} W_i T^{-1} \approx P_j^{-1} - \Gamma^{-1} + \varsigma I_n$ and $T^{-\top} W_j T^{-1} \approx P_i^{-1} - \Gamma^{-1} + \varsigma I_n$, while observing that $\hat{x}_i = T\mu_i$ and $\hat{x}_j = T\mu_j$, the approximation of γ in (4.24) results in the following algebraic expression:

$$\begin{aligned} \gamma &\approx (P_i^{-1} + P_j^{-1} - 2\Gamma^{-1} + 2\varsigma I_n)^{-1} \times \\ &\quad ((P_j^{-1} - \Gamma^{-1} + \varsigma I_n)\hat{x}_i + (P_i^{-1} - \Gamma^{-1} + \varsigma I_n)\hat{x}_j). \end{aligned} \quad (4.25)$$

Simulation further verify that the value of $\epsilon \in \mathbb{R}_{>0}$ in (4.23c), which determines ς , has a negligible effect on the above approximation of the mutual mean.

This completes the EI state fusion method. The basic fusion formulas are given by (4.10), whereas the algebraic expressions for calculating Γ and γ are given in Theorem 4.4.7 and (4.25), respectively. Hence, for S_i , D_i , S_j and D_j as specified in (4.19a) and ς conform to (4.23c), the EI algorithm is summarized as follows:

Algorithm of EI

$$\begin{aligned}
 P_f &= (P_i^{-1} + P_j^{-1} - \Gamma^{-1})^{-1}; \\
 \hat{x}_f &= P_f(P_i^{-1}\hat{x}_i + P_j^{-1}\hat{x}_j - \Gamma^{-1}\gamma); \\
 \text{where } D_\Gamma &= \text{diag}_{q \in \mathbb{Z}_{[1,n]}} (\max\{1, \{D_j\}_{qq}\}); \\
 \Gamma &= S_i D_i^{\frac{1}{2}} S_j D_\Gamma S_j^{-1} D_i^{\frac{1}{2}} S_i^{-1}; \\
 \gamma &= (P_i^{-1} + P_j^{-1} - 2\Gamma^{-1} + 2\varsigma I_n)^{-1} \times \\
 &\quad ((P_j^{-1} - \Gamma^{-1} + \varsigma I_n)\hat{x}_i + (P_i^{-1} - \Gamma^{-1} + \varsigma I_n)\hat{x}_j).
 \end{aligned}$$

4.5 Illustrative example

This section focusses on a pure fusion example to study the results of EI in relation to the alternative fusion method CI. A more elaborated case-study to assess EI and CI in distributed state estimation applications is part of the next chapter. In the considered fusion example the two prior estimates x_i and x_j are characterized by $\hat{x}_i = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}$, $P_i = \begin{pmatrix} 2.5 & -1 \\ -1 & 1.2 \end{pmatrix}$, $\hat{x}_j = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ and $P_j = \begin{pmatrix} 0.8 & -0.5 \\ -0.5 & 4 \end{pmatrix}$. Then, Figure 4.4 depicts the results of EI and CI.

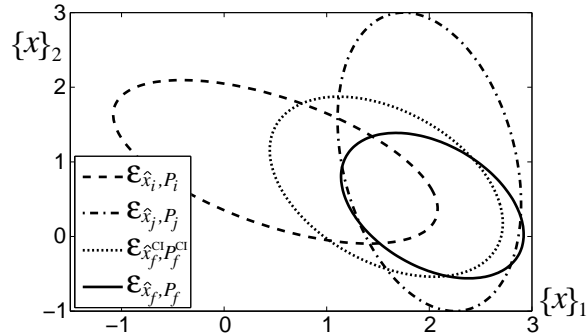


Figure 4.4: Two prior and fused estimates according to EI and CI.

The figure indicates that EI obtains a fused estimate with the smallest uncertainty in every direction of the state-space defined by x , as the ellipsoid $\mathcal{E}_{\hat{x}_f, P_f}$ is smaller in area than $\mathcal{E}_{\hat{x}_f^{\text{CI}}, P_f^{\text{CI}}}$. The result stems from the fact that

EI, in contrast to CI, satisfies Property 4.2.2. This property formalizes the intuition that estimation-error (uncertainty) after fusion should be reduced.

A second example, in which $\hat{x}_i = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$, $P_i = \begin{pmatrix} 1.5 & -0.4 \\ -0.4 & 0.5 \end{pmatrix}$, $\hat{x}_j = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$ and $P_j = \begin{pmatrix} 0.34 & 0.1 \\ 0.1 & 1 \end{pmatrix}$, is depicted in Figure 4.5. Therein, the results of EI are presented for three values of ϵ , which is a parameter to support a numerically stable approximation of γ . Note that the fusion results are almost similar, as the ellipsoidal sub-level-sets $\mathcal{E}_{\hat{x}_f, P_f}$ are on top of each other. As such, the sub-optimality of an approximated mutual mean γ by ϵ is negligible.

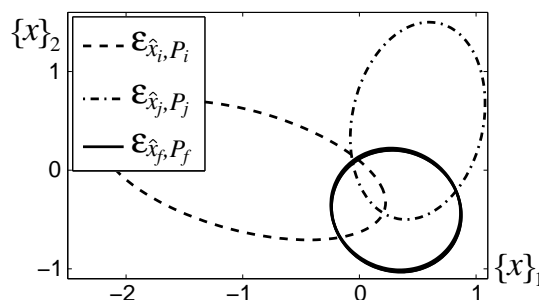


Figure 4.5: Two prior and fused estimates according to EI for three different values of ϵ , i.e., 0.001, 0.01 and 0.1.

Remark 4.5.1 The correlation of x_i and x_j , in case of pair-wise independent x_{ii} , x_{ij} and x_{jj} , was derived in (4.8) as $\text{cov}(x_i, x_j) = P_i \Gamma^{-1} P_j$. This means that one can substitute $P_{ij} = P_i \Gamma^{-1} P_j$ of the “best linear unbiased estimate” (BLUE) in Theorem 4.2.1, where Γ has a value that corresponds to Section 4.4.2. Simulations verify that such a BLUE fusion result is equal to x_f of EI. However, it should still be verified whether EI obtains a consistent fusion result.

4.6 Conclusions

A novel state fusion method was proposed, labeled as ellipsoidal intersection, for fusing two Gaussian estimates of the same state with an unknown correlation. The distinguishing feature of ellipsoidal intersection, compared to alternative fusion methods, is an explicit characterization of the unknown correlation *a priori* to the derivation of fusion formulas. This enables a fusion method that is based on merging independent parts of the prior estimates only. Besides enabling a fusion algorithm with a low computational complexity, ellipsoidal intersection guarantees that the uncertainty after fusion is reduced with respect to the prior estimates. A fusion example already demonstrated the effectiveness of ellipsoidal intersection, though a more elaborated comparison is made in the next chapter.

Cooperative state estimation

5.1 System description 5.2 Cooperative Kalman filters 5.3 Illustrative example	5.4 Nonlinear state estimation 5.5 Conclusions
--	---

Distributed state estimation refers to a method for computing a local estimate of the global state in each node of the networked system. Typically, current methods focus on optimizing the local estimation results per node individually. However, since nodes interact with each other, the estimation algorithm performed by each node will result in an *emerging* behavior on the network level. Therefore, an alternative approach is to optimize the estimation results from a network point of view via a desired emerging behavior. This chapter introduces *cooperation* as the desired emerging behavior. Cooperation is characterized by satisfying the global covariance property, for which a novel distributed state-estimator is derived. In the proposed solution, nodes perform a state-estimator locally, such as the Kalman filter, to process their measurement and obtain a local estimate. Neighboring nodes then share their local estimate for state fusion. An asymptotic analysis of the proposed solution not only studies the conditions for establishing stable estimation results throughout the network but, simultaneously, proves that global covariance is attained. In addition, the benefits of cooperation in estimation are demonstrated in two illustrative case studies: a scalable platoon of vehicles (linear) and tracking shockwaves on a highway (nonlinear).

5.1 System description

Distributed state estimation refers to a collection of estimation strategies often used in large-scale or ad-hoc sensor networks. Each node of such a network is equipped with a CPU, a sensor and a two-way radio, allowing nodes to exchange data with neighboring nodes. Since the sensory data and computational resources are (evenly) distributed among the nodes, distributed state-estimators assign a local estimation algorithm to each node in the network. As such, nodes typically compute a local estimate of the global state. A generalized set-up that corresponds to distributed state estimation is illustrated in Figure 5.1, followed by a detailed description.

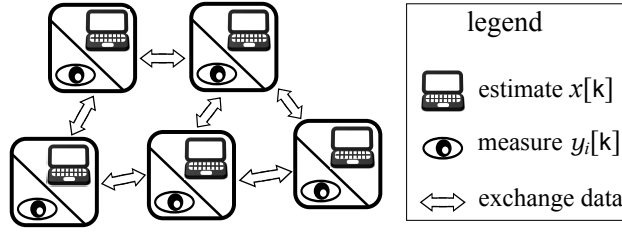


Figure 5.1: A typical set-up for distributed estimation, where each node computes a local estimate of the global state. The network topology causes that computational and sensing resources are spatially distributed.

Communication The network consists of N sensor nodes and is modeled as an undirected, connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{C})$, see Section 1.5. Each node $i \in \mathcal{N}$ is identified with a unique number, where $\mathcal{N} := \mathbb{Z}_{[1, N]}$. Further, the set $\mathcal{N}_{i(q)} \subseteq \mathcal{N}$ is defined as a collection of all nodes $j \in \mathcal{N}$ that have a minimal graph-distance of $q \in \mathbb{Z}_{>0}$ to node i , see Definition 1.5.4. Hence, *neighboring* nodes $j \in \mathcal{N}_{i(1)}$ that exchange data with node i have a graph-distance of 1. Note that a connected graph implies $\cup_{q \in \mathbb{Z}_+} \{\mathcal{N}_{i(q)}\} = \mathcal{N}$, for all $i \in \mathcal{N}$.

Process A node $i \in \mathcal{N}$ observes the perturbed, dynamical process of (1.1), for some constant sampling time $\tau_s \in \mathbb{R}_{>0}$ and $u(t) = 0$, i.e.,

$$x[k] = Ax[k-1] + w[k-1], \quad (5.1a)$$

$$y_i[k] = C_i x[k] + v_i[k]. \quad (5.1b)$$

The state vector is denoted as $x \in \mathbb{R}^n$, whereas $y_i \in \mathbb{R}^{m_i}$ corresponds to the measurement of node i . Both the process-noise $w \in \mathbb{R}^n$ and measurement-noise $v_i \in \mathbb{R}^{m_i}$ are characterized by Gaussian PDFs, for some $Q \in \mathbb{R}^{n \times n}$ and $V_i \in \mathbb{R}^{m_i \times m_i}$, i.e.,

$$p(w[k]) := G(w[k], 0, Q) \quad \text{and} \quad p(v_i[k]) := G(v_i[k], 0, V_i).$$

The goal of the sensor network is to compute a local estimate $x_i \in \mathbb{R}^n$ of the global state x in each node i . Since the process model is linear and both noises are Gaussian distributed, it is appropriate to assume that the random variable $x_i[k]$ is Gaussian distributed as well, i.e., $p(x_i[k]) := G(x_i[k], \hat{x}_i[k], P_i[k])$ for some mean $\hat{x}_i[k] \in \mathbb{R}^n$ and error-covariance $P_i[k] \in \mathbb{R}^{n \times n}$.

The challenges for state estimation in sensor networks are twofold. Not only should the employed estimation strategy exploit the large amount of local measurements but, in addition, it is desired that the corresponding implementation is robust to the evidential changes that will be present in such networks. Distributed state estimation has the potential to handle these challenges, though there are still some drawbacks on methods currently available.

These drawbacks mainly arise from the fact that their starting point is optimizing the local estimation results per node *individually*. As such, strict requirements are necessary to guarantee that each node receives all the information as specified during the design. Since these requirements are likely to be violated by system changes, an alternative approach is to solve the estimation problem from a network point of view, rather than from individual nodes. In that case, the set-up depicted in Figure 5.1 is regarded as a *network of state estimating nodes* that, due to the interaction between neighboring nodes, will emerge with a particular behavior. Hence, an alternative approach for improving robustness to system changes is formulating an appropriate behavior that should emerge from the local estimation algorithms performed in each node. To that extent, this chapter proposes a cooperative behavior, i.e., the state estimating nodes have the will to assist other nodes.

Let us further explain the objective of this chapter, which is to design a distributed state-estimator that emerges with a cooperative behavior. A description of such behavior for state estimation was given as a solution of Problem 1.3.3: *A distributed state-estimator that attains the global covariance property*. Therein, *global covariance* states that the error-covariance of each node in the network is a combination of all error-covariances found across its nodes, see Property 1.3.1. However, a corresponding characterization of this property is yet to be given. To that extent, recall that each node computes a local estimate of x_i according to the Gaussian PDF $p(x_i[k]) = G(x_i[k], \hat{x}_i[k], P_i[k])$. Further, let us introduce $\chi_{i,j} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ as a mapping function for covariance matrices, such that the resulting mapped covariance matrix of $\chi_{i,j}(\cdot)$ has bounded eigenvalues if the original covariance matrix is bounded as well. Then, a distributed state-estimator that attains the global covariance property satisfies the following inequality,

$$P_i[k] \preceq \chi_{i,j}(P_j[k-c]), \quad \forall i, j \in \mathcal{N} \text{ and } c \in \mathbb{Z}_{\geq 0}. \quad (5.2)$$

Basically, *global covariance* implies that the local estimation results of node i are improved by previously obtained estimation results from another node j . Hence, any other node j assists node i in computing a local estimate of x_i .

The above characterization of cooperative estimation is applied for observing the linear process model (5.1), in the next section on *cooperative Kalman filters* (CKFs). A justification of its design is given first, along with the results of currently available solutions, followed by a derivation of the corresponding local estimation algorithm performed in each node. After that, an asymptotic analysis gives the conditions for an asymptotically bounded $P_i[k]$ and further proves the global covariance property. This property is also demonstrated in an illustrative example of tracking the leading vehicle in a scalable platoon of “intelligent” vehicles. Additionally, an extension to non-linear processes is studied, which includes the possibility to have different types of estimators among the nodes for processing their local measurement.

As such, nodes with different computational complexities are supported in the same network. This improves the applicability of the proposed estimation approach in real-life sensor networks. An application example of tracking shockwaves on a highway demonstrates that such a *heterogenous, cooperative state-estimator* is feasible.

5.2 Cooperative Kalman filters

Before a justification of the proposed CKFs is given, let us start with a summary of the overall solution. The idea of CKFs stems from the three social laws that are followed by ants, see Section 1.3.2. These laws emerge into a cooperative behavior of ants and enables the colony to cope with a wide variety of unknown changes in their environment. Therefore, the local estimation algorithm of a node in CKFs is based on the same three social laws, i.e.,

- Division of labour is met by performing a local Kalman filter (KF) per node i for computing $\hat{x}_i[k]$ and $P_i[k]$ based on $y_i[k]$;
- Communication between individuals is defined by exchanging the local estimate $x_i[k]$ with the neighboring nodes $j \in \mathcal{N}_{i(1)}$;
- Cooperation implies that nodes fuse their local estimates x_i with the ones received from neighboring nodes.

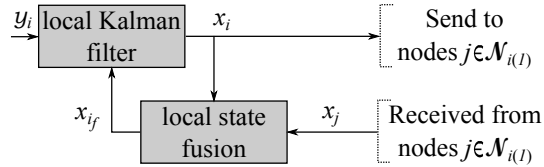


Figure 5.2: Schematic set-up of the local estimation algorithm performed by a single node i in a network of cooperative Kalman filters.

Figure 5.2 schematically depicts the local estimation algorithm, for a node i , that is in line with the above three laws of CKFs. In this set-up, the measurement $y_i[k]$ is used by a local KF to determine the local estimate $x_i[k]$, i.e., $p(x_i[k]) = G(x_i[k], \hat{x}_i[k], P_i[k])$. Nodes then exchange the result of their local KF with each other, due to which node i receives the Gaussian PDF of $x_j[k]$, for all $j \in \mathcal{N}_{i(1)}$. These received estimates are then merged with $x_i[k]$ in a state fusion method, e.g., *ellipsoidal intersection* of Chapter 4. The resulting fused estimate is denoted with $x_{i_f}[k]$ and follows $p(x_{i_f}[k]) := G(x_{i_f}[k], \hat{x}_{i_f}[k], P_{i_f}[k])$, for some $\hat{x}_{i_f}[k] \in \mathbb{R}^n$ and $P_{i_f}[k] \in \mathbb{R}^{n \times n}$.

5.2.1 Justification of the design

A justification of CKFs is given according to the three social laws of ant colonies. In addition, the currently available solutions on distributed Kalman filtering will be taken into account at each social law. For clarity of expression, the time index k is omitted throughout this section.

Division of labour: The burden of merging N local measurements is divided among the nodes by pre-processing y_i at node i . For state estimation it makes sense that such pre-processing involves the computation of a local estimate x_i by a local KF. In fact, this solution is repeatedly used in non-centralized Kalman filtering for the last 30 years. See, for example, the proposed methods of (Hassan et al., 1978; Speyer, 1979; Hashmipour et al., 1988; Durant-Whyte et al., 1990; Roy et al., 1991; Mutambara and H.F., 2000; Khan and Moura, 2007; Olfati-Saber, 2009). Some methods also merge neighboring measurements in the local KF of node i to improve its estimation results even further, i.e., y_j for all $j \in \mathcal{N}_{i(1)}$. However, note that this solution is not in line with a division of labour, as the each local measurement y_i is similarly processed by multiple nodes. An additional drawback of exchanging measurement is related to communication.

Communication between individuals: The variables that a node i can share with neighboring nodes are y_i and x_i . Communication implies that the receiving nodes must be able to understand the exchanged variables. In case node i exchanges measurements, then this means that also $y_i = C_i x + v_i$ of the local process model in (5.1) must be shared with node $j \in \mathcal{N}_{i(1)}$. Even though exchanging C_i and V_i apart from y_i might be feasible, it imposes strict requirements on the interaction of nodes, especially for extensions where the local measurement is modeled with a nonlinear function. Moreover, these strict requirements are likely to be violated by system changes, due to which one cannot guarantee that other nodes $j \neq i$ have the necessary information available for processing y_i . The alternative of exchanging local estimates is still possible. In particular, as there exist several methods that can merge x_i and x_j without requiring any additional information other than $p(x_i)$ and $p(x_j)$, i.e., not even their correlation. Hence, exchanging local estimates suits an approach for communication between individuals and thus for cooperation.

Cooperation: A cooperative behavior of nodes, for the purpose of state estimation, was formulated as a distributed state-estimator that satisfies the global covariance property. To that extent, nodes exchange local estimates, as exchanging y_i will not result in a fulfilment of this property, see, e.g., (Durant-Whyte et al., 1990). Local estimates can be combined by synchronization and fusion strategies. Examples of synchronization strategies within distributed estimation are found in (Xiao et al., 2005; Olfati-Saber, 2009). The idea emerged from synchronizing the different internal clocks of net-

worked systems. Similarly, the local means \hat{x}_i , for all $i \in \mathcal{N}$, can also be synchronized, as they all represent an estimated value of the same state x . However, synchronization of the local means induces a synchronization of the local estimation errors as well, rather than a reduction. For example, nodes with accurate sensor readings will still obtain erroneous estimates as a result of synchronization methods, when other nodes in the network have inaccurate sensor readings. Or worse, if one not in the network has unstable estimation results, then other nodes will mimic this behavior when synchronization is employed. The alternative approach of fusion takes the local error-covariances P_i and P_j explicitly into account when merging x_i and x_j . As such, fusion methods can neglect the inaccurate parts of x_i and x_j and thereby, ensure that the uncertainty after fusion is reduced. Figure 5.3 illustrates such a reduction by comparing the results of fusion and synchronization. Based on this analysis, the proposed CKFs approach performs a state fusion method to merge the received estimates from neighboring nodes with the local estimate.

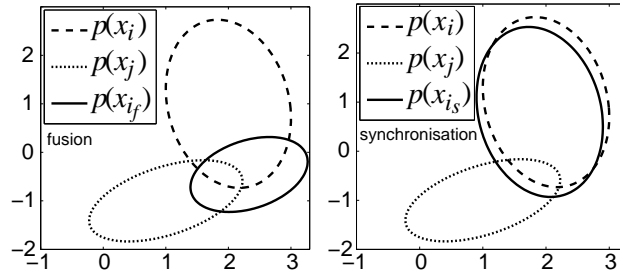


Figure 5.3: A comparison of fusion and synchronization for two Gaussian estimates x_i and x_j . The PDF after fusion via ellipsoidal intersection of Chapter 4 is denoted as $p(x_{i_f}) = G(x_{i_f}, \hat{x}_{i_f}, P_{i_f})$. The PDF after synchronization via the common averaging-formula $P_{i_s} = P_i$ and $\hat{x}_{i_s} = \omega \hat{x}_i + (1 - \omega) \hat{x}_j$ with $\omega = 0.1$ is denoted as $p(x_{i_s}) = G(x_{i_s}, \hat{x}_{i_s}, P_{i_s})$. Note that PDFs are represented as ellipsoidal sub-level-set, i.e., $G(x, \mu, \Sigma) \rightarrow \mathcal{E}_{\mu, \Sigma}$. A graphical characterization of such a sub-level-set is found in Figure 1.9, though let us point out that a larger covariance Σ implies a larger area-size of $\mathcal{E}_{\mu, \Sigma}$.

5.2.2 Implementation of CKFs

Before the CKF set-up of Figure 5.2 is summarized into an algorithm, a choice on the employed state fusion method must be made. As mentioned in the previous section, fusion of two arbitrary estimates x_i and x_j into a fused estimate x_{i_f} should be based on $p(x_i)$ and $p(x_j)$ only. This means that the correlation of x_i and x_j , i.e., $cov(x_i, x_j)$, is not available. Otherwise, the

sensor network should keep track of shared estimates between any two nodes, which imposes the impractical requirement of global communication.

A detailed overview of fusion methods that deal with unknown correlations is presented in Section 4.3. It was mentioned that a popular fusion method is *covariance intersection* (Julier and Uhlmann, 1997b), for which some extensions are found in (Hanebeck et al., 2001; Niehsen, 2002; Franken and Hupper, 2005; Julier, 2009; Wang and Li, 2009). However, a fundamental issue with the covariance intersection approach is that both \hat{x}_{i_f} and P_{i_f} are parameterized before correlation is treated. As such, any effects of the unknown correlation are mapped on the restricted parametrization of x_{i_f} , which introduces conservatism in the fusion results. A less conservative fusion method is *ellipsoidal intersection* (EI) of Chapter 4. EI proposes a parametrization of the prior estimates x_i and x_j to establish an explicit characterization of correlations, before deriving the fusion formulas. As a result, correlation can be maximized, due to which EI is able to merge the estimate x_i with an independent part of x_j into a single fused estimate. The parametrization of EI was introduced via a *mutual covariance* $\Gamma \in \mathbb{R}^{n \times n}$ and a *mutual mean* $\gamma \in \mathbb{R}^n$. An algebraic expression of these variables, for $S_i, D_i, S_j, D_j \in \mathbb{R}^{n \times n}$ as specified in (4.19a) and $\varsigma \in \mathbb{R}_{>0}$ conform to (4.23c), are given in the following fusion formulas of EI, which compute x_{i_f} as a fused estimate of the two prior estimates x_i and x_j :

$$P_{i_f} = (P_i^{-1} + P_j^{-1} - \Gamma^{-1})^{-1}; \quad (5.3a)$$

$$\hat{x}_{i_f} = P_{i_f}(P_i^{-1}\hat{x}_i + P_j^{-1}\hat{x}_j - \Gamma^{-1}\gamma);$$

$$\text{where } D_\Gamma = \text{diag}_{q \in \mathbb{Z}_{[1,n]}}(\max\{1, \{D_j\}_{qq}\}); \quad (5.3b)$$

$$\Gamma = S_i D_i^{\frac{1}{2}} S_j D_\Gamma S_j^{-1} D_i^{\frac{1}{2}} S_i^{-1}; \quad (5.3c)$$

$$\begin{aligned} \gamma &= (P_i^{-1} + P_j^{-1} - 2\Gamma^{-1} + 2\varsigma I_n)^{-1} \times \\ &\quad ((P_j^{-1} - \Gamma^{-1} + \varsigma I_n)\hat{x}_i + (P_i^{-1} - \Gamma^{-1} + \varsigma I_n)\hat{x}_j). \end{aligned} \quad (5.3d)$$

An important property of EI, that will be instrumental in the asymptotic analysis of CKFs, is a reduction of the uncertainty after fusion. A proof of this property is found in Chapter 4.

Proposition 5.2.1 *Let the two prior estimates x_i and x_j be given. Then $P_{i_f} \preceq P_i$ and $P_{i_f} \preceq P_j$ hold for their fused estimate x_{i_f} according to EI.*

Now, all aspects of the CKFs approach have been addressed and the local estimation algorithm performed by a node i is presented, next. To that extent, note that Figure 5.2 illustrates a schematic implementation of the CKFs algorithm, i.e., a local KF followed by a local state fusion according to EI. Fusion of $x_i[k]$ with multiple estimates is commonly conducted recursively. This means that $x_i[k]$ is merged with the first received $x_j[k]$, after which their fusion result is further merged with the local estimate that is received

next, and so on. For clarity of exposition, let the initial local estimate at the k -th sample instant be denoted as $x_{i(0)} := x_i[k]$. Then this recursive behavior implies that $x_{i(l)}$ is defined as the fusion result of $x_{i(l-1)}$ and the l -th received local estimate $x_j[k]$, which will be denoted as $x_{j(l)}$, for all $l \in \mathbb{Z}_{[1,L]}$ and $L := \#\mathcal{N}_{i(1)}$. The final estimate, obtained after fusion of all neighboring estimates $x_j[k]$ with $x_i[k]$, is thus $x_{i_f}[k] := x_{i(L)}$. Hence, a summary of the CKF algorithm at node i , where the local KF is expressed in the information form of (1.5), yields

The cooperative Kalman filter at node i

Step 1: Kalman filter (information form of (1.5))

$$P_i[k^-] = AP_{i_f}[k-1]A^\top + Q;$$

$$P_i[k] = (P_i^{-1}[k^-] + C_i^\top V_i^{-1} C_i)^{-1};$$

$$\hat{x}_i[k] = P_i[k](P_i^{-1}[k^-](A\hat{x}_{i_f}[k-1]) + C_i^\top R_i^{-1} y_i[k]);$$

Step 2: state fusion (EI)

$$\hat{x}_{i(0)} = \hat{x}_i[k], \quad P_{i(0)} = P_i[k];$$

for $l = 1, \dots, L$, do:

$$\hat{x}_{j(l)} = \hat{x}_j[k], \quad P_{j(l)} = P_j[k], \quad j(l) \in \mathcal{N}_{i(1)};$$

$$\Gamma_{(l)} = \text{MutualCovariance}(P_{i(l-1)}, P_{j(l)}), \text{ i.e.,} \quad (5.3c);$$

$$\gamma_{(l)} = \text{MutualMean}(P_{i(l-1)}, P_{j(l)}, \Gamma_{(l)}, \hat{x}_{i(l-1)}, \hat{x}_{j(l)}), \text{ i.e.,} \quad (??);$$

$$P_{i(l)} = (P_{i(l-1)}^{-1} + P_{j(l)}^{-1} - \Gamma_{(l)}^{-1})^{-1};$$

$$\hat{x}_{i(l)} = P_{i(l)}(P_{i(l-1)}^{-1}\hat{x}_{i(l-1)} + P_{j(l)}^{-1}\hat{x}_{j(l)} - \Gamma_{(l)}^{-1}\gamma_{(l)});$$

end

$$\hat{x}_{i_f}[k] = \hat{x}_{i(L)}, \quad P_{i_f}[k] = P_{i(L)}.$$

The notation k^- is used to emphasize predicted estimates from updated ones. Next, the emergent behavior CKFs is studied in asymptotic analysis.

5.2.3 Asymptotic analysis

This section proves that a connected¹ network of state estimating nodes fulfills the *global covariance* property, when the CKFs approach is employed. Moreover, realistic conditions are derived under which $P_{i_f}[k]$ is asymptotically bounded by another covariance matrix. To that extent, let us introduce the covariance matrix $\Sigma_i[k] \in \mathbb{R}^{n \times n}$, for each node $i \in \mathcal{N}$, with a Riccati difference equation (RDE) and $\Sigma_i[-1] = P_{i_f}[-1]$. In line with other update formulas, let us express the RDE for updating Σ_i in an equivalent information

¹Connected means that a graph path between any two nodes $i, j \in \mathcal{N}$ exists.

form, i.e.,

$$\Sigma_i[k] = ((A\Sigma_i[k-1]A^\top + Q)^{-1} + C_i^\top V_i^{-1} C_i)^{-1}, \quad \forall k \in \mathbb{Z}_+, \forall i \in \mathcal{N}. \quad (5.4)$$

The asymptotic properties of the RDE were presented in (Chan et al., 1984).

Proposition 5.2.2 *Let (A, C_i) be an observable pair and let $\lambda_q(A_i) \leq 1$ hold, for all $q \in \mathbb{Z}_{[1,n]}$. Then, $\Sigma_i[\infty] := \lim_{k \rightarrow \infty} \Sigma_i[k]$ is a stabilizing solution of the RDE in (5.4) that is unique and independent of $\Sigma_i[-1]$.*

Definition 5.2.3 The set $\mathcal{N}_{\text{RDE}} \subseteq \mathcal{N}$ is defined as the collection of all nodes $i \in \mathcal{N}$ that satisfy the hypothesis of Proposition 5.2.2.

The theoretical analysis of the CKF starts with two properties of the error-covariances P_{i_f} and P_i , for which $\mathcal{N}_{i(0,1)} := \{i\} \cup \mathcal{N}_{i(1)}$. The first property considers a node $i \in \mathcal{N}$ and its direct neighboring nodes.

Lemma 5.2.4 *Let each node i employ the CKF. Then, $P_{i_f}[k] \preceq P_j[k]$ and $P_i[k+1] \preceq AP_j[k]A^\top + Q$ hold for all $j \in \mathcal{N}_{i(0,1)}$ and $k \in \mathbb{Z}_+$.*

Lemma 5.2.4 is proven in Appendix E.1. Notice that this lemma indicates that the received error-covariances of node i are bounded by their neighboring error-covariances as well. If, for example, node $j \in \mathcal{N}_{i(2)}$ is a direct neighbor of a certain node $h \in \mathcal{N}_{i(1)}$, i.e., $j \in \mathcal{N}_{h(1)}$, then Lemma 5.2.4 establishes that $P_h[k] \preceq AP_j[k-1]A^\top + Q$. Moreover, since node h is a direct neighbor of node i , the same lemma also gives that $P_{i_f}[k] \preceq P_h[k]$. Hence, although node $j \in \mathcal{N}_{i(2)}$ is not a direct neighbor of node i , there exists a bound on $P_{i_f}[k]$ that depends on $P_j[k-1]$. In the second property, this reasoning is extended to show that $P_{i_f}[k]$ is bounded by a prediction of $P_j[k-c]$ to the k -th sample instant for any node $j \in \mathcal{N}_{i(c+1)}$. Therein, $c \in \mathbb{Z}_{\geq 1}$ is such that the graph-distance between nodes i and j is equal to $c+1$.

Lemma 5.2.5 *Let each node i employ the CKF. Then, it holds that $P_{i_f}[k] \preceq A^c P_j[k-c](A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top$, for all $k \in \mathbb{Z}_{\geq c}$, $j \in \mathcal{N}_{i(c+1)}$ and $c \in \mathbb{Z}_{\geq 1}$.*

Lemma 5.2.5 is proven in Appendix E.2. When this lemma is combined with Lemma 5.2.4, they prove that the error-covariance of node i , i.e., $P_{i_f}[k]$, is a combination of the error-covariances found across all nodes in the network. As such, *the CKFs approach is a distributed state-estimator that attains the global covariance property*. More precisely, $P_{i_f}[k] \preceq \chi_{i,j}(P_j[k-c])$ is satisfied for all $i, j \in \mathcal{N}$, in case the mapping function $\chi_{i,j}(\cdot)$ is characterized as follows:

$$\chi_{i,j}(P_j[k]) := \begin{cases} P_j[k] & \text{if } j \in \mathcal{N}_{i(0,1)}, k \in \mathbb{Z}_+, \\ A^c P_j[k-c](A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top & \text{if } j \in \mathcal{N}_{i(c+1)}, k \in \mathbb{Z}_{\geq c}. \end{cases}$$

Next, let us continue with a derivation of the asymptotic bounds on $P_{i_f}[k]$, by relating $P_{i_f}[k]$ of the CKF to $\Sigma_i[k]$ of the RDE.

Lemma 5.2.6 *Let each node i employ the CKF. Then, $P_i[k] \preceq \Sigma_i[k]$ holds for all $k \in \mathbb{Z}_+$ and $P_i[-1] = \Sigma_i[-1]$, where $\Sigma_i[k]$ satisfies the RDE in (5.4).*

Lemma 5.2.6 is proven in Appendix E.3. The analysis continues by employing the above mapping function $\chi_{i,j}(\cdot)$ to introduce $\bar{\Sigma}_{i,j}[k] := \chi_{i,j}(\Sigma_j[k-c])$, for all $j \in \mathcal{N}$ and $c \in \mathbb{Z}_{\geq 1}$, i.e.,

$$\bar{\Sigma}_{i,j}[k] = \begin{cases} \Sigma_j[k] & \text{if } j \in \mathcal{N}_{i(0,1)}, k \in \mathbb{Z}_+, \\ A^c \Sigma_j[k-c] (A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top & \text{if } j \in \mathcal{N}_{i(c+1)}, k \in \mathbb{Z}_{\geq c}. \end{cases}$$

The asymptotic value $\bar{\Sigma}_{i,j}[\infty]$ follows the above formula if $\Sigma_j[\infty]$ exists, i.e., for every node $j \in \mathcal{N}_{\text{RDE}}$. Now, the main result of this asymptotic analysis is presented, i.e., $P_{i_f}[\infty] := \lim_{k \rightarrow \infty} P_{i_f}[k]$ of the CKFs approach is bounded under realistic conditions.

Theorem 5.2.7 *Let each node i employ the CKF and let $\mathcal{N}_{\text{RDE}} \neq \emptyset$. Then, $P_{i_f}[\infty]$ exists and satisfies $P_{i_f}[\infty] \preceq \bar{\Sigma}_{i,j}[\infty]$, for all $j \in \mathcal{N}_{\text{RDE}}$.*

The proof of Theorem 5.2.7 is presented in Appendix E.4. This theorem further implies that the asymptotic bounds on $P_{i_f}[\infty]$ exists for all the nodes $i \in \mathcal{N}$, given that \mathcal{N}_{RDE} is non-empty, i.e., there exists at least one node that has a *stable* local estimation result. Similar aspects are shown in an illustrative example, next.

5.3 Illustrative example

In this example, the estimation error of the proposed CKFs is compared to alternative distributed Kalman filtering algorithms. The benchmark application is a platoon of four “intelligent” vehicles equipped with a cooperative adaptive cruise controller, see, for example, (van Arem et al., 2006). The main reason for choosing this application is because each vehicle should obtain an estimate of the longitudinal position and speed of the leading vehicle in that platoon. As such, the state x is defined as this longitudinal position and speed of vehicle 1. The example considers estimation and does not present any control, due to which the following set-up is considered:

Communication Each vehicle communicates with its front and rear vehicle.

Hence, the graph-model $\mathcal{G}(\mathcal{V}, \mathcal{C})$ of this network is characterized by $\mathcal{V} := \{v_1, v_2, v_3, v_4\}$ and $\mathcal{C} := \{(v_i, v_j) \mid |i - j| = 1\}$, for all $i, j \in \mathbb{Z}_{[1,4]}$.

Process The process model is the double integrator with $\tau_s = 0.1$, i.e.,

$$x[k] = \begin{pmatrix} 1 & 0.1 \\ 0 & 1 \end{pmatrix} x[k-1] + \begin{pmatrix} 0.005 \\ 0.1 \end{pmatrix} a[k-1],$$

where the acceleration is specified as $a[k] = 0.1 + 3 \sin(0.02k)$. Since acceleration is unknown to the estimator, the process-noise is modeled as

$w[k] = \begin{pmatrix} 0.005 \\ 0.1 \end{pmatrix} a[k]$ with a corresponding covariance $Q = 10^{-3} \begin{pmatrix} 0.1 & 2.5 \\ 2.5 & 50 \end{pmatrix}$, while the initial state is $x[-1] = \begin{pmatrix} 10 \\ 15 \end{pmatrix}$. Further, to have a realistic scenario, each vehicle measures its own position and speed and the distance to the vehicle in front. Due to this dispersion of the sensors, only vehicles 1 and 2 have local measurements related to x , i.e.,

$$y_1[k] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x[k] + v_1[k], \quad p(v_1[k]) = G(v_1[k], 0, 0.05I_2),$$

$$y_2[k] = (1 \ 0)x[k] + v_2[k], \quad p(v_2[k]) = G(v_2[k], 0, 0.5).$$

Vehicles 3 and 4 have no sensors that depend on the state. As such, their y_3 and y_4 are modeled via $C_3 = C_4 = (0 \ 0)$ and $V_3^{-1} = V_4^{-1} = 0$.

The platoon initially consists of three vehicles. After 5 seconds the fourth vehicle enters the platoon from the rear. Each vehicle in the platoon performs a local estimation algorithm that is in line with a certain distributed Kalman filter. Three different methods are compared that start with equivalent estimates, i.e., $\hat{x}_i[-1] = \begin{pmatrix} 10 \\ 15 \end{pmatrix}$ and $P_i[-1] = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$, for all $i \in \mathbb{Z}_{[1,4]}$:

- The proposed CKFs;
- The cooperative covariance intersection KFs (CCIKFs), which is similar to the CKFs only that state fusion is performed according to covariance intersection of (Julier and Uhlmann, 1997b) instead of EI;
- The distributed consensus information filter (DCIF) presented in (Olfati-Saber, 2009). The DCIF employs a local KF in each node i on y_j , for all $j \in \mathcal{N}_{i(0,1)}$, followed by a synchronization step on the local mean $\hat{x}_i[k]$ with the neighboring means $\hat{x}_j[k]$, for all $j \in \mathcal{N}_{i(1)}$.

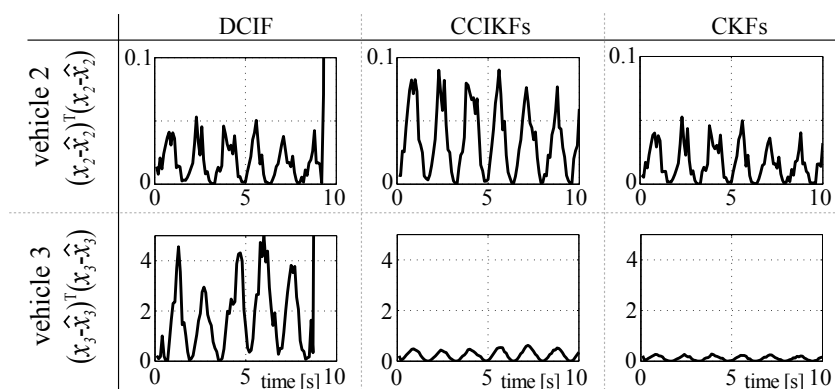


Figure 5.4: A comparison of the state estimation error of vehicles 2 and 3 for the distributed Kalman filtering methods DCIF, CCIKFs and CKFs. After $t = 5$ seconds a fourth vehicle joins the platoon of three vehicles.

The estimation-error of vehicles 2 and 3 is depicted in Figure 5.4. Notice that CKFs clearly outperform CCIKFs in both vehicles, as the error with CCIKFs is twice as large. This is mainly a result of the employed fusion method covariance intersection in CCIKFs, which computes a conservative fusion result that is a convex combination of the local and neighboring estimates. More precisely, since vehicles 3 and 4 have no measurements related to the state, the estimation results computed by their local KF has large errors. The convex combination procedure of covariance intersection then causes that these erroneous estimates affect the estimation results of vehicles 1 and 2 as well. Instead, CKFs employs EI as a state fusion method, due to which the global covariance property is attained. This means that the erroneous estimates x_3 and x_4 do *not* affect the results of vehicles 1 and 2 for CKFs, as EI detects their “inaccuracies” via P_3 and P_4 . Further, note that the estimation error of x_2 is lower than the error of x_3 for both the CKFs and CCIKFs. This can be explained by the communication topology, which introduces an additional communication step for vehicle 3 compared to vehicle 2 upon receiving information from y_1 . Let us continue with analyzing the results of the third estimator. Figure 5.4 indicates that the DCIF yields an unstable estimate when vehicle 4 is added to the platoon. To explain such behavior, let us point out that a requirement of the DCIF is that the local state estimate x_i in each vehicle i should be *observable* based on its local and neighboring measurements $\cup_{j \in \mathcal{N}_{i(1)}} \{y_j\}$. This requirement is not satisfied in the considered set-up, as vehicle 4 does not receive any measurement related to x . Hence, the estimation error in the fourth vehicle diverges, which is then emulated by the other vehicles in the platoon due to the synchronization approach of the DCIF. Therefore, the CKFs outperforms both the CCIKFs and the DCIF in this illustrative example.

A more detailed comparison of distributed Kalman filtering algorithms is studied in Chapter 6. This chapter continues with nonlinear processes.

5.4 Nonlinear state estimation

Currently existing methods of distributed state estimation perform the same estimation algorithm in each node locally. For example, in CKFs each node employs a local KF before fusion. However, note that the proposed cooperative approach is not restricted to a network where nodes perform the KF for processing their local measurement. The only requirement, imposed by the fusion method EI, is that the local estimate x_i is characterized by a Gaussian PDF, for all $i \in \mathcal{N}$. Hence, it is irrelevant whether this Gaussian was computed by a KF, or by nonlinear state estimation methods, such as the extended KF and the unscented KF. The set-up of a node’s local estimation algorithm that corresponds to this generalization of CKFs, i.e., where the local measurement is processed by any local state estimator that has Gaussian estimation results, is depicted in Figure 5.5.

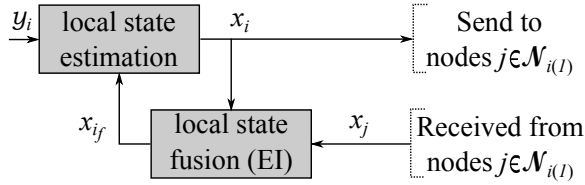


Figure 5.5: Schematic set-up of the local estimation algorithm performed by a single node i in a network of cooperative state-estimators.

Therefore, this section studies the feasibility of cooperative extended KFs and cooperative unscented KFs. Therein, each node employs the CKF of Section 5.2 and replaces the part that corresponds to the local KF with either an extended or unscented KF. In addition, a mixture of local state-estimators among the nodes is investigated. In those types of distributed estimators, some nodes perform the extended KF for processing their local measurement, while others employ the unscented KF. This *heterogeneous distributed state estimator* not only allows different computational requirements per node in the network. Also, new nodes that employ a different (novel) estimator for computing $x_i[k]$ and/or processing $y_i[k]$ can be added to an existing network during operation, without a re-design of the active nodes in the network. A generalized description of cooperative state estimation is given next, after rewriting the model of (5.1) to accommodate nonlinear processes.

Process The process model, for some sampling time $\tau_s \in \mathbb{R}_{>0}$, yields

$$x[k] = f(x[k-1], w[k-1]), \quad (5.5a)$$

$$y_i[k] = h_i(x[k]) + v_i[k]. \quad (5.5b)$$

The function $f : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ describes the (non)linear state dynamics, while $h_i : \mathbb{R}^n \rightarrow \mathbb{R}^{m_i}$ defines a (non)linear mapping of x to y_i . All other aspects of the system description, i.e., communication and Gaussian PDFs for $w[k-1]$ and $v_i[k]$, are similar as to Section 5.1.

5.4.1 Cooperative state estimation

In *cooperative* state estimation all the nodes perform a generalization of the CKF algorithm, as it is illustrated in the schematic set-up of Figure 5.5. This generalization implies that $\hat{x}_i[k]$ and $P_i[k]$ are calculated via either a local KF, an extended KF or an unscented KF. A brief account on the latter two state estimation methods is given first, before continuing with the case-study of tracking shockwave on a highway.

Extended Kalman filter

Similar as to the KF of Section 1.2, the extended KF first performs a prediction of x followed by an update. The predicted mean $\hat{x}_i[k^-]$ is computed via the nonlinear process model. However, to predict the error-covariance $P_i[k^-]$, nonlinear dynamics are linearized around the previously estimated working point $x[k-1] = \hat{x}_i[k-1]$ and $w[k-1] = 0$. This linearization is characterized by Jacobian matrices of f and h_i in (5.5), i.e., $F_i[k] := \nabla_x f(\hat{x}_i[k-1], 0)$, $E_i[k] := \nabla_w f(\hat{x}_i[k-1], 0)$ and $H_i[k] := \nabla_x g_i(\hat{x}_i[k-1])$. To summarize, the extended KF employs the following set of equations to calculate $\hat{x}_i[k]$ and $P_i[k]$:

$$\begin{aligned}\hat{x}_i[k^-] &= f(\hat{x}_{i_f}[k-1], 0), \\ P_i[k^-] &= F_i[k]P_{i_f}[k-1]F_i^\top[k] + E_i[k]QE_i^\top[k], \\ K_i[k] &= P_i[k^-]H_i^\top[k] (H_i[k]P_i[k^-]H_i^\top[k] + V_i)^{-1}, \\ \hat{x}_i[k] &= \hat{x}_i[k^-] + K_i[k] (y_i[k] - h_i(\hat{x}_i[k])), \\ P_i[k] &= (I - K_i[k]H_i[k]) P_i[k^-].\end{aligned}\tag{5.6}$$

Although the extended KF enjoys low computational power, its estimation error depends on the support to linearize the process model of (5.5). When estimation results of the extended KF are not satisfactory, one can employ an unscented KF.

Unscented Kalman filter

The unscented KF calculates $\hat{x}_i[k]$ and $P_i[k]$ by applying the nonlinear model of (5.5) on various values of x and w at $k-1$, also known as *sigma-values*. These values are selected from an augmented vector $\mu := \begin{pmatrix} x \\ w \end{pmatrix}$ at $k-1$. The corresponding mean $\hat{\mu}_i \in \mathbb{R}^{2n}$ and covariance $U_i \in \mathbb{R}^{2n \times 2n}$ of μ at node i follow from $p(x_{i_f}[k-1])$ and $p(w[k-1])$, i.e.,

$$\hat{\mu}_i[k-1] = \begin{pmatrix} \hat{x}_{i_f}[k-1] \\ 0 \end{pmatrix}, \quad U_i[k-1] = \begin{pmatrix} P_{i_f}[k-1] & 0 \\ 0 & Q \end{pmatrix}.$$

Based on the above variables, a total of $M := 4n + 1$ different sigma-values of $\mu[k-1]$ are selected, which are denoted as $\hat{\mu}_{i,q}[k-1] \in \mathbb{R}^{2n}$ for all $q \in \mathbb{Z}_{[1,M]}$. To that extent, let $\tilde{\mu}_{i,r} \in \mathbb{R}^{n+m}$ be defined as a scaled version of the r -th eigenvector of $U_i[k-1]$, i.e., $\tilde{\mu}_{i,r} := \sqrt{\lambda_r(U_i[k-1])} \cdot \nu_r(U_i[k-1])$ for all $r \in \mathbb{Z}_{[1,2n]}$. Then the sigma-values $\hat{\mu}_{i,q}[k-1]$, for all $q \in \mathbb{Z}_{[1,M]}$ and some $c \in \mathbb{R}_{>0}$, yields

$$\hat{\mu}_{i,q}[k-1] := \begin{cases} \hat{\mu}_i[k-1] + c\tilde{\mu}_{i,q} & \text{if } q \in \mathbb{Z}_{[1,2n]}, \\ \hat{\mu}_i[k-1] - c\tilde{\mu}_{i,q-2n} & \text{if } q \in \mathbb{Z}_{[2n+1,M-1]}, \\ \hat{\mu}_i[k-1] & \text{if } q = M. \end{cases}\tag{5.7}$$

The process model of (5.5) is performed on each of the above sigma-values to obtain predictions of $x[k]$ and $y_i[k]$, i.e.,

$$\hat{x}_{i,q}[k^-] := f(\hat{\mu}_{i,q}[k-1]) \quad \text{and} \quad \hat{y}_{i,q}[k^-] := h_i(\hat{x}_{i,q}[k^-]), \quad \forall q \in \mathbb{Z}_{[1,M]}.$$

Then, with a similar notation of $[k^-]$ and $[k]$ to denote predicted and updated estimates, respectively, and for some collection of weights $\omega_q \in \mathbb{R}_{>0}$, the unscented KF can be summarized as follows:

$$\begin{aligned}\hat{x}_i[k] &= \hat{x}_i[k^-] + S_i[k] (R_i[k] + V_i)^{-1} (y_i[k] - \hat{y}_i[k^-]), \\ P_i[k] &= P_i[k^-] - S_i[k] (R_i[k] + V_i) S_i^\top[k].\end{aligned}$$

$$\begin{aligned}\text{Where, } \hat{x}_i[k^-] &= \sum_{q=1}^M \omega_q \hat{x}_{i,q}[k^-], \quad \hat{y}_i[k^-] = \sum_{q=1}^M \omega_q \hat{y}_{i,q}[k^-], \\ P_i[k^-] &= \sum_{q=1}^M \omega_q (\hat{x}_{i,q}[k^-] - \hat{x}_i[k^-]) (\hat{x}_{i,q}[k^-] - \hat{x}_i[k^-])^\top, \\ R_i[k] &= \sum_{q=1}^M \omega_q (\hat{y}_{i,q}[k^-] - \hat{y}_i[k^-]) (\hat{y}_{i,q}[k^-] - \hat{y}_i[k^-])^\top, \\ S_i[k] &= \sum_{q=1}^M \omega_q (\hat{x}_{i,q}[k^-] - \hat{x}_i[k^-]) (\hat{y}_{i,q}[k^-] - \hat{y}_i[k^-])^\top.\end{aligned}\tag{5.8}$$

The design parameters $c \in \mathbb{R}_{>0}$ of (5.7) and $\omega_q \in \mathbb{R}_{>0}$ of (5.8) are commonly chosen as follows: $c = \sqrt{n+m+\alpha}$, $\omega_M = \frac{\alpha}{n+m+\alpha}$ and $\omega_q = \frac{1}{2(n+m+\alpha)}$, for all $q \in \mathbb{Z}_{[1, M-1]}$ and some $\alpha \in \mathbb{R}_+$.

Estimating x of nonlinear processes via an unscented KF typically reduces estimation error, when compared to an extended KF, at the cost of high computational requirements. Therefore, a trade-off must be made between accuracy and computational complexity, to decide which estimator is employed by a node i for processing the local measurement y_i (and thus computing the local estimate x_i). A benchmark application of tracking shockwaves on a highway is investigated to analyze different aspects in this trade-off.

5.4.2 Benchmark application of tracking shockwaves

The traffic shockwave is a spatial-temporal phenomenon typically emerging from high density traffic. It is characterized by an increase in vehicle density and a decrease in vehicle speed. Shockwaves “travel” along the highway upstream, i.e., in opposite direction to the traffic, and are regularly the source of a forthcoming traffic jam. The case-study is to initiate a shockwave first, after which the goal is to track this (simulated) shockwave using aggregated measurements of speed and density at certain road segments. To that extent, consider a stretch of a one-lane road that is divided into 20 segments of each $L = 0.5$ [km], as depicted in Figure 5.6.

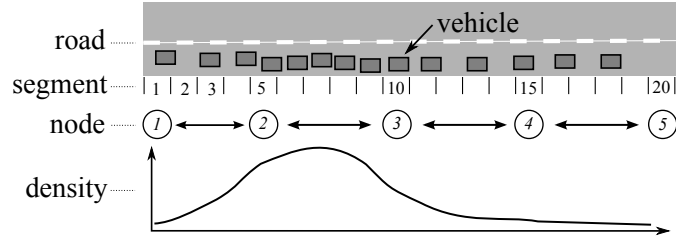


Figure 5.6: Illustration of the set-up for simulating shockwaves, i.e., the road, vehicles, segments, sensor nodes and the average vehicle density. The length of each segment has a different scale for illustration purposes.

Communication The network consists of 5 nodes that are spatially distributed along the highway. Node 1 is located at road segment 1, node 2 at segment 5, node 3 at segment 10, node 4 at segment 15 and node 5 at road segment 20. Each node exchanges data with its direct neighbors, i.e., $\mathcal{N}_1 = \{2\}$, $\mathcal{N}_2 = \{1, 3\}$, $\mathcal{N}_3 = \{2, 4\}$, $\mathcal{N}_4 = \{3, 5\}$ and $\mathcal{N}_5 = \{4\}$.

Process The process model of a traffic shockwave was presented in (Hegyi et al., 2005). Therein, $s^l \in \mathbb{R}$ and $\rho^l \in \mathbb{R}$ denote the average speed in [km/h] and density in [vehicles/km] of the l -th road segment, respectively. The model characterizes s^l and ρ^l as a nonlinear function of the same variables at segment $l - 1$ and $l + 1$, i.e.,

$$\begin{aligned} \rho^l[k+1] &= \rho^l[k] + \frac{\tau_s}{L} (\rho^{l-1}[k]s^{l-1}[k] - \rho^l[k]s^l[k]), \\ s^l[k+1] &= s^l[k] + \frac{\tau_s}{\zeta} \left(v_{free} e^{-\frac{1}{\alpha} \left(\frac{\rho^l[k]}{\rho_{crit}} \right)^\alpha} - s^l[k] \right) \\ &\quad + \frac{\tau_s}{L} s^l[k] (s^{l-1}[k] - s^n[k]) - \frac{\eta\tau_s}{\zeta L} \frac{\rho^{l+1}[k] - \rho^l[k]}{\rho^l[k] + \kappa}. \end{aligned}$$

The following parameters are used in the above process model:

- $\eta = 191$ - speed relaxation term [km²/h];
- $\zeta = 0.0039$ - time constant of the speed relaxation term [h];
- $\kappa = 254$ - speed anticipation term [vehicles/km];
- $\rho_{crit} = 33.0$ - maximum density on the lane [vehicles/km];
- $\alpha = 5.61$ - parameter for the non-compliance of drivers to speed limits;
- $v_{free} = 89.9$ - free-flow speed of the lane [km/h];
- $\tau_s = \frac{10}{3600}$ - sampling time [h].

Figure 5.7 depicts the resulting density profile of the simulated shockwave. The wave starts at road segment 20 with an increased vehicle density, after which it travels towards road segment 1 in approximately 35 minutes. The state-vector is defined as $x := (s^1 \dots s^l \rho^1 \dots \rho^l)^\top$.

Notice that the above process model requires $\rho^0[k]$, $\rho^{21}[k]$ and $s^0[k]$. As nodes do not have these values available for estimation, they are modeled as process noise with a mean that is copied from the estimated speed and density of their neighboring segment, i.e.,

$$\begin{aligned} p(\rho^0[k]) &:= G(\rho^0[k], \hat{\rho}^1[k], 40), & p(s^0[k]) &:= G(s^0[k], \hat{s}^1[k], 40), \\ p(\rho^{21}[k]) &:= G(\rho^{21}[k], \hat{\rho}^{20}[k], 40). \end{aligned}$$

The variables $\hat{\rho}^1[k]$, $\hat{s}^1[k]$ and $\hat{\rho}^{20}[k]$ in the above PDFs denote the means of $\rho^1[k]$, $\rho^{20}[k]$ and $s^1[k]$, respectively, as estimated in the corresponding node. Further, each sensor node measures the average speed and density of their corresponding segment, i.e.,

$$y_i[k] = \begin{pmatrix} \rho^{q_i}[k] \\ s^{q_i}[k] \end{pmatrix} + v_i[k] \text{ and } q_i := \begin{cases} 1 & \text{if } i = 1, \\ 5(i-1) & \text{if } i \in \mathbb{Z}_{[2,5]}. \end{cases}$$

Three configurations of cooperative state estimation are employed to recover the average density at all segments based on the available measurements. In each configuration node i performs a modified version of the CKF algorithm in Section 5.2.2 by replacing the local KF.

- The cooperative extended KF (CEKF) substitutes the local KF of the CKF algorithm with an extended KF in all nodes $i \in \mathbb{Z}_{[1,5]}$.
- The cooperative unscented KF (CUKF) substitutes the local KF of the CKF algorithm with an unscented KF in all node $i \in \mathbb{Z}_{[1,5]}$.
- The heterogenous cooperative state-estimator (HCSE) substitutes the local KF of the CKF algorithm with an extended KF in nodes 2 and 4, while nodes 1, 3 and 5 employ an unscented KF as substitute.

All nodes start with equivalent estimates, i.e., $s^l[-1] = 85$ and $\rho^l[-1] = 30$, for all $l \in \mathbb{Z}_{[1,20]}$. Figure 5.7 shows the real and estimated density profile ρ in time along the stretch of road, as it is computed by node 3. The estimation results at other nodes are similar to node 3 and therefore omitted.

Figure 5.7 indicates that the CEKF is not able to reconstruct the shockwave properly. The employed linearization of CEKF on the process model gives approximation errors that are probably too high for tracking the shockwave smoothly, as the estimated wave damps shortly after it was measured. Results of HCSE show that this improper tracking of the CEKF can be solved by replacing the extended KF at nodes 1, 3 and 5 with an unscented KF. Already after the first 5 minutes HCSE has similar results as the CUKF. Further, note all three set-ups enjoy the global covariance property, since node 3 at road segment 10 is able to track the shockwave already from the moment that the wave is firstly measured at node 5. This proves that node 3 exploits the local information of node 5, even though they do not communicate.

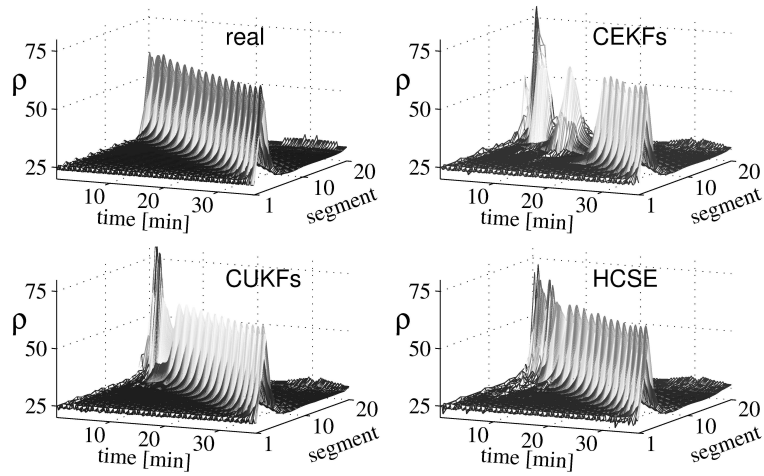


Figure 5.7: The real vehicle density of all 20 segments in time and their estimated values at node 3 according to the CEKF, CUKF and HCSE.

Beside nonlinear process models, another objective for proposing HCSE was to allow different computational requirements per node in the network. In this benchmark, nodes that employed an extended KF had an average computational time of 5 [ms] per sample instant, which increased up to 20 [ms] for nodes that performed an unscented KF algorithm. Hence, the HCSE has estimation errors that are comparable to the CUKF, while allowing a decrease in computational requirements for some nodes. This improves applicability of the proposed distributed state-estimator in sensor networks.

5.5 Conclusions

The concept of cooperative estimation was presented, suitable for sensor networks that are subject to system changes. To that extent, two novel distributed state-estimators (DSEs) were proposed, in which each node performs two steps iteratively, i.e., it runs a particular state-estimator to process its measurements, followed by state fusion method to merge the resulting local estimate with the estimates received from the neighboring nodes. The distinguishing feature of the proposed DSEs is that the estimation problem was solved from a network point of view, so that the network of state estimating nodes emerges into a *cooperative* estimator. This cooperative approach was assessed for observing linear processes first, resulting in *cooperative Kalman filters*. It was proven that this DSE attains the global covariance property and has asymptotic bounds on the error-covariance of each local estimate. The advantages of this developed DSE with respect to alternative distribu-

ted Kalman filtering algorithms was demonstrated in a benchmark example of a scalable platoon of vehicles. Additionally, nonlinear process models were investigated in an illustrative application scenario of tracking shockwaves on a highway. The analysis of this second DSE showed that having a mixture of state-estimators across the nodes could relax computational requirements of some nodes, without inducing a significant effect on estimation results in the network. As such, practical implementations of the proposed DSEs are feasible, even for sensor networks in harsh environments with many system changes.

An overview of distributed Kalman filtering

6.1	System description	6.4	Exchanging local estimates
6.2	Synchronization versus fusion	6.5	Illustrative examples
6.3	Exchanging measurements	6.6	Extended algorithms
		6.7	Conclusions

The established state-estimator for a linear process with Gaussian noise distributions is the Kalman filter. Classical implementations of its algorithm achieve optimal estimation results when measurements are acquired centrally. However, as sensors are more often deployed in a sensor network, with limitations in communication and computational resources, a centralized approach of the Kalman filter becomes infeasible. To solve this issue, distributed solutions are being developed in many different research communities. Keeping track of these solutions is an excessive task, since they are widely scattered throughout the literature. Therefore, this chapter presents a comprehensive survey on distributed Kalman filters found across these communities. Their theoretical results are compared in asymptotic analyses according to a unified description. Moreover, a critical assessment is performed in two real-life inspired case studies to fulfill the main objective: provide insight and argumentation for choosing a suitable distributed Kalman filter when deploying a sensor network.

6.1 System description

Distributed Kalman filtering (DKF) refers to a collection of estimation strategies often used in large-scale or ad-hoc sensor networks for observing linear processes. Each node of such a network is equipped with a CPU, a sensor and a two-way radio, allowing nodes to exchange data with neighboring nodes. Since the sensory data and computational resources are (evenly) distributed among the nodes, DKF solutions assign a local estimation algorithm to each node in the network. As such, nodes typically compute a local estimate of the global state. A generalized set-up that corresponds to distributed state estimation is illustrated in Figure 6.1, followed by a detailed description.

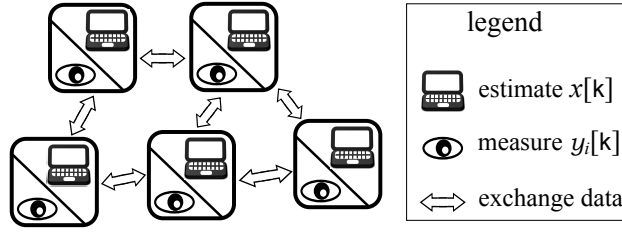


Figure 6.1: A typical set-up for distributed Kalman filtering, where each node computes a local estimate of the global state. The network topology causes that computational and sensing resources are spatially distributed.

Communication The network consists of N sensor nodes and is modeled as an undirected, connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{C})$, see Section 1.5. Each node $i \in \mathcal{N}$ is identified with a unique number, where $\mathcal{N} := \mathbb{Z}_{[1, N]}$. Further, the set $\mathcal{N}_{i(q)} \subseteq \mathcal{N}$ is defined as a collection of all nodes $j \in \mathcal{N}$ that have a minimal graph-distance of $q \in \mathbb{Z}_{>0}$ to node i , see Definition 1.5.4. Hence, *neighboring* nodes $j \in \mathcal{N}_{i(1)}$ that exchange data with node i have a graph-distance of 1. Note that a connected graph implies $\cup_{q \in \mathbb{Z}_+} \{\mathcal{N}_{i(q)}\} = \mathcal{N}$, for all $i \in \mathcal{N}$.

Process A node $i \in \mathcal{N}$ observes the perturbed, dynamical process of (1.1), for some constant sampling time $\tau_s \in \mathbb{R}_{>0}$ and $u(t) = 0$, i.e.,

$$x[\mathbf{k}] = Ax[\mathbf{k}-1] + w[\mathbf{k}-1], \quad (6.1a)$$

$$y_i[\mathbf{k}] = C_i x[\mathbf{k}] + v_i[\mathbf{k}]. \quad (6.1b)$$

The state vector is denoted as $x \in \mathbb{R}^n$, whereas $y_i \in \mathbb{R}^{m_i}$ corresponds to the measurement of node i . Both the process-noise $w \in \mathbb{R}^n$ and measurement-noise $v_i \in \mathbb{R}^{m_i}$ are characterized by Gaussian PDFs, for some $Q \in \mathbb{R}^{n \times n}$ and $V_i \in \mathbb{R}^{m_i \times m_i}$, i.e.,

$$p(w[\mathbf{k}]) := G(w[\mathbf{k}], 0, Q) \quad \text{and} \quad p(v_i[\mathbf{k}]) := G(v_i[\mathbf{k}], 0, V_i).$$

The goal of the sensor network is to compute a local estimate $x_i \in \mathbb{R}^n$ of the global state x in each node i . Since the process model is linear and both noises are Gaussian distributed, it is appropriate to assume that the random variable $x_i[\mathbf{k}]$ is Gaussian distributed as well, i.e., $p(x_i[\mathbf{k}]) := G(x_i[\mathbf{k}], \hat{x}_i[\mathbf{k}], P_i[\mathbf{k}])$ for some mean $\hat{x}_i[\mathbf{k}] \in \mathbb{R}^n$ and error-covariance $P_i[\mathbf{k}] \in \mathbb{R}^{n \times n}$.

The main challenge for state estimation in sensor networks is to exploit the large amount of local measurements. In addition, some applications desire that the corresponding implementation is robust to changes present in the network. Numerous solutions on DKF have been (are still being) developed to address these challenges, mostly for observing linear processes, such as

the one presented in (1.1). Their development is based on some early non-centralized KFs for a network that supports global communication, see, e.g., (Speyer, 1979; Hassan et al., 1978; Hashmipour et al., 1988; Roy et al., 1991). The glimpse of DKF solutions presented in Section 1.3.2 showed that their popularity in various research communities led to a wide variety of solutions scattered throughout the literature. Several overviews on DKF are presented in (Felter, 1990; Hespanha et al., 2007; Sijts et al., 2008; Ribeiro et al., 2010; Garin and Schenato, 2011). However, they could only address a selective area of DKF, due to the growing amount of developed methodologies in various research communities. Moreover, each community complies on its own assumptions and arguments that are not clearly stated for outsiders.

Therefore, this chapter presents a solution to Problem 1.3.4: *A comprehensive overview on distributed Kalman filters that assists in choosing a suitable solution for a sensor network application.* The presentation distinguishes four types of DKF strategies. Each strategy is a combination of the following two design options:

- Exchanging local *measurements* or local *estimates*;
- *Synchronize* or *fuse* the exchanged estimation variables;

The overview addresses characteristic solutions that were developed in the control, fusion and multi-agent communities, along these four types of DKF approaches. Within each approach, popular solutions are assessed on their theoretical and practical capabilities, for which a common notation is employed to present their local estimation algorithms. The theoretical assessment involves an analysis under which conditions the corresponding DKF solution is *stable*, i.e., $P_i[\infty] := \lim_{k \rightarrow \infty} P_i[k]$ has bounded eigenvalues. After that, a practical evaluation is performed in two real-life inspired application examples, i.e., object tracking and environmental monitoring. A discussion on recent extensions in DKF completes the overview. Nonetheless, before the four DKF approaches are presented, let us start with the main contributions on synchronization and fusion methods, next. These two methods are frequently used to operate on the data exchanged between nodes.

6.2 Synchronization versus fusion

Synchronization and fusion methods typically consider a similar network to the one depicted in Figure 6.1, with the difference that each node i has a *parameter* $\theta_i \in \mathbb{R}^n$ that represents a local version of some global *parameter* $\theta \in \mathbb{R}^n$. See, also, the corresponding set-up illustrated in figure 6.2. Note that, for clarity of the proposed methods, synchronization as well as fusion do not consider time-varying signals, such as the process' state. Nonetheless, similar as to DKF, they define that nodes share their local parameter with neighboring nodes to improve the overall results of these local parameters θ_i as an estimate of the global θ , for all $i \in \mathcal{N}$. Synchronization has the

goal to achieve a consensus, while its fusion alternative has the objective to reduce uncertainty. As both will return in many DKF strategies, let us give a brief account on each of them. To that extent, let θ_i denote a random vector that is represented with a Gaussian distribution, i.e., $p(\theta_i) = G(\theta_i, \hat{\theta}_i, \Theta_i)$, for some suitable $\hat{\theta}_i$ and Θ_i .

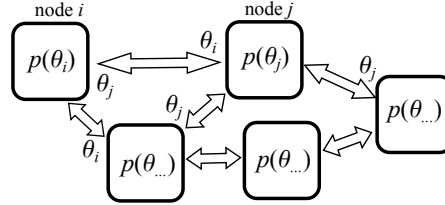


Figure 6.2: A network suitable for synchronization and fusion methods, where nodes exchange and improve their local parameters, e.g., θ_i of node i .

6.2.1 Synchronization

Synchronization methods aim to reduce conflicting results between the local parameters θ_i , for all $i \in \mathcal{N}$. Such an objective makes sense, as each θ_i is a local representative of the same global θ . Many distributed algorithms for synchronization (averaging or consensus) were proposed to diminish the difference of the means $\hat{\theta}_i - \hat{\theta}_j$, for any two $i, j \in \mathcal{N}$. See, for example, (Jadbabaie et al., 2003; Xiao and Boyd, 2004; Xiao et al., 2005; Tahbaz Salehi and Jadbabaie, 2010). Their general idea is to perform $L \in \mathbb{Z}_{>0}$ weighted averaging cycles in each node i on the local and neighboring parameters, i.e., on $\hat{\theta}_i$ and $\hat{\theta}_j$ for all $j \in \mathcal{N}_{i(1)}$. To that extent, let $\hat{\theta}_{i_s} \in \mathbb{R}^n$ denote the synchronized parameter-mean of node i after L averaging cycles. Further, let $W_{ii}, W_{ij} \in \mathbb{R}^{n \times n}$, for all $j \in \mathcal{N}_{i(1)}$, denote some weighting matrices. Then, synchronization methods perform the following recursive averaging cycle L times at each node i , for all $l \in \mathbb{Z}_{[1,L]}$ and $\theta_{i(0)} := \theta_i$, i.e.,

$$\hat{\theta}_{i(l+1)} = W_{ii} \hat{\theta}_{i(l)} + \sum_{j \in \mathcal{N}_{i(1)}} W_{ij} \hat{\theta}_{j(l)} \quad \text{and} \quad W_{ii} := I_n - \sum_{j \in \mathcal{N}_{i(1)}} W_{ij}, \quad (6.2)$$

Notice that the last cycle obtains the synchronized result, i.e., $\hat{\theta}_{i_s} := \hat{\theta}_{i(L)}$.

Most research on synchronization concentrates on developing suitable values for the weights W_{ij} (and W_{ii}). Some examples of scalar weights were proposed in (Jadbabaie et al., 2003; Xiao et al., 2005), where $d_i := \#\mathcal{N}_{i(1)}$ is the number of neighbors and $\epsilon < \min\{d_1, \dots, d_N\}$, i.e.,

$$\begin{aligned} \text{Nearest neighboring weights} & \quad W_{ij} := (1 + d_i)^{-1}, & \quad \forall j \in \mathcal{N}_{i(1)}; \\ \text{Maximum degree weights} & \quad W_{ij} := (1 + \epsilon)^{-1}, & \quad \forall j \in \mathcal{N}_{i(1)}; \\ \text{Metropolis weights} & \quad W_{ij} := (1 + \max\{d_i, d_j\})^{-1}, & \quad \forall j \in \mathcal{N}_{i(1)}. \end{aligned}$$

Each of these weights corresponds to a particular synchronization objective, though an important property of many methods is to preserve the average. Conditions to attain this property for the averaging cycles of (6.2) were established in (Xiao et al., 2005) with scalar weights $W_{ij} \in \mathbb{R}$. To that extent, let $\mathbf{1} \in \mathbb{R}^N$ and $\mathbf{1}\mathbf{1} \in \mathbb{R}^{N \times N}$ be conform to $\{\mathbf{1}\}_q := 1$ and $\{\mathbf{1}\mathbf{1}\}_{qr} := 1$, for all $q, r \in \mathcal{N}$. Further, $U \in \mathbb{R}^{N \times N}$ is defined via the elements $\{U\}_{qr} = W_{qr}$, for all $\{q, r \in \mathcal{N} | W_{qr} \text{ exists}\}$, while $\{U\}_{qr} = 0$ otherwise.

Theorem 6.2.1 *Let each node i perform the synchronization step of (6.2), for some $\theta_i \in \mathbb{R}^{n \times n}$, and let $\mathbf{1}^\top U = \mathbf{1}^\top$, $U\mathbf{1} = \mathbf{1}$ and $|\lambda_q(U - N^{-1} \cdot \mathbf{1}\mathbf{1}^\top)| < 1$, for all $q \in \mathbb{Z}_{[1, N]}$. Then $\lim_{L \rightarrow \infty} \hat{\theta}_{i(L)} = N^{-1} \sum_{j \in \mathcal{N}} \hat{\theta}_j$ holds for all $i \in \mathcal{N}$.*

A proof of this theorem is presented in (Xiao et al., 2005), while similar results are found in (Jadbabaie et al., 2003) as well. Therein, it was shown that synchronization with *nearest neighboring weights* does not preserve the average but that $\hat{\theta}_{i_s}$ depends on the network topology. The *maximum degree weights* and *metropolis weights* do satisfy the average value as a consensus. However, note that *maximum degree weights* require global information to establish ϵ in every node, which reduces its applicability in sensor networks.

6.2.2 Fusion

Fusion methods aim to reduce the uncertainty on the local parameters θ_i , for all $i \in \mathcal{N}$. To that extent, they typically propose a fusion function $\Omega : \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ that merges two parameters θ_i and θ_j into a new fused parameter $\theta_{i_f} \in \mathbb{R}^n$. Since all local parameters are characterized by a Gaussian distribution, let us assume that the fused random vector θ_{i_f} is Gaussian distributed as well, i.e., $p(\theta_{i_f}) := G(\theta_{i_f}, \hat{\theta}_{i_f}, \Theta_{i_f})$ for some suitable mean $\hat{\theta}_{i_f}$ and covariance Θ_{i_f} . Further, let us point out that fusion of more than two estimates is commonly conducted recursively. This means that θ_i is merged with the first received θ_j , after which their fusion result is further merged with the local parameter that is received next, and so on. Nonetheless, this section presents some novel fusion functions for merging *two* local parameters θ_i and θ_j .

A first important fusion function $\Omega(\cdot, \cdot)$ was presented in (Speyer, 1979; Bar-Shalom and Campo, 1986), when correlations are known.

Corollary 6.2.2 *Let θ_{i_f} denote any linear fusion result of the two local parameters θ_i and θ_j and let $\Theta_{ij} := \text{cov}(\theta_i, \theta_j)$. Then, the fusion function $\theta_{i_f} = \Omega(\theta_i, \theta_j)$, characterized by*

$$\begin{aligned} \Theta_{i_f} &= \Theta_i - (\Theta_i - \Theta_{ij})(\Theta_i + \Theta_j - \Theta_{ij} + \Theta_{ij}^\top)^{-1}(\Theta_i - \Theta_{ij}^\top) \text{ and} \\ \hat{\theta}_{i_f} &= \hat{\theta}_i + (\Theta_i - \Theta_{ij})(\Theta_i + \Theta_j - \Theta_{ij} + \Theta_{ij}^\top)^{-1}(\hat{\theta}_j - \hat{\theta}_i), \end{aligned}$$

establishes the unbiased fusion result with minimal $\text{tr}(\Theta_{i_f})$. In addition, $\Theta_{i_f} = (\Theta_i^{-1} + \Theta_j^{-1})^{-1}$ and $\hat{\theta}_{i_f} = \Theta_{i_f}(\Theta_i^{-1}\hat{\theta}_i + \Theta_j^{-1}\hat{\theta}_j)$, for $\Theta_{ij} = 0_{n \times n}$.

A drawback of the above fusion function is that the correlation $\text{cov}(\theta_i, \theta_j)$ should be available, which could induce impractical requirements for sensor networks. To solve the issue, alternative fusion methods were recently proposed that can cope with unknown correlations. Perhaps the most popular method was introduced as *covariance intersection* (CI) in (Julier and Uhlmann, 1997b). The fusion function of CI, which is denoted as $\theta_{i_f} = \Omega_{\text{CI}}(\theta_i, \theta_j)$, is characterized by a weighted averaging of the local parameters θ_i and θ_j , for some weight $\omega \in \mathbb{R}_{[0,1]}$, i.e.,

$$\Theta_{i_f} := (\omega\Theta_i^{-1} + (1-\omega)\Theta_j^{-1})^{-1}, \quad \hat{\theta}_{i_f} := \Theta_{i_f}(\omega\Theta_i^{-1}\hat{\theta}_i + (1-\omega)\Theta_j^{-1}\hat{\theta}_j),$$

The popularity of CI led to various approaches for determining ω , see, e.g., (Hanebeck et al., 2001; Niehsen, 2002; Franken and Hupper, 2005). Some examples, are $\omega = \text{tr}(P_j)(\text{tr}(P_j) + \text{tr}(P_i))^{-1}$ and the *Kullback-Leibler*¹ inspired weight $\omega = d(p_i||p_j)(d(p_i||p_j) + d(p_j||p_i))^{-1}$. Note the similarity of CI to synchronization of 6.2, which also holds the drawback of CI: a reduction in uncertainty after fusion is not attained, i.e., $\Theta_{i_f} \not\leq \Theta_i$ and $\Theta_{i_f} \not\leq \Theta_j$.

The fusion method *ellipsoidal intersection* (EI) of Chapter 4 does satisfy $\Theta_{i_f} \preceq \Theta_i$ and $\Theta_{i_f} \preceq \Theta_j$. The distinguishing feature of EI is an explicit characterization of unknown correlation, so that the fusion formulas are based on the independent parts of θ_i and θ_j that are to be fused. In line with the proposed characterization of unknown correlations, a summary of EI starts by introducing the *mutual covariance* $\Gamma \in \mathbb{R}^{n \times n}$ and *mutual mean* $\gamma \in \mathbb{R}^n$. Further, let $D_i, D_j \in \mathbb{R}^{n \times n}$ and $S_i, S_j \in \mathbb{R}^n$ denote two diagonal and rotation matrices, respectively, obtained by the eigenvalue decompositions $\Theta_i = S_i D_i S_i^{-1}$ and $D_i^{-\frac{1}{2}} S_i^{-1} \Theta_j S_i D_i^{-\frac{1}{2}} = S_j D_j S_j^{-1}$. Then the fusion function of EI, denoted as $\theta_{i_f} = \Omega_{\text{EI}}(\theta_i, \theta_j)$, for some $\varsigma \in \mathbb{R}_+$, yields

$$\Theta_{i_f} = (\Theta_i^{-1} + \Theta_j^{-1} - \Gamma^{-1})^{-1}, \quad (6.3a)$$

$$\hat{\theta}_{i_f} = \Theta_{i_f}(\Theta_i^{-1}\hat{\theta}_i + \Theta_j^{-1}\hat{\theta}_j - \Gamma^{-1}\gamma), \quad (6.3b)$$

$$\text{where, } \Gamma = S_i D_i^{\frac{1}{2}} S_j D_j S_j^{-1} D_i^{\frac{1}{2}} S_i^{-1},$$

$$D_\Gamma = \text{diag}_{q \in \mathbb{Z}_{[1,n]}} (\max \{1, \{D_j\}_{qq}\}), \quad (6.3c)$$

$$\gamma = (W_i + W_j)^{-1}(W_i\hat{\theta}_i + W_j\hat{\theta}_j),$$

$$W_i = \Theta_j - \Gamma + \varsigma I_n, \quad W_j = \Theta_i - \Gamma + \varsigma I_n.$$

The value of ς allows for a numerically robust computation of the *mutual mean* γ . A suitable value, yields $\varsigma = 0$ if $|1 - \{D_j\}_{qq}| > 10\epsilon$ holds for all $q \in \mathbb{Z}_{[1,n]}$, while $\varsigma = \epsilon$ otherwise, for some approximation parameter $\epsilon \in \mathbb{R}_{>0}$. More details on this fusion method are found in Chapter 4. Here, let us continue with a brief comparison of synchronization versus fusion.

¹The *Kullback-Leibler* divergence $d(\cdot||\cdot)$ was introduced in Section 3.2.1.

Based on their formulas, the main difference between synchronization and fusion is the incorporation of the local covariance Θ_i , which models the uncertainty (error) on the local parameter θ_i as an estimate of θ . In addition, let us recall the illustrative comparison of synchronization and fusion presented in Chapter 5. Figure 6.3 depicts this comparison, which is obtained when θ_i and θ_j are either synchronized or fused with each other. The result of synchronization, i.e., $p(\theta_{i_s}) = G(\theta_{i_s}, \hat{\theta}_{i_s}, \Theta_{i_s})$, is computed according to one averaging cycle of (6.2) with $L = 1$ and $W_{ij} = 0.1$. Note that, in line with standard synchronization methods, only the means $\hat{\theta}_i$ and $\hat{\theta}_j$ are synchronized and that $\Theta_{i_s} = \Theta_i$. The fusion alternative employs the EI formulas of (6.3) to compute $p(\theta_{i_f}) = G(\theta_{i_f}, \hat{\theta}_{i_f}, \Theta_{i_f})$. Let us emphasize that Figure 6.3 is not included to decide which method is better. It is merely an example to illustrate the goal of synchronization (reduce conflicting results) with respect to the goal of fusion (reduce uncertainty), as they will re-appear in the theoretical assessment of the survey that is presented next.

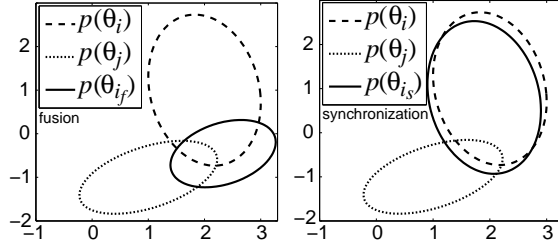


Figure 6.3: A comparison of synchronization versus fusion. Note that PDFs are represented as ellipsoidal sub-level-set, i.e., $G(\theta, \mu, \Sigma) \rightarrow \mathcal{E}_{\mu, \Sigma}$. A graphical characterization of such a sub-level-set is found in Figure 1.9, though let us point out that a larger covariance Σ implies a larger area-size of $\mathcal{E}_{\mu, \Sigma}$

6.3 Exchanging measurements

The first two DKF approaches of this overview consider the exchange of local measurements. As such, each node i receives the local measurement y_j of its neighboring nodes $j \in \mathcal{N}_{i(1)}$, which can then be fused (approach 1) or synchronized (approach 2). Notice that y_i and y_j , for any two $i, j \in \mathcal{N}$, may have a different representations, i.e., C_i is not necessarily equal to C_j . Hence, one cannot fuse nor synchronize y_i and y_j instantaneously, as they could correspond to different elements of x . To solve this issue, each local measurement $y_i[k] = C_i x[k] + v_i[k]$ is transformed into its *information form*, which is characterized by the *information state* $z_i \in \mathbb{R}^n$ and *information covariance* $Z_i \in \mathbb{R}^{n \times n}$, i.e.,

$$z_i[k] := C_i^\top V_i^{-1} y_i[k] \quad \text{and} \quad Z_i[k] := C_i^\top V_i^{-1} C_i, \quad \forall i \in \mathcal{N}, k \in \mathbb{Z}_+. \quad (6.4)$$

The main reason that the term “information” is used, for denoting z_i and Z_i , is because the same transformation was introduced in the *Information filter* of (1.5). This Information filter has equivalent estimation results as the Kalman filter but is less complex in the presence of many measurements. For that reason, the Information filter and thereby, the information form of y_i , is a favorable approach for estimation in sensor networks.

6.3.1 Approach 1: fusion of measurements

Existing solutions that are in line with the first DKF approach perform fusion on the exchanged measurements, for which it is typically assumed that local measurements are independent, i.e., $cov(y_i, y_j) = 0_{n \times n}$ for all $i, j \in \mathcal{N}$. These measurements can be shared in their normal form or in their information form. The information form of z_i and Z_i is chosen here, as it imposes fewer requirements and simplifies the fusion formulas. In addition, note that the correlation is available for fusion as well, due to which the fusion formulas presented in Corollary 6.2.2 can be employed. However, it was shown in (Durant-Whyte et al., 1990) that these optimal fusion formulas can be rewritten into a simple addition, when the information form is used, i.e., $z_{i_f}[k] = z_i[k] + z_j[k]$ and $Z_{i_f}[k] = Z_i[k] + Z_j[k]$. As fusion of more than two variables is commonly conducted recursively, the fused measurement at node i obtained after merging $z_i[k]$ and $Z_i[k]$ with the measurement information received from neighboring nodes, i.e., $z_j[k]$ and $Z_j[k]$ for all $j \in \mathcal{N}_{i(1)}$, yields

$$z_{i_f}[k] := z_i[k] + \sum_{j \in \mathcal{N}_{i(1)}} z_j[k] \quad \text{and} \quad Z_{i_f}[k] := Z_i + \sum_{j \in \mathcal{N}_{i(1)}} Z_j[k]. \quad (6.5)$$

Note that the above fused measurement is still in the information form. Therefore, it can directly be used as input to an Information filter at node i for computing a local estimate of the state, as depicted in Figure 6.4.

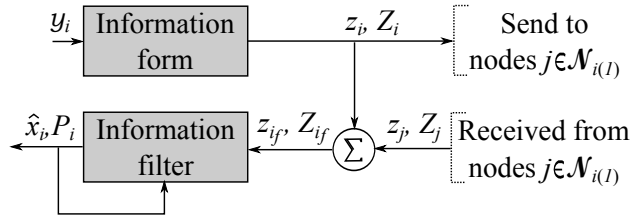


Figure 6.4: Schematic set-up of a local estimation algorithm performed by a node i that typically corresponds to a DKF approach where local measurements from neighboring nodes $j \in \mathcal{N}_{i(1)}$ are fused before being exploited by an Information filter for computing the local estimation results \hat{x}_i and P_i .

The DKF solution of Figure 6.4 was proposed in (Durant-Whyte et al.,

1990) and is labeled as *distributed information filter* (DIF). Therein, a node i performs the following DIF algorithm, for $\mathcal{N}_{i(0,1)} := \{i\} \cup \mathcal{N}_{i(1)}$.

Algorithm of the DIF

$$\begin{aligned} P_i[k^-] &= AP_i[k-1]A^\top + Q; \\ P_i[k] &= \left(P_i^{-1}[k^-] + \sum_{j \in \mathcal{N}_{i(0,1)}} Z_j[k] \right)^{-1}; \\ \hat{x}_i[k] &= P_i[k] \left(P_i^{-1}[k^-] A \hat{x}_i[k-1] + \sum_{j \in \mathcal{N}_{i(0,1)}} z_j[k] \right). \end{aligned}$$

The notation k^- emphasizes predicted variables from updated ones. This simple, yet effective, DKF solution triggered many extensions of the DIF to reduce communication requirements. For example, by asynchronous communication, or by quantization of the exchanged values, see, e.g., (Mallick et al., 2001; Ribeiro et al., 2006, 2010). On top of that, several assessments of the DIF were presented in (Schlosser and Kroschel, 2007; Hasu and Koivo, 2006), addressing out-of-sequence-measurements and adaptive communication rates.

An additional assessment in this overview involves the asymptotic behavior of the DIF. To that extent, one can derive that the Information filter with z_{i_f} and Z_{i_f} of (6.5) establishes equivalent estimation results as a Kalman filter that is based on the fused measurement $y_{i_f}[k] := H_i x[k] + v_{i_f}$, where $p(v_{i_f}) = G(v_{i_f}, 0, R_i)$, $R_i := \text{diag}_{j \in \mathcal{N}_{i(0,1)}}(V_j)$ and $H_i := \text{col}_{j \in \mathcal{N}_{i(0,1)}}(C_j)$. As such, the asymptotic properties of $P_i[k]$ of the DIF are equivalent to the ones of the error-covariance of a Kalman filter based y_{i_f} .

Theorem 6.3.1 *Let (A, H_i) be an observable pair. Then, there exists a unique and stabilizing solution $P_i[\infty] := \lim_{k \rightarrow \infty} P_i[k]$ of the DIF at node i , independent of $P_i[-1]$, if $\lambda_q(A) \leq 1$, for all $q \in \mathbb{Z}_{[1,n]}$.*

A proof of this theorem directly results from the asymptotic properties of a Kalman filter that were established by analyzing convergence of the Ricatti difference equation in (Chan et al., 1984). The theorem further indicates an important design aspect of the DIF, i.e., a local estimate x_i is based on the node's local and neighboring measurements $\cup_{j \in \mathcal{N}_{i(0,1)}} \{y_j\}$. Hence, only the measurements produced in the direct neighborhood of a node are exploited for computing its local estimate. This further means that the DIF is most successful in applications where the global state can be estimated from its own and neighboring measurements, for example, object tracking applications. However, this further implies that different nodes will obtain different local estimation results, as a local estimate x_i is based on a (unique) subset of local measurements. This issue is addressed in the next DKF approach.

6.3.2 Approach 2: synchronization of measurements

Existing solutions in line with the second DKF approach perform synchronization on the exchanged measurements that, similar as to DKF solutions in the previous approach, are assumed to be independent. Further, since y_i and y_j , for any two $i, j \in \mathcal{N}$, may have different representations, the presented solutions will become less complex when nodes share their measurement y_i in the information form of (6.4). The local measurement information z_i and Z_i are then individually synchronized with the corresponding received measurement information z_j and Z_j from its neighboring nodes $j \in \mathcal{N}_{i(1)}$, by performing L averaging cycles in line with (6.2). After L averaging cycles a node i obtains the synchronized measurement in an information form, which, similar as to the DIF, can directly be used as input to an Information filter for computing a local estimate of the state, as depicted in Figure 6.4.

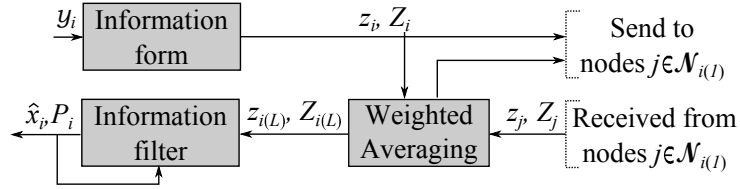


Figure 6.5: Schematic set-up of a local estimation algorithm performed by a node i that typically corresponds to a DKF approach in which local measurements from neighboring nodes $j \in \mathcal{N}_{i(1)}$ are synchronized before being exploited by an Information filter for computing \hat{x}_i and P_i .

Existing DKF solutions that are in line with the set-up of Figure 6.5 are found in (Olfati-Saber, 2007; Kirti and Scaglione, 2008) and are labeled as consensus information filter (CIF). Therein, a node i performs the following CIF algorithm, for some $z_{i(0)}[k] := C_i^T V_i^{-1} y_i[k]$ and $Z_{i(0)}[k] := C_i^T V_i^{-1} C_i$.

Algorithm of the CIF

Synchronisation on local measurements

for $l = 1 : L$, do

$$Z_{i(l)}[k] = \sum_{j \in \mathcal{N}_{i(0,1)}} W_{ij} Z_{j(l-1)}[k], \quad z_{i(l)}[k] = \sum_{j \in \mathcal{N}_{i(0,1)}} W_{ij} z_{j(l-1)}[k];$$

end

Information filter

$$P_i[k^-] = A P_i[k-1] A^T + Q;$$

$$P_i[k] = (P_i^{-1}[k^-] + Z_{i(L)}[k])^{-1};$$

$$\hat{x}_i[k] = P_i[k] (P_i^{-1}[k] A \hat{x}_i[k-1] + z_{i(L)}[k]).$$

The idea of the CIF is that synchronization on local measurements induces synchronization of local estimates, which can also be proven.

Corollary 6.3.2 *Let each node i employ CIF with maximum degree weights or metropolis weights. Further, let $\hat{x}_i[0]$ and $P_i[0]$ be equal for all $i \in \mathcal{N}$. Then, also $\hat{x}_i[k]$ and $P_i[k]$ are equal for all $i \in \mathcal{N}$, $k \in \mathbb{Z}_+$ and $L \rightarrow \infty$.*

A proof of this corollary is a direct result of the fact that $z_{i(L)}[k]$ and $Z_{i(L)}[k]$ attain the averaged measurement in information form at each node, i.e., $Z_{i(L)}[k] = N^{-1} \sum_{i \in \mathcal{N}} C_i^T V_i^{-1} C_i$ and $z_{i(L)}[k] = N^{-1} \sum_{i \in \mathcal{N}} C_i^T V_i^{-1} y_i[k]$ hold for all $i \in \mathcal{N}$ and $L \rightarrow \infty$. Since these values are equal in every node at each sample k , the resulting local estimate $x_i[k]$ is equivalent for all $i \in \mathcal{N}$. In reality, the CIF will already achieve similar results for smaller L depending on the network size. However, a typical requirement in DKF, for satisfying the limitations in communication, is that data is exchanged once per sample instant, i.e., $L = 1$. In that case, the next result shows that the DIF obtains a smaller error-covariance than the CIF. To that extent, let $P_i^{\text{DIF}}[k]$ and $P_i^{\text{CIF}}[k]$ denote their corresponding error-covariance at node i , respectively.

Lemma 6.3.3 *Let each node i employ the DIF in parallel to the CIF, with $L = 1$ and $P_i^{\text{CIF}}[0] = P_i^{\text{DIF}}[0]$. Then, $P_i^{\text{DIF}}[k] \preceq P_i^{\text{CIF}}[k]$ holds for all $k \in \mathbb{Z}_+$.*

The lemma is proven in Appendix F.1.

The CIF is not a popular DKF approach as it requires to communicate L times at each sample instant, while communication is one of the limiting resources in sensor networks. Moreover, if $L = 1$, then Theorem 6.3.3 indicates that the DIF achieves better estimation results. Nonetheless, a theoretical extension of the CIF was presented in (Kamgarpour and Tomlin, 2008). Therein, conditions on the weights $W_{ij} \in \mathbb{R}^{n \times n}$ were derived to guarantee that $z_{i(L)}$ and $Z_{i(L)}$ represent the *fused* rather than the *synchronized* measurement. This would then imply that the CIF mimics the centralized Kalman filter. However, no actual values of the weights were indicated, by which the fused result would be obtained, while $L \rightarrow \infty$ is required as well. Therefore, given that the objective is synchronizing x_i , for all $i \in \mathcal{N}$, literature indicates that exchanging local state-estimates is more effective, which is shown next.

6.4 Exchanging local estimates

The next two DKF approaches exchange the local estimates, which are typically determined by pre-processing the node's own measurement locally with a Kalman filter. Then, shared local estimates are either synchronized (approach 3) or fused (approach 4). One of the main advantages when exchanging local estimates is that local measurement information spreads through the entire network, even when nodes exchange data only once per sample instant. This is probably why the next two DKF approaches have recently gained attention, especially the third approach.

6.4.1 Approach 3: synchronization of state estimates

Existing solutions that are in line with the third DKF approach synchronize the exchanged local state estimates. Some first aspects of such an approach were studied by performing one averaging cycle of (6.2) on the results of the DIF in each node i . i.e., substituting $\hat{\theta}_{i(0)} = \hat{x}_i[k]$, $\hat{\theta}_{j(0)} = \hat{x}_j[k]$ and $L = 1$ into (6.2). An illustration of such a DKF solution is depicted in Figure 6.6, while some studies associated to this DKF are presented in (Olfati-Saber and Shamma, 2005; Olfati-Saber, 2007; Kirti and Scaglione, 2008). A simplified implementation was assessed in (Carli et al., 2008). Therein, each node i employs the following set of equations, in which $\mathcal{N}_{i(0,1)} := \{i\} \cup \mathcal{N}_{i(1)}$, for computing a local mean $\hat{x}_{i_s}[k]$ that is synchronized with local means obtained in other nodes of the network.

$$\hat{x}_{i_s}[k] = \sum_{j \in \mathcal{N}_{i(0,1)}} W_{ij} \hat{x}_j[k], \quad \hat{x}_i[k] = (1 - \kappa_i) \hat{x}_{i_s}[k-1] + \kappa_i y_i[k], \quad (6.6)$$

Note that the variable κ_i in (6.6) could be regarded as the Kalman gain. In addition, the assessment presented in (Carli et al., 2008) showed that minimization of the estimation error by jointly optimizing κ_i and W_{ij} , for all $j \in \mathcal{N}_{i(0,1)}$, is a non-convex problem. Hence, choosing κ_i as the Kalman gain affects the weights W_{ij} and vice-versa, which raised new challenges.

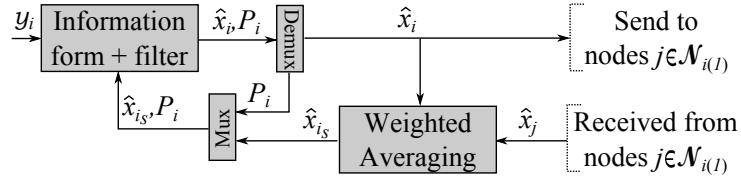


Figure 6.6: Schematic set-up of a local estimation algorithm performed by a node i that typically corresponds to a DKF approach where local estimates from neighboring nodes $j \in \mathcal{N}_{i(1)}$, obtained by their Information filter, are synchronized. The routine “Demux” separates the incoming variables \hat{x}_i and P_i , whereas “Mux” combines the incoming variables \hat{x}_{i_s} and P_i .

A popular DKF solution that addresses such a joint optimization was introduced as the *distributed consensus information filter* (DCIF) in (Olfati-Saber, 2009). Therein, a node i performs the following DCIF algorithm, for some design parameter $\epsilon \in \mathbb{R}_{(0.1\tau_s, 10\tau_s)}$.

Algorithm of the DCIF**Information filter**

$$P_i[\mathbf{k}^-] = AP_i[\mathbf{k}-1]A^\top + Q;$$

$$P_i[\mathbf{k}] = \left(P_i^{-1}[\mathbf{k}^-] + \sum_{j \in \mathcal{N}_{i(0,1)}} Z_j[\mathbf{k}] \right)^{-1};$$

$$\hat{x}_i[\mathbf{k}] = P_i[\mathbf{k}] \left(P_i^{-1}[\mathbf{k}^-] A \hat{x}_{i_s}[\mathbf{k}-1] + \sum_{j \in \mathcal{N}_{i(0,1)}} z_j[\mathbf{k}] \right);$$

Synchronisation

$$\mu_i[\mathbf{k}^-] = \sum_{j \in \mathcal{N}_{i(1)}} A(\hat{x}_{j_s}[\mathbf{k}-1] - \hat{x}_{i_s}[\mathbf{k}-1]);$$

$$\hat{x}_{i_s}[\mathbf{k}] = \hat{x}_i[\mathbf{k}] + \epsilon P_i[\mathbf{k}^-] (\text{tr}(P_i^{-\top}[\mathbf{k}^-] P_i[\mathbf{k}^-])^{\frac{1}{2}} + 1)^{-1} \mu_i[\mathbf{k}^-].$$

A first observation of the DCIF algorithm is that its error-covariance $P_i[\mathbf{k}]$ is computed with the same formulas that were used for determining the error-covariance of the DIF. Hence, the asymptotic properties of $P_i[\mathbf{k}]$ in the DCIF correspond to the results obtained in Theorem 6.3.1. Moreover, note that the DCIF and the DIF are equivalent DKF solutions in case $\epsilon = 0$. A different observation is that the first and second moment of $x_i[\mathbf{k}]$, i.e., $\hat{x}_{i_s}[\mathbf{k}]$ and $P_i[\mathbf{k}]$, have different update formulas, since only the means are synchronized. A consequence of this difference is $P_i[\mathbf{k}]$ does *not* represent a model for $\text{cov}(x_i[\mathbf{k}] - \hat{x}_{i_s}[\mathbf{k}])$ anymore. Or differently, that $P_i[\mathbf{k}]$ is a poor approximation of the uncertainty of the local estimate $x_i[\mathbf{k}]$ and thus of its estimation-error. However, the main objective of the DCIF is to attain a consensus on the mean of local estimates, which was proven in (Olfati-Saber, 2009).

Theorem 6.4.1 *Let each node i perform the DCIF and let $\sum_{j \in \mathcal{N}_{i(0,1)}} Z_j \succ 0$, for all $i \in \mathcal{N}$. Then $\lim_{\mathbf{k} \rightarrow \infty} \|\hat{x}_{i_s}[\mathbf{k}] - x[\mathbf{k}]\|_2$ is bounded for a sufficiently small $\epsilon \in \mathbb{R}_{>0}$. Moreover, $\lim_{\mathbf{k} \rightarrow \infty} \hat{x}_{i_s}[\mathbf{k}]$ has equivalent values for all $i \in \mathcal{N}$.*

The DCIF is one of many solutions that fit within this DKF strategy. See, for example, an extensive survey on synchronization algorithms for control and estimation purposes in (Garin and Schenato, 2011). Therefore, let us briefly address three other popular DKF solutions that are associated to this approach of a synchronization on local estimates.

- **Distributed minimum variance estimator** (Speranzon et al., 2008): The method computes only a mean of x_i , for a process model in (6.1) that is simplified to the scalar case $x \in \mathbb{R}$ and $A = 1$. The update then involves $\hat{x}_i[\mathbf{k}] = \sum_{j \in \mathcal{N}_{i(0,1)}} W_{ij} \hat{x}_j[\mathbf{k}-1] + \sum_{j \in \mathcal{N}_{i(0,1)}} \kappa_{ij} y_j[\mathbf{k}]$. Algebraic formulas of the weights $W_{ij} \in \mathbb{R}^{n \times n}$ and $\kappa_{ij} \in \mathbb{R}^{n \times m_i}$, for all $i, j \in \mathcal{N}$, are derived by minimizing the estimation error and can be determined on-line.

- **Off-line DKF** (Alriksson and Rantzer, 2006): The method computes only a mean of x_i in line with (6.6), while including a process model, i.e., $\hat{x}_{i_s}[k] = \sum_{j \in \mathcal{N}_{i(0,1)}} W_{ij} \hat{x}_j[k]$ and $\hat{x}_i[k] = (1 - \kappa_i C_i) A \hat{x}_{i_s}[k-1] + \kappa_i y_i[k]$. An extension of this off-line DKF that takes communication noise into account was presented in (Mosquera and Jayaweera, 2008). However, as the weights $W_{ij} \in \mathbb{R}^{n \times n}$ and $\kappa_{ij} \in \mathbb{R}^{n \times m_i}$ are jointly determined off-line, for all $i, j \in \mathcal{N}$, the method is sensitive to system changes.
- **Kalman consensus algorithm** (Ren et al., 2005): The method synchronizes both the mean and covariance. Hence, $P_i[k] = (P_{i_s}^{-1}[k^-] + Z_i)^{-1}$ and $\hat{x}_i[k] = P_i[k](P_{i_s}^{-1}[k^-] \hat{x}_{i_s}[k^-] + z_i[k])$, in which $P_{i_s}[k^-]$ and $\hat{x}_{i_s}[k^-]$ are individually obtained by one averaging cycle of (6.2) on their corresponding predicted variables at node i and its neighboring nodes $j \in \mathcal{N}_{i(1)}$. The original method included communication noise, which is omitted for clarity of the algorithm. A similar DKF method for synchronizing P_i^{-1} and $P_i^{-1} \hat{x}_i$ is found in (Casbeer and Beard, 2009).

One of the most important design aspects for this third DKF approach, is that the concept of synchronization will balance (not reduce!) the estimation error in the network. As such, nodes with accurate sensor readings will still obtain erroneous estimates, when other nodes in the network have inaccurate sensor readings. Or worse, if x is not observable from the measurements acquired by one node i of the network, i.e., from $\cup_{j \in \mathcal{N}_{i(0,1)}} \{y_j\}$, then this node i yields unstable estimation results and thus a diverging estimation error and error-covariance. More importantly, the other nodes in the network will duplicate this behavior as a result of the synchronization procedure. For that reason, it is required that each node i attains stable estimation results that are based on the acquired measurements only. Such a requirement was also necessary to establish Theorem 6.4.1, i.e., $\sum_{j \in \mathcal{N}_{i(0,1)}} Z_j \succ 0$. Therefore, based on this brief theoretical analysis, the third DKF approach is most effective when nodes have similar local measurements that result in an observable local estimate. This makes object tracking a favorable application scenario for DKF solution that synchronize local estimates. Further, note that the weights W_{ij} should be chosen in line with Theorem 6.2.1 to preserve the average and not introduce a bias on the estimates. Solutions to prevent the spread of instability caused by one node are found in the next DKF approach.

6.4.2 Approach 4: fusion of state estimates

Existing solutions that are in line with the fourth DKF approach perform fusion on the exchanged local state estimates. Typically, each node employs a Kalman filter (or Information filter) for processing its own local measurement $y_i[k]$ and thereby, compute an updated local estimate of $x_i[k]$ conform a Gaussian distribution. Local estimates are then shared with neighboring nodes as input to a state fusion method, see Figure 6.7. The resulting fused

estimate is denoted as $x_{i_f}[k]$ and follows a Gaussian distribution, i.e., $p(x_{i_f}) = G(x_{i_f}, \hat{x}_{i_f}, P_{i_f})$, for some suitable $\hat{x}_{i_f}[k]$ and $P_{i_f}[k]$. The survey focusses on fusion methods that can cope with unknown correlations. Otherwise, the sensor network should keep track of all the estimates that are shared between nodes, which imposes unrealistic requirements.

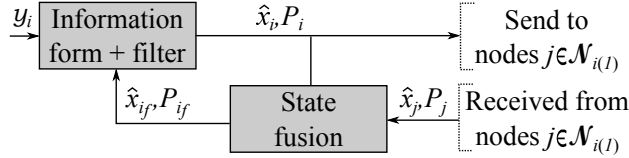


Figure 6.7: Schematic set-up of a local estimation algorithm performed by node i that is typical for DKF approach where local estimates from neighboring nodes $j \in \mathcal{N}_{i(1)}$, obtained by their Information filter, are fused.

Standard fusion methods merge two local estimates $x_i[k]$ and $x_j[k]$ into a single fused estimate. Fusion of more than two estimates is commonly conducted recursively. This means that node i fuses $x_i[k]$ of its Information filter with the first received estimate, denoted as $x_{j(1)}[k]$, after which their fusion result is further merged with the estimate that is received next, i.e., $x_{j(2)}[k]$, and so on. Therefore, let us denote $p(x_{i(l)}[k]) = G(x_{i(l)}[k], \hat{x}_{i(l)}[k], P_{i(l)}[k])$ as the fusion result of node i , after fusing $p(x_{i(l-1)}[k])$ with the l -th received estimate $p(x_{j(l)}[k]) = G(x_{j(l)}[k], \hat{x}_{j(l)}[k], P_{j(l)}[k])$. A DKF solution that adopts this reasoning was introduced in Chapter 5 as *cooperative Kalman filters* (CKFs). CKFs employs the state fusion method *ellipsoidal intersection* to merge two estimates according to the fusion function $\Omega_{\text{Ei}}(\cdot, \cdot)$ of (6.3). As such, the corresponding CKFs algorithm performed by a node i , yields

Algorithm of the CKFs

Information filter

$$P_i[k^-] = AP_{i_f}[k-1]A^\top + Q;$$

$$P_i[k] = (P_i^{-1}[k^-] + Z_i[k])^{-1};$$

$$\hat{x}_i[k] = P_i[k](P_i^{-1}[k^-]A\hat{x}_{i_f}[k-1] + z_i[k]);$$

Fusion

$$\hat{x}_{i(0)} := \hat{x}_i[k] \quad \text{and} \quad P_{i(0)} := P_i[k];$$

for $l = 1, \dots, L$ and $L := \#\mathcal{N}_{i(1)}$, do:

$$\hat{x}_{j(l)} := \hat{x}_j[k] \quad \text{and} \quad P_{j(l)} := P_j[k], \quad j \in \mathcal{N}_{i(1)};$$

$$x_{i(l)} = \Omega_{\text{Ei}}(x_{i(l-1)}, x_{j(l)});$$

end

$$\hat{x}_{i_f}[k] = \hat{x}_{i(L)}, \quad P_{i_f}[k] = P_{i(L)}.$$

The above CKFs employs *ellipsoidal intersection* for state fusion, as it guarantees a reduction in the uncertainty after fusion. This is a vital property for proving asymptotic bounds on $P_{i_f}[k]$, which is recalled from Section 5.2.3. Let us simplify those results, so to improve the clarity of the statement.

Theorem 6.4.2 *Let each node i perform the CKFs and let there exists at least one node $j \in \mathcal{N}$, for which (A, C_j) is an observable pair and $\lambda_q(A) \leq 1$ holds for all $q \in \mathbb{Z}_{[1,n]}$. Then, $\lambda_{\max}(P_{i_f}[\infty])$ is bounded for all nodes $i \in \mathcal{N}$.*

A different property of CKFs is that the information of any local measurement y_i is exploited for computing a local estimate x_j at any (other) nodes $j \in \mathcal{N}$. A similar property also holds for the following DKF solution.

The *gossip interactive Kalman filter* (GIKF), which was introduced in (Kar, 2011), starts with the same Information filter as the CKFs for computing $P_i[k]$ and $\hat{x}_i[k]$ of the local estimate $x_i[k]$. However, $x_{i_f}[k]$ is determined by *swapping* the local x_i with x_j of a randomly selected neighboring node j . As such, $\hat{x}_{i_f}[k] = \hat{x}_j[k]$, $P_{i_f}[k] = P_j[k]$ and $\hat{x}_{j_f}[k] = \hat{x}_i[k]$ and $P_{j_f}[k] = P_i[k]$ are followed for one unique $j \in \mathcal{N}_{i(1)}$ at each instant $k \in \mathbb{Z}_+$. It was proven in (Kar, 2011) that the probability of an unbounded $P_{i_f}[k]$, i.e., $\Pr(|P_{i_f}| \rightarrow \infty)$, is zero for all $i \in \mathcal{N}$, given that three conditions are satisfied.

Theorem 6.4.3 *Let A^{-1} exists, $(A, Q^{\frac{1}{2}})$ be a stabilizable pair and let $\sum_{i=1}^N C_i^T C_i$ have full rank. Then $\lim_{\alpha \rightarrow \infty} \sup_{k \in \mathbb{Z}_+} \Pr(|P_{i_f}[k]| \geq \alpha) = 0$.*

An important design consideration of GIKF is that the local estimate x_i is based on a unique set of measurements. This is because $y_i[k]$ is used to update $x_i[k]$ and it is not shared with any other node. Yet, each local estimate $x_i[k]$ is based on sensor readings from all the nodes in the network, for large k .

An important design aspect, for both the GIKF and CKFs, is that the mean and error-covariance of a local estimate x_i are based on the available sensor readings from all over the network. Such DKF solutions are beneficial in sensor networks that have many different types of local measurements. In additional, this fourth DKF approach does not require any initialization with neighboring nodes, for example, to re-evaluate the weights W_{ij} for synchronization. Hence, fusion of local estimates is advantageous in large-scale sensor networks that are subject to system changes, e.g., in the network topology, or new nodes that are added to an existing network during operation. However, please note that a trade-off for attaining the reduction in uncertainty (and estimation error) with CKFs is an increase of computational complexity.

6.5 Illustrative examples

The theoretical overview on DKF presented in the previous sections summarized the different approaches that are found in literature. An indication of

their main design considerations was given for each approach, though a practical evaluation of various DKF solutions in two real-life application examples is presented in this section. The first application is multi-object tracking on a parking lot using four static cameras. The second application is following a 2D environmental diffusion process in time with a large-scale, ad-hoc sensor network. Six DKF solutions are compared according to the four approaches.

- The DIF from approach 1, i.e., fuse measurements;
- The CIF from approach 2, i.e., that synchronize measurements;
- The DCIF from approach 3, i.e., synchronize estimates. In addition, a second DKF solution associated to this approach is included. This solution is denoted as DKF-SC and follows the set-up of Figure 6.6;
- The CKFs and GIKF from approach 4, i.e., fuse estimates.

6.5.1 Object tracking with camera

The first application example involves object tracking, in which the goal is to track 10 moving objects (humans) in a small network of four cameras (nodes). The cameras are placed 3 meters above the ground at different locations to cover a certain range of a parking lot in \mathbb{R}_{XY} , i.e., \mathbb{R}_{XY} represents the ground plane having a X -direction and a Y -direction. The camera image is updated at approximately 10 [Hz] and is processed for detecting moving objects in the stream of camera images. This is done by estimating a background image of the camera, via low-pass filtering, which after subtraction of the current camera image results in “blobs” that each could represent a moving object. The center-of-mass of each blob is referred to as *detection*. It cannot be assumed that each detection is related to a moving object in the real scene, as false detections will be present. Figure 6.8(a) illustrates a top-view of the application set-up and Figure 6.8(b) depicts a snapshot of camera 4.

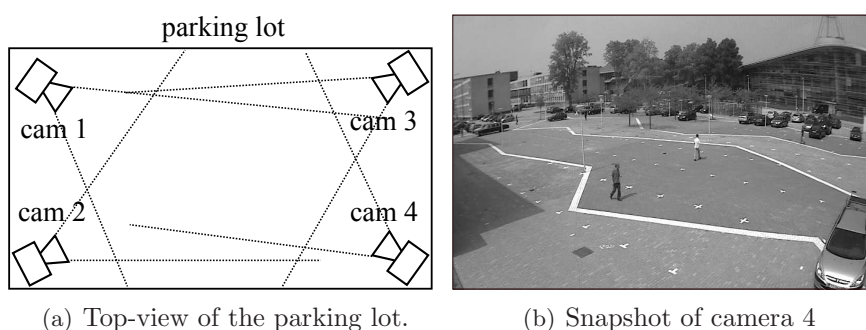


Figure 6.8: The object tracking set-up in a parking lot with 4 cameras

Communication Each camera can exchange data with neighboring cameras, i.e., $\mathcal{N}_{1(1)} = \{2, 4\}$, $\mathcal{N}_{2(1)} = \{1, 3\}$, $\mathcal{N}_{3(1)} = \{2, 4\}$, $\mathcal{N}_{4(1)} = \{1, 3\}$.

Process Tracking a single human is equivalent to estimating the position and speed of its random walk in time. Therefore, let the state $x^{(q)} \in \mathbb{R}^4$ denote a collection of the position and speed of the q -th human (in both the X -direction and Y -direction). Then, the process model is described by two double integrators, for which the sampling time is $\tau_s = 0.5$, i.e.,

$$x^{(q)}[k] = \begin{pmatrix} 1 & \tau_s & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau_s \\ 0 & 0 & 0 & 1 \end{pmatrix} x^{(q)}[k-1] + w[k-1].$$

The process-noise $w \in \mathbb{R}^4$ depends on the unknown acceleration of humans and is characterized by $Q = \begin{pmatrix} W & 0_{2 \times 2} \\ 0_{2 \times 2} & W \end{pmatrix}$ and $W = 0.15 \begin{pmatrix} \frac{1}{4}\tau_s^4 & \frac{1}{2}\tau_s^3 \\ \frac{1}{2}\tau_s^3 & \tau_s^2 \end{pmatrix}$. Local position measurements $y_i \in \mathbb{R}_{XY}$ of a q -th human are obtained from the detections in the coordinate-frame of the corresponding camera (camera-frame), i.e., by projecting a detection on \mathbb{R}_{XY} . See also Figure 6.9 for an illustration. However, since none of the cameras has information on which q -th human caused what detection, a local measurement y_i at camera i is of the following form:

$$y_i[k] = C_i x^{(q)}[k] + v_i[k] \quad \text{and} \quad C_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (6.7)$$

Based on these local measurements, the DKF methods compute a local estimate on $x_i^{(q)}$ of the q -th human in each i -th camera, i.e., for some suitable mean $\hat{x}_i^{(q)}$ and error-covariance $P_i^{(q)}$. To that extent, the measurement y_i of (6.7) must first be associated to a human q , as this information is not available in the detections. Association is done according to the method proposed in (Bar-Shalom and Li, 1995), which computes the following probabilistic measure

$$d_{i,q} := e^{-\left(C_i \hat{x}_i^{(q)} - y_i\right)^\top \left(C_i^\top \left(P_i^{(q)}\right)^{-1} C_i + V_i\right)^{-1} \left(C_i \hat{x}_i^{(q)} - y_i\right)}, \quad \forall q \in \mathbb{Z}_+.$$

The highest “probability” $d_{i,q}$ then indicates to which human q the local measurement y_i should be associated with. Still, any $d_{i,q}$ must satisfy a lower threshold to be associated, so to prevent the association of false detections. Further, let us point out that when cameras exchange local measurements, then they include the association information.

A practical aspect of the measurement-noise $v_i \in \mathbb{R}_{XY}$ is that it depends on the location of the human with respect to the camera. In general, $cov(v_i) = V_i$ will increase with a larger distance of the human to the camera. This aspect is also illustrated in Figure 6.9, though further details are omitted for clarity of the comparison.

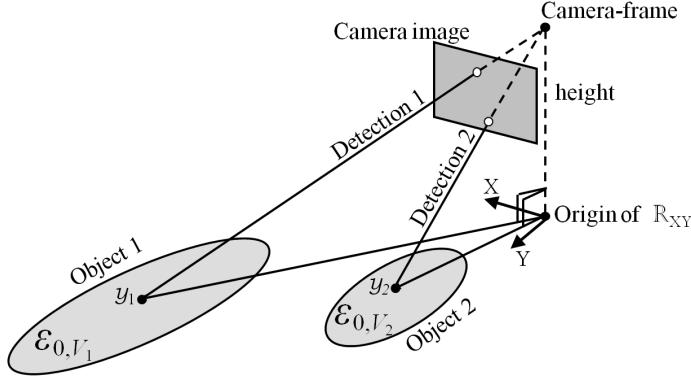


Figure 6.9: A projection from the camera-frame to \mathbb{R}_{XY} implies that the measurement-noise v_i on the object's position in the ground-plane \mathbb{R}_{XY} depends the position of its detection in the camera-frame. Note that the Gaussian PDF $p(v_i) = G(v_i, 0, V_i)$ is graphically represented by its sub-level-set $\mathcal{E}_{0, V_i} \subset \mathbb{R}_{XY}$, for which an illustrative explanation is found in Figure 1.9.

Before presenting the tracking results of the different DKF solutions, let us point out that the real trajectories of the observed humans are not available. Therefore, apart from the assessed DKF solutions, all detections are acquired by a centralized Kalman filter to obtain an optimal estimate $\hat{x}^{(q)}[k]$ and $P^{(q)}[k]$ that characterizes some kind of “ground truth” trajectory per human q . Then, instead of the estimation error, the estimated position of the centralized Kalman filter is compared to the corresponding position estimated by a particular DKF solution. This is used to compute the average distance error $\Delta \in \mathbb{R}_+$, which is determined per camera $i \in \mathcal{N}$ and per human $q \in \mathbb{Z}_{[1,10]}$ according to the following formula

$$\Delta = \frac{1}{4} \sum_{i=1}^4 \frac{1}{10} \left(\sum_{q=1}^{10} \frac{1}{40} \left(\sum_{k=1}^{40} \sqrt{\eta_i^{(q)}[k]} \right) \right), \quad (6.8)$$

$$\eta_i^{(q)}[k] = (\{\hat{x}^{(q)}[k]\}_1 - \{\hat{x}_i^{(q)}[k]\}_1)^2 + (\{\hat{x}^{(q)}[k]\}_3 - \{\hat{x}_i^{(q)}[k]\}_3)^2.$$

A top-view of the estimated trajectories is depicted in Figure 6.10. Note that the assessed DKF solutions estimate four trajectories per human q , each corresponding to a local estimate $x_i^{(q)}$ at the i -th camera. These four trajectories per human can be compared within one DKF solution and thereby, give an indication to what extent a consensus is achieved on the different local estimates throughout these four cameras. Further, the results of the average distance error between the centralized Kalman filter and a particular DKF solution is presented in Table 6.1.

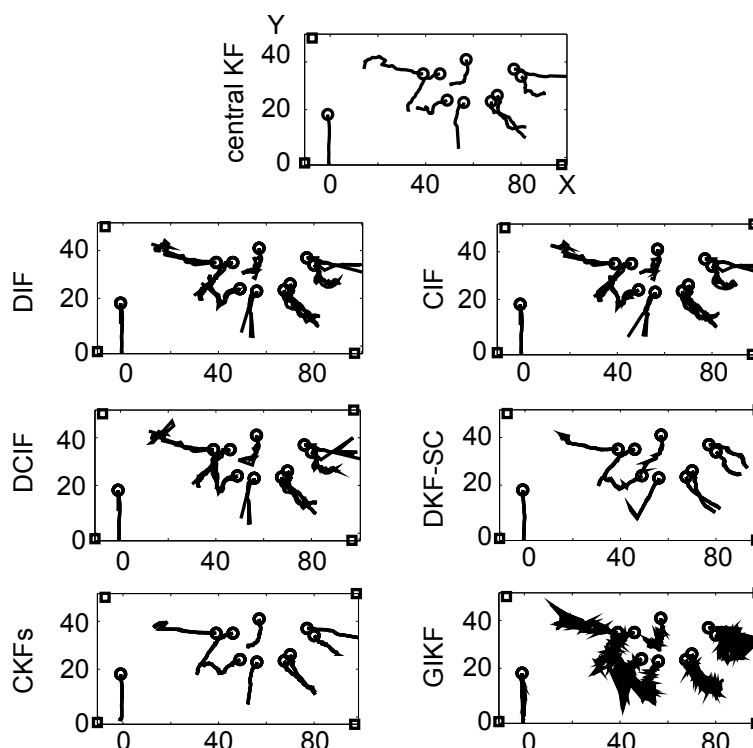


Figure 6.10: The estimated tracks, i.e., trajectories, of the humans in all cameras. The initial position of each human is marked with the symbol \circ , after which four estimated tracks follow per initial human-position. Each track corresponds to one local $x_i^{(q)}$, i.e., an estimated track in camera i associated to human q . The four camera-positions are marked with the symbol \square .

Table 6.1: The average distance Δ [m] of (6.8) between the estimated position of the centralized Kalman filter and one of the assessed DKF methods (per camera, per tracked human and per sample instant).

	DIF	CIF	DCIF	DKF-SC	CKFs	GIKF
δ	0.98	1.17	1.00	2.08	1.16	2.17

The results of the DIF and CIF in Figure 6.10, i.e., when comparing the four local estimation results per human, indicate that a consequence of exchanging local measurements yields a difference in the local estimation results per node (camera). The CIF could solve this aspect by performing multiple averaging cycles, as the current case-study adopted one averaging cycle

$L = 1$. Although this would improve the consensus between the different local estimates, Table 6.1 indicates that the obtained estimation results of the DIF and CIF are already close to the centralized Kalman filter. The main reason is that this application set-up has four cameras. Hence, exchanging local measurements implies that each camera already receives three out of four different local measurements. Therefore, in this small sensor network it is questionable whether exchanging local estimates and thereby, have the potential to obtain more measurement information per local estimate, is beneficial. In fact, Table 6.1 indicates that the other four DKF solutions, which exchange local estimates, do not result in smaller distances between their estimated trajectories and the trajectories obtained by the centralized Kalman filter. However, keep in mind that the centralized Kalman filter does not obtain the true trajectories but an estimate. Nonetheless, Figure 6.10 indicates that consensus between the different local estimates is improved when the DCIF, DKF-SC and the CKFs are employed rather than the DIF and the CIF (even though the CKFs does not aim for a consensus per se). It should be mentioned that the DCIF was very sensitive to its design parameter ϵ , which was set to $\epsilon = 0.1$. Perhaps a smaller value yields a better consensus between the tracks, as there are still quite some difference in consensus between results of the DCIF and the results of the DKF-SC and CKFs. A last observation in this application example is obtained from trajectories estimated by the GIKF. The estimation results of this DKF solution show severe variations. This is mainly induced by the accuracy of local measurements $y_i \in \mathbb{R}_{XY}$ that are quite different per camera i . More precisely, the transformation of a detection from the camera-frame to \mathbb{R}_{XY} results in a very accurate position measurement with respect to the azimuth of the camera but a very poor distance measurement. Hence, local estimates of the GIKF are given large updates with respect to the azimuth of the camera, which differs per sample instant as gossiping implies that local estimates are swapped from one camera to another.

Overall, the results of the assessed DKF solutions indicate that they are comparable in estimation error, with an exception to the DKF-SC and GIKF. Further, note that the CKFs solution computes the eigenvalues of P_i . As such, the CKFs algorithm will require additional computational power compared to the other DKF methods, though a state dimension of 4 implies that it is still in a comparable range. The next application example of a diffusion process simulates a larger network size.

6.5.2 Environmental diffusion process

In this section the DKF methods are assessed on a diffusion process in the presence of wind. To that end, consider an area of 1200×1200 meters containing a chemical source. As time passes, chemical matter spreads across the area due to diffusion and wind. To simulate this spread of chemical matter,

let us divide the area into a grid with a grid-size of 100 meters. The center of each grid-box is defined as a grid-point. The spread of chemical matter is simulated by computing the concentration level $\rho^{(q)} \in \mathbb{R}_+$ at each q -th grid-point. The concentration level $\rho^{(q)}$ [m^{-3}] depends on the corresponding levels at neighboring grid-points, which are denoted as q_n for north, q_s for south, q_e for east and q_w for west. See also Figure 6.11 for a graphical representation of these grid-points relative to the q -th grid-point. More precisely, the *time-continuous* process model of $\rho^{(q)}(t)$, for some $a, a_n, a_s, a_e, a_w \in \mathbb{R}$ and for all $q \in \mathbb{Z}_{[1,144]}$, yields

$$\dot{\rho}^{(q)} = a\rho^{(q)} + a_n\rho^{(q_n)} + a_s\rho^{(q_s)} + a_e\rho^{(q_e)} + a_w\rho^{(q_w)} + u^{(q)}, \quad \forall t \in \mathbb{R}_+$$

The above variable $u^{(q)} \in \mathbb{R}_+$ parameterizes the production of chemical matter at a grid-point q and yields $u^{(18)} = 75$, $u^{(28)} = 75$, $u^{(29)} = 100$, $u^{(30)} = 100$ and $u^{(40)} = 175$ for all $t \in \mathbb{R}_+$, while $u^{(q)} = 0$ for all other $q \in \mathbb{Z}_{[1,144]}$. The remaining parameters are chosen to establish a change in the wind direction from north to north-east-east:

- North: $a = \frac{-12}{800}$, $a_n = \frac{1}{800}$, $a_s = \frac{7}{800}$, $a_e = \frac{2}{800}$ and $a_w = \frac{2}{800}$, for all $t \leq 300$.
- North-east-east: $a = \frac{-10}{800}$, $a_n = \frac{1}{800}$, $a_s = \frac{3}{800}$, $a_e = \frac{1}{800}$ and $a_w = \frac{5}{800}$, for all $t > 300$.

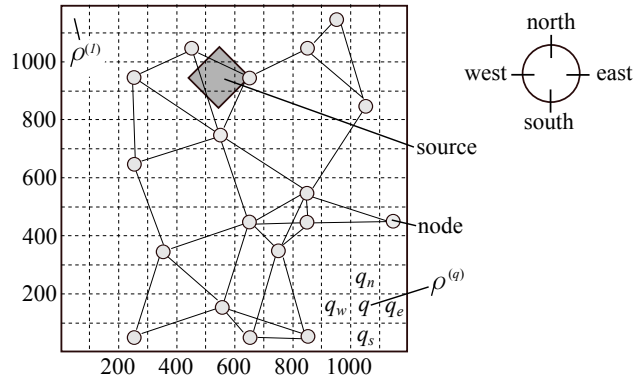


Figure 6.11: The monitored area is divided into a grid. Each grid-point q has four neighbors q_n , q_s , q_e and q_w , i.e., one to the north, south, east and west of grid-point q , respectively. The chemical matter produced by the source spreads through the area due to diffusion and wind.

An ad-hoc sensor network is deployed in the area to reproduce the concentration levels at each grid-point. This means that the assessed DKF solutions are employed for an estimation of the state vector $x[k]$ in each node i with a sampling time of $\tau_s = 10$ seconds, where $x[k] := (\rho^{(1)}[k] \quad \rho^{(2)}[k] \dots \rho^{(144)}[k])^\top$.

Communication The network consists of 18 sensor nodes that are randomly distributed across the area. The communication range per node is 370 meters and the resulting network topology is depicted in Figure 6.11.

Process Neither the wind direction nor values of the chemical source are available at the nodes. Therefore, the assessed DKF solutions employ a simplified version of the diffusion model in *continuous time*, i.e.,

$$\dot{\rho}^{(q)} = \alpha \rho^{(q)} + \alpha_n \rho^{(q_n)} + \alpha_s \rho^{(q_s)} + \alpha_e \rho^{(q_e)} + \alpha_w \rho^{(q_w)} + w^{(q)}, \quad \forall t \in \mathbb{R}_+$$

with $\alpha = \frac{-12}{800}$, $\alpha_n = \frac{3}{800}$, $\alpha_s = \frac{3}{800}$, $\alpha_e = \frac{3}{800}$ and $\alpha_w = \frac{3}{800}$. The unknown source and model uncertainties are represented by the process-noise $w^{(q)} \in \mathbb{R}$, for all $q \in \mathbb{Z}_{[1,144]}$. This noise is characterized by a Gaussian PDF $p(w^{(q)}[k]) = G(w^{(q)}[k], 0, 2 \cdot 10^3)$, for all $k \in \mathbb{Z}_+$ and all $q \in \mathbb{Z}_{[1,144]}$. Information on some of these state elements is provided by the sensor nodes. More precisely, each node i measures the concentration level at its corresponding grid-point, i.e., $y_i[k] = \rho^{(q)}[k] + v_i[k]$, for some $q \in \mathbb{Z}_{[1,144]}$ and $p(v_i[k]) = G(v_i[k], 0, 0.5)$, for all $i \in \mathbb{Z}_{[1,18]}$. The real concentration levels at the instants $t = 300$ and $t = 600$ are illustrated in Figure 6.12.

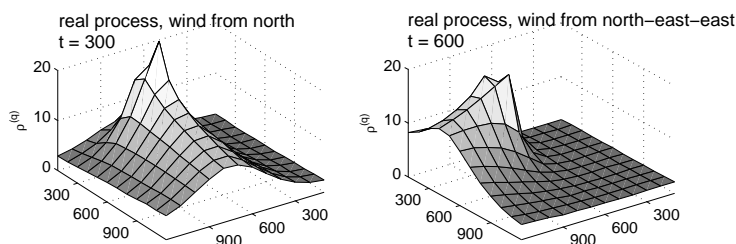


Figure 6.12: The simulated concentration levels at the different grid-points for two instances of the time, i.e., $t = 300$ seconds and $t = 600$ seconds.

The concentration profiles depicted in Figure 6.12 will be used to evaluate the DKF solutions. To that extent, the estimation error in node 1 of the assessed DKF methods is depicted in Figure 6.13, for $t = 300$, and in Figure 6.14, for $t = 600$. In contrast to the first application example, the CIF of this sensor network performs 5 averaging cycles, i.e., $L = 5$. Additionally, Table 6.2 lists the average, squared error of nodes 2 until 18 with respect to node 1, for each assessed DKF method and at a certain time instant t , i.e.,

$$\eta(t) := \frac{1}{17} \sum_{i=2}^{18} (\hat{x}_1(t) - \hat{x}_i(t))^T (\hat{x}_1(t) - \hat{x}_i(t)).$$

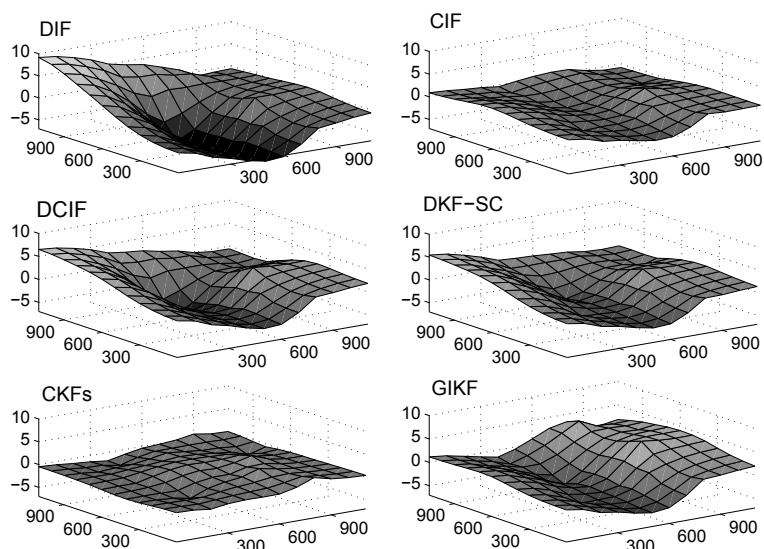


Figure 6.13: The estimation error at each grid-point per DKF solution, for $t = 300$ seconds.

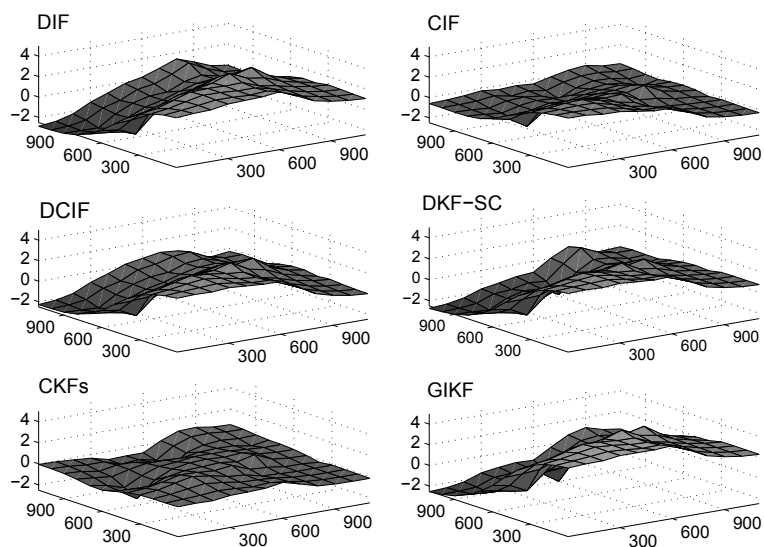


Figure 6.14: The estimation error at each grid-point per DKF solution, for $t = 600$ seconds.

Figure 6.13 and Figure 6.14 indicate that the DIF has the worst performance of all the assessed DKF solutions. Not only in estimation error of

Table 6.2: The average, squared error of the estimated state values at nodes 2 until 18 compared to node 1, which indicates the consensus between the different local estimates

	DIF	CIF	DCIF	DKF-SC	CKFs	GIKF
$\eta(300)$	26.1	0.75	4.80	3.15	0.31	7.12
$\eta(600)$	10.3	0.64	2.11	1.75	0.12	4.44

node 1 but Table 6.2 further implies that the local estimates computed by the DIF show large differences. The fact that the DIF exchanges measurements explains this behavior, i.e., it causes that the local estimate x_i will merely obtain accurate concentration levels in the close region of node i . A similar behavior is prevented in the CIF by performing 5 averaging cycles when synchronizing local measurements. However, a trade-off for preventing this behavior is that it requires 5 times more communication power. Hence, the results of the DIF and CIF show that exchanging measurements in large-scale sensor networks is suboptimal, due to the limitations in communication resources. The four remaining DKF algorithms exchange local estimates. The DCIF and DKF-SC, which synchronize local estimates, have a mutually comparable performance with respect to the previous object tracking application. Also in this example, the DKF-SC has a better consensus than the DCIF, see Table 6.2, but the DCIF has less estimation error compared to the DKF-SC, see Figure 6.13 and Figure 6.14. Yet, notice that the estimation error of node 1 is still high for both methods. This is caused by the synchronization procedure on local estimates. Similar as to the DIF, these DKF solutions establish a local estimate x_i that will merely obtain accurate concentration levels in the close region of node i . The other concentration levels that are not in the close region will be inaccurate. Synchronization then enforces a consensus and balances the accurate estimates with the inaccurate ones. This is reflected in Table 6.2, which indicate that the DCIF and DKF-SC yield a better consensus than the DIF, though it is not the best consensus result that is obtained. Note that the CKFs, which shows the least estimation error in node 1, has the best consensus properties as well. Moreover, node 1 is even capable of estimating the concentration levels at all four corners of the network with a very little estimation errors. The results are even better than the CIF that requires 5 times more communication power. Hence, employing the *ellipsoidal intersection* fusion approach on local estimates tremendously improves the estimation results throughout the entire network. However, this improvement is not attained when local estimates are swapped, i.e., the GIKF. To improve the results of the GIKF one should reduce the sampling time, as local estimates will then manoeuvre through the network with shorter interval times.

Overall, the CKFs shows an exceptional performance for large-scale sensor networks with respect to the other DKF algorithms. Comparable results can be achieved with the CIF, although that requires five times more communication power, which is not desired in sensor networks.

These two real-life inspired case-study conclude the practical assessment of DKF. They were used for indicating the benefits of the difference DKF approaches as well as their drawbacks. Next, let us complete this overview with a discussion on recent extensions to the field of distributed state estimation.

6.6 Extended algorithms

The DKF methods, as they were discussed in previous sections, show that there are four fundamentally different approaches to address distributed estimation. Often, these each of these approaches was extended to allow for a reduced local state or nonlinear processes.

Solutions that reduce local state vectors aim to decrease the computational power of each node. To that end, a node i divides the global state x in two unique parts: one part θ_i that is estimated by node i and another part φ_i that is not estimated by node i . Preferably, different nodes i and j also have different local state vectors θ_i , φ_i , θ_j and φ_j . An extension of the DIF with this property was proposed in (Mutambara and H.F., 2000). Therein, the reduced state vector θ_i is estimated, while φ_i is regarded as process noise. A related method in (Khan and Moura, 2007) proposed an additional communication step of local estimates between the nodes, so that each node i could reconstruct the unknown φ_i by combining the received estimates θ_j from neighboring nodes $j \in \mathcal{N}_{i(1)}$. Notice that this imposes a strict requirement on the network topology. An alternative solution of reduced local state vector regarding synchronization of local estimates was proposed in Stankovic et al. (2009). Therein, each node employs a local process model to update θ_i by exploiting the local measurement y_i . Nodes then exchange these local estimates to establish a synchronized estimate of the *global state* x . In fact, determining a local process model for θ_i from the global one is often based on heuristics and requires further investigations.

Solutions for nonlinear processes are being developed to enable distributed state estimation beyond the Kalman filtering algorithm and thereby, increase the amount of applicable set-ups. One obvious solution is to adopt the DIF approach and replace the Kalman filter (or Information filter) with a particular nonlinear state estimator, such as the extended Kalman filter that was proposed in (Regazonni, 1994). A similar DIF solution for nonlinear processes that included a quantization on the exchanged measurements was presented in (Kar et al., 2008). Further, in (Farina and Scattolini, 2009) a

solution was presented that exchanges local estimates, which are then combined with a moving horizon scheme for distributed state estimation. Also, Section 5.4 studies the effects of nonlinear processes in a distributed state-estimator based on fusion of local estimates. Nonetheless, a clear overview of nonlinear, distributed state estimation is not yet available.

6.7 Conclusions

A comprehensive survey on distributed Kalman filtering solutions that are found across several communities was presented. To that end, four different types of strategies were considered, i.e., fusion or synchronization of local measurements and fusion or synchronization of local estimates. Their differences in communication showed to be negligible, as most methods that exchange local measurements assume the information form when exchanging data, which is of the same size as the local estimate.

Further, a unified description was introduced for comparing the theoretical results of each of the solutions, i.e., the corresponding set of state update formulas followed by an asymptotic analysis. The results indicated that the local estimates throughout the sensor network have different values in case measurements are exchanged. This also includes the main advantage of exchanging local estimates, which is that local measurement information spreads through the entire network. Even under the condition that nodes exchange data only once per sample instant.

In addition, a critical assessment was performed in two real-life inspired sensor network applications. The first application involved object tracking in a parking lot based on four static cameras. For that application, estimation results of the assessed solutions were comparable, except for one gossip based approach. In the second example, the sensor network had a larger network-size for monitoring a diffusion process of a chemical matter. In this application, the cooperative (fusion) approach showed superior performance with respect to the other distributed Kalman filtering solutions.

To summarize, this chapter provides an initial insight and argumentation for choosing a suitable distributed Kalman filtering strategy when deploying a sensor network.

Conclusions

7.1 Contributions

7.2 Ideas for future research

A summary of the main contributions and a collection of several possible directions for future research conclude this thesis.

7.1 Contributions

The research presented in this thesis focusses on state estimation in networked systems. Networked systems can manage large amount of sensor data. However, they often lack the required communication and/or computational resources for processing the excessive quantity of produced measurements in classical implementations, i.e., centralized and synchronous in time. Limitations in communication were addressed by reducing the amount of measurements with *event sampling*, while a *distributed estimation* approach was employed when both communication as well as computational resources are limited. In particular, the key contributions of this thesis are the following:

- Stable state estimation results for (any) event sampling strategy;
- Integration of event based state estimation in a control system;
- Fusion of two estimates with unknown correlations;
- Global covariance as emergent behavior in distributed state estimation;
- A comprehensive overview on distributed Kalman filtering.

Let us present these contributions in more detail.

7.1.1 Exploiting event sampled measurements

Classical implementations of a centralized state estimator require that new measurements are sampled synchronously in time. However, recently, measurement data is exchanged via data connections that are known to have a limited capacity regarding the number of exchanged data packages per second. As such, solutions that reduce the amount of sampled measurements are sought for. Event based sampling has the potential to enable this reduction, as new measurements are sampled when a predefined event occurs

ont he sensor value. However, if event sampling is not accounted for in the estimation strategy, it can cause unstable estimation results. To that extent, the event based state-estimator (EBSE) was developed in Chapter 2 that supports any event sampling strategy and yet attains stable estimation results. Supporting any type of event sampling strategy allows the sensor node to adopt different event criteria depending on, for example, the expected lifetime of its battery. However, it can result in situations that after some time t no event will occur anymore and thus no new measurement will be received by the estimator. To guarantee stability, the EBSE exploits an implicit property of event sampling, i.e., not receiving a new measurement still gives information on the current sensor value. Exploiting this property means that the proposed EBSE performs an update on its estimation results at two different types of sample instants: at event instants, when a new measurement value is received, and at instants synchronously in time, when no measurement is received. In the latter case, the update based on *inherent* knowledge that the measured value lies within a *bounded set*. A derivation to obtain such a bounded set at any given time instant was established from a general mathematical formulation of event sampling. The fact that estimation results of the EBSE are regularly updated in time guarantees stability, even in the situation that no new measurement is received anymore. This was proven by establishing the conditions for an asymptotic bound on the error-covariance of the EBSE. Furthermore, the effectiveness of the EBSE with respect to estimation error and solutions for coping with package loss were demonstrated in an illustrative example of object tracking.

The promising results of this stable EBSE were further assessed in a control system to create a new type of event based controller. Existing event control solutions assume that the entire state is measured with a specific event sampling strategy. The novelty of the event control design in Chapter 3 is that the EBSE processes the event sampled measurements, which are not necessarily equal to the state, so that a time synchronous controller (robust MPC) can be employed to optimize a stabilizing control action based on the estimated state. As a result, the proposed event based controller is suitable for *various* event sampling strategies, while stability of the closed-loop control scheme is *decoupled* from the event triggering criteria. To that extent, an integration procedure was developed that interprets the EBSE results for the employed robust MPC. More precisely, the error-covariance of the EBSE is transformed into explicit polytopic bounds on the estimation error. It was proven that the robust MPC achieves ISS with respect to estimation errors and, moreover, that it optimizes the closed-loop ISS gain. Simulations provide convincing and promising evidence of the potential of the proposed method, though it should be noted that the set-up integrates a *stochastic* EBSE with a *deterministic* MPC. Additional to the integration, a novel event sampling strategy was proposed to obtain a more constant performance of the EBSE and thus of the control system.

7.1.2 Fusion of estimates with unknown correlation

State fusion addresses the problem of merging two prior estimates of the same state into a single fused estimate. Such methods are beneficial in distributed state estimation, for merging the different local estimates of the corresponding sensor network and thereby, improve its estimation results. However, as keeping track of shared data is infeasible in most networks, correlations of the different local estimates will not be available. Hence, the considered state fusion problem cannot assume that the correlation of its prior estimates is known. Current fusion solutions that agree to this assumption are too conservative, in the sense that a reduction in uncertainty (estimation error) after fusion is not attained. Yet, such a reduction is reasonable to insist, as prior information is merged. Therefore, Chapter 4 proposed the state fusion method *ellipsoidal intersection* (EI), in which a reduction in uncertainty after fusion is guaranteed. To that extent, a novel parametrization of the prior estimates is proposed that results in an explicit characterization of correlation. This parametrization involves the introduction of three new estimates, each representing an independent part of the prior estimate. State fusion is then equivalent to merging these three independent estimates, for which algebraic fusion formulas were derived. To assure that the newly introduced estimates are independent, EI determines a worst case scenario by maximizing the derived characterization of the correlation. Further, the guaranteed reduction in uncertainty was illustrated in a pure fusion example.

7.1.3 Cooperation in distributed estimation

Distributed state estimation refers to a collection of estimation strategies, often used in large-scale or ad-hoc sensor networks, where each node performs an estimation algorithm to compute a local estimate of the global state. The main challenges in distributed estimation are to exploit the large amount of local measurements, while at the time be robust to the evidential changes that will be present in sensor networks. Existing solutions impose strict requirements to address these challenges, since they focus on optimizing the estimation results per node *individually*. Instead, the distributed approach of Chapter 5 solves the estimation problem from a network point of view by describing a desired *emergent behavior*, i.e., cooperation, before deriving the corresponding local estimation algorithms. To that extent, a characterization for attaining cooperation was proposed first. This involves a distributed state-estimator that fulfills the *global covariance* property, i.e., the error-covariance of each node in the network is a combination of all error-covariances found across its nodes. Then, a corresponding local estimation algorithm was derived for observing the state of linear processes, labeled as *cooperative Kalman filters*. Therein, each node performs the Kalman filtering algorithm to process its local measurement and thereby, compute a local estimate of the state. These estimates are shared with neighboring nodes as

input a state fusion method, i.e., ellipsoidal intersection of Chapter 4. Ellipsoidal intersection proved to be a vital element for establishing asymptotic bounds on the error covariance of each local estimate, i.e., stability, under realistic conditions. An illustrative example of tracking the leading vehicle in a platoon of scalable size further showed the benefits of this cooperative strategy compared to popular alternative distributed Kalman filters. In addition, nonlinear process models were investigated in a benchmark application of tracking shockwaves on a highway. This analysis showed that even a mixture of local state-estimators among the nodes is achievable, e.g., some nodes perform the extended Kalman filter, while others perform the unscented Kalman filter to process their local measurements. Such a set-up could reduce computational requirements of some nodes, without comprising on the estimation results. Although the proposed approach supports linear as well as nonlinear processes, a large amount of existing solutions already present a distributed implementation of the Kalman filter.

Therefore, a comprehensive survey on distributed Kalman filtering solutions that are found across several communities was presented. The objective is to provide an initial insight and argumentation when choosing a suitable distributed Kalman filtering strategy for a particular sensor network application. To that extent, existing distributed Kalman filters were divided into four types of strategies, depending on what local variables are shared and how they are used for improving the estimation results in the overall network. Differences in communication requirements showed to be negligible, as most method that exchange local measurements assume the Information form when communicating, which is of the same size as the local estimate. Further, the overview presented the theoretical expectations of the different approaches, along with an asymptotic analysis of their estimation results. To that extent, each distributed Kalman filtering approach was presented via a unified description. On top of these theoretical results, a critical assessment was performed in two real-life inspired sensor networks to fulfill the main objective. The first set-up considered an object tracking application based on four static cameras. The estimation results of all assessed strategies were comparable, except for one gossip based approach. In the second example the sensor network was larger in size and a diffusion process of chemical matter was to be followed. For this set-up, the cooperative Kalman filters showed superior performance compared to the other distributed Kalman filtering solutions, though it also requires additional computational power.

7.2 *Ideas for future research*

Based on the ideas presented in this thesis, several directions for future research from the author's point of view are presented, next.

7.2.1 Event based estimation and control

Current research on estimation with event sampled measurements focusses on minimizing the number of samples, while attaining a particular performance of the estimator. The idea of an event based state-estimator suitable for any event sampling strategy was, up until now, not covered by current literature and can thus provide new insights for dealing with event based sampling. For example, in event based control. To that extent, note that the proposed event based state-estimator supports a control algorithm that runs synchronously in time and that it decouples the event-triggering procedure from closed-loop stability. An example of such a control set-up was studied in Chapter 3 and integrated a *stochastic* state-estimator with a *deterministic* controller. Although the presented results were convincing, other integration procedures, or even estimation and control combinations, could be promising as well

7.2.2 Fusion of estimates

The proposed method on cooperative state estimation, presented in Chapter 5, is based on fusing the local estimate of a node with the estimates received from neighboring nodes. The proposed method for fusing more than two estimates adopts a recursive fusion procedure, i.e., the local estimate is merged with the estimate that is received first, after which their fusion result is further merged with the estimate that is received next, and so on. Note that the order of arrival of neighboring estimates could influence its outcome. Hence, an open question is what the order of the neighboring estimates should be, so to minimize the estimation error of the resulting fused estimate.

7.2.3 Cooperative estimation of reduced states

The distributed solutions for state estimation that were presented in this thesis assume that each node in the network computes a local estimate of the global state. In Section 6.6 some methods were presented that do not require this assumption. Instead, they propose a different distributed solution, where each node computes a part of the global state. As such, The global state is distributed among the nodes, possibly with overlapping state elements between some nodes in the network. Notice that such a set-up is more in line with ant colonies, as individual ants will not have a global view of the surroundings of their colony. Furthermore, such a distributed estimation approach results in a more efficient usage of the resources, as each node estimates those state elements that are relevant to its purpose. Establishing such a set-up for a network of cooperative state-estimators, as proposed in Chapter 5, requires a localized state-space model and a state fusion method that can cope with unequal local state vectors. Especially the issue on state fusion is not yet found in literature and offers opportunities for future research.

Bibliography

- Aggoun, L., Elliot, R., 2004. *Measure Theory and Filtering*. Cambridge University Press.
- Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E., 2002. *Wireless Sensor Networks: a survey*. Elsevier, *Computer Networks* 38, 393–422.
- Alessio, A., Lazar, M., Bemporad, A., Heemels, W. P. M. H., 2007. Squaring the circle: An algorithm for obtaining polyhedral invariant sets from ellipsoidal ones. *Automatica* 43 (12), 2096–2103.
- Alriksson, P., Rantzer, A., 2006. Distributed Kalman filter using weighted averaging. In: 17-th Int. Symp. on Mathematical Theory of Networks and Systems. Kyoto, Japan.
- Alriksson, P., Rantzer, A., 2007. Experimental evaluation of a distributed Kalman filter algorithm. In: 46-th IEEE Conf. on Decision and Control. New Orleans, USA, pp. 5499 – 5504.
- Anderson, B. D. O., Moore, J. B., 1979. *Optimal filtering*. Prentice-Hall.
- Åström, K., Bernhardsson, B., 2002. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In: 41-st IEEE Conf. on Decision and Control. Las Vegas, USA, pp. 2011 – 2016.
- Åström, K. J., 2008. Event based control. In: *Analysis and Design of Nonlinear Control Systems: In Honor of Alberto Isidori*. eds. A. Astolfi and L. Marconi, Springer-Verlag, pp. 127–147.
- Bar-Shalom, Y., Campo, L., 1986. The effect of the common process noise on the two-sensor fused-track covariance. *IEEE Trans. on Aerospace and Electronic Systems* AES-22 (6), 803–805.
- Bar-Shalom, Y., Li, R., 1995. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS.
- Belmega, E., Lasaulce, S., Debbah, M., 2009. A trace inequality for positive definite matrices. *Journal of inequalities in pure and applied mathematics* 10 (1), Article 5.
- Benaskeur, A. R., 2002. Consistent fusion of correlated data sources. In: *IEEE 28-th Ann. Conf. of the Industrial Electronics Society*. Sevilla, Spain, pp. 2652–2656.
- Bernstein, D., 2005. *Matrix Mathematics*. Princeton University Press.

- Bhatia, R., 1997. Matrix Analysis. Springer-Verlag, New York, Inc.
- Boyd, S., Vandenberghe, L., 2004. Convex optimization. Cambridge University Press.
- Brockett, R. W., Liberzon, D., 2000. Quantized feedback stabilization of linear systems. *IEEE Trans. on Automatic Control* 47 (7), 1279–1289.
- Carli, R., Chiuso, A., Schenato, L., Zampieri, S., 2008. Distributed Kalman filtering based on consensus strategies. *IEEE journal in selected areas in communications* 26 (4), 622–633.
- Casbeer, D. W., Beard, R., 2009. Distributed information filtering using consensus filters. In: American Control Conf. St. Louis, USA, pp. 1882 – 1887.
- Chan, S. W., Goodwin, G. C., Sin, K. S., 1984. Convergence properties of the Riccati difference equation in optimal filtering of nonstabilizable systems. *IEEE Trans. on Automatic Control* 29, 110–118.
- Chen, L., Arambel, P., Mehra, R., 2002. Fusion under unknown correlation: Covariance intersection as a special case. In: 5-th IEEE Int. Conf. on Information Fusion. pp. 905–912.
- Chong, C. Y., Kumar, S. P., 2003. Sensor networks: Evolution, opportunities and challenges. In: *Proc. of the IEEE*. Vol. 91(8). pp. 1247–1256.
- Cogill, R., 2009. Event-based control using quadratic approximate value functions. In: 48-th IEEE Conf. on Decision and Control. Shanghai, China, pp. 5883 – 5888.
- Cogill, R., Lall, S., Hespanha, J. P., 2007. A constant factor approximation algorithm for optimal estimation subject to communication costs. In: American Control Conf. New York, NY, pp. 305 – 311.
- Cortés, J., 2009. Distributed krigged Kalman filter for spatial estimation. *IEEE Trans. on Automatic Control* 54 (12), 2816–2827.
- Cover, T. M., Thomas, J. A., 1991. Elements of information theory. Wiley-Interscience, New York.
- Curry, R., 1970. Estimation and control with quantized measurements. MIT Press.
- D’Antona, G., Monti, A., Ponci, F., Rocca, L., 2006. A distributed state estimator for electric power systems in avionic and naval applications. In: Instrumentation and Measurement Technology Conf. Sorrento, Italy, pp. 2312 – 2316.

- Di Cairano, S., Bemporad, A., Caldelli, A., 2007. Moving target detection and tracking in wireless sensor networks. In: European Control Conf. Kos, Greece, pp. 2218–2223.
- Dimarogonas, D. V., Johansson, K. H., 2009. Event-triggered control for multi-agent systems. In: 48-th IEEE Conf. on Decision and Control. Shanghai, China, pp. 7131 – 7136.
- Donkers, M. C. F., 2012. Networked and event-triggered control systems. Ph.D. thesis, Eindhoven University of Technology.
- Durant-Whyte, H., Rao, B., Hu, H., 1990. Towards a fully decentralized architecture for multi-sensor data fusion. In: 1990 IEEE Int. Conf. on Robotics and Automation. Cincinnati, USA, pp. 1331–1336.
- Edwards, A. L., 1979. Multiple regression and the analysis of variance and covariance. W. H. Freeman & Co Ltd.
- Farina, M., Ferrari-Trecate, G., Scattolini, R., 2009. A moving horizon scheme for distributed state estimation. In: 48-th IEEE Conf. on Decision and Control. Shanghai, China, pp. 7036 – 7042.
- Felter, S., 1990. An overview of decentralized Kalman filters. In: IEEE 1990 Southern Tier Technical Conf. Birmingham, USA, pp. 79–87.
- Franken, D., Hupper, A., 2005. Improved fast covariance intersection for distributed data fusion. In: 8-th Int. Conf. on Information Fusion. Philadelphia, PA, USA, pp. WbA23:1–7.
- Gales, M. J. F., Airey, S. S., 2006. Product of Gaussians for speech recognition. *Comput. Speech Lang.* 20, 22–40.
- Garin, F., Schenato, L., 2011. Networked Control Systems. Vol. 406 of Lecture Notes in Control and Information Sciences. Springer, Ch. A survey on distributed estimation and control applications using linear consensus algorithms, pp. 75–107.
- Grewal, M. S., Andrews, A. P., 1993. Kalman filtering: Theory and practise. Rootledge.
- Hanebeck, U. D., Briechle, K., Horn, J., 2001. A tight bound for the joint covariance of two random vectors with unknown but constrained cross-correlation. In: IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems. Baden-Baden, Germany, pp. 85–90.
- Hashmipour, H., Roy, S., Laub, A., 1988. Decentralized structures for parallel Kalman filtering. *IEEE Trans. on Automatic Control* 33 (1), 88–93.

- Hassan, M., Salut, G., Singh, M., Titli, A., 1978. A decentralized algorithm for the global Kalman filter. *IEEE Trans. on Automatic Control* 23 (2), 262–267.
- Hasu, V., Koivo, H., 2006. Decentralized Kalman filter in wireless sensor networks - case studies. In: *Advances in Computer, Information, and Systems Sciences, and Engineering*. Springer, The Netherlands, pp. 61–68.
- Heemels, W. P. M. H., Gorter, R. J. A., van Zijl, A., van den Bosch, P. P. J., Weiland, S., Hendrix, W. H. A., Vonder, M. R., 1999. Asynchronous measurement and control: A case study on motor synchronization. *Control Engineering Practice* 7, 1467–1482.
- Heemels, W. P. M. H., Sandee, J. H., van den Bosch, P. P. J., 2008. Analysis of event-driven controllers for linear systems. *Int. Journal of Control* 81 (4), 571–590.
- Hegyi, A., De Schutter, B., Hellendoorn, H., Jun. 2005. Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C* 13 (3), 185–209.
- Henningsson, T., Johannesson, E., Cervin, A., 2008. Sporadic event-based control of first-order linear stochastic systems. *Automatica* 44 (11), 2890–2895.
- Hespanha, J., Naghshtabrizi, P., Xu, Y., 2007. A survey of recent results in networked control systems. In: *Proc. of the IEEE*. Vol. 95(1). pp. 138 – 162.
- Imer, O., Basar, T., 2005. Optimal estimation with limited measurements. In: *44-th IEEE Conf. on Decision and Control*. Seville, Spain, pp. 1029 – 1034.
- Imer, O., Basar, T., 2006. Optimal control with limited controls. In: *American Control Conf. Minneapolis, USA*, pp. 298 – 303.
- Jadbabaie, A., Lin, J., Morse, A., 2003. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on Automatic Control* 48 (6), 988–1001.
- Julier, S., 2009. Estimating and exploiting the degree of independent information in distributed data fusion. In: *12-th Int. Conf. on Information Fusion*. Seattle, USA, pp. 772–779.
- Julier, S. J., Uhlmann, J. K., 1997a. A new extension of the Kalman filter to nonlinear systems. In: *AeroSense: The 11-th Int. Symp. on Aerospace/Defense Sensing, Simulation and Controls*. Orlando, Florida, pp. 182–193.

- Julier, S. J., Uhlmann, J. K., 1997b. A non-divergent estimation algorithm in the presence of unknown correlations. In: American Control Conf. Piscataway, USA, pp. 2369–2373.
- Kahn, J. M., Katz, R. H., Pister, K. S. J., 1999. Next century challenges: Mobile networking for “smart dust”. In: 5-th ACM/IEEE Int. Conf. on Mobile Computing and Networking. Seattle, USA, pp. 271 – 278.
- Kalman, R., 1960. A new approach to linear filtering and prediction problems. Trans. of the ASME Journal of Basic Engineering 82 (D), 35–42.
- Kamgarpour, M., Tomlin, C., 2008. Convergence properties of a decentralized Kalman filter. In: 47-th IEEE Conf. on Decision and Control. Cancun, Mexico, pp. 3205–3210.
- Kar, S. Moura, J., 2011. Gossip and distributed Kalman filtering: Weak consensus under weak detectability. IEEE Trans. on Signal Processing 59 (4), 1766 – 1784.
- Kar, S., Moura, J. M. F., Ramanan, K., 2008. Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication. Computing Research Repository abs/0809.0009v1.
- Kellett, C. M., Teel, A. R., 2005. On the robustness of \mathcal{KL} -stability for difference inclusions: Smooth discrete-time Lyapunov functions. SIAM Journal on Control and Optimization 44 (3), 777–800.
- Khan, U., Moura, J., 2007. Distributed Kalman filters in sensor networks: Bipartite fusion graphs. In: IEEE 14-th Workshop on Statistical Signal Processing. Madison, USA, pp. 700–704.
- Kirti, S., Scaglione, A., 2008. Scalable distributed Kalman filtering through consensus. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing. Las Vegas, USA, pp. 2725 – 2728.
- Kofman, E., Braslavsky, J. H., 2006. Level crossing sampling in feedback stabilization under data-rate constraints. In: IEEE Conf. on Decision and Control. San Diego, California, pp. 4423–4428.
- Kotecha, J., Djurić, P., 2003. Gaussian sum particle filtering. IEEE Trans. Signal Processing 51 (10), 2602–2612.
- Kullback, L., Leibler, R. A., 1951. On information and sufficiency. Annals of Mathematical Statistics 22, 79–86.
- Kurzhanskiy, A. A., Varaiya, P., May 2006. Ellipsoidal toolbox. Tech. Rep. UCB/EECS-2006-46, University of California, Berkeley.

- Lazar, M., Heemels, W. P. M. H., 2008. Optimized input-to-state stabilization of discrete-time nonlinear systems with bounded inputs. In: American Control Conf. Seattle, Washington, pp. 2310–2315.
- Lebesgue, H. L., 1902. Integrale, longueur, aire. Ph.D. thesis, University of Nancy.
- Lehmann, D., Lunze, J., 2009. Event-based control: A state-feedback approach. In: European Control Conf. Budapest, Hungary, pp. 1717–1721.
- Lehmann, D., Lunze, J., 2010. A state-feedback approach to event-based control. *Automatica* 46, 211–215.
- Lewis, F. L., 2005. Wireless sensor networks. In: Smart environments: Technologies, protocols, applications. Lecture Notes in Computer Science. Wiley, New York, pp. – Chapter 2 –.
- M. Rabi, G. M., Baras, J., 2006. Multiple sampling for estimation on a finite horizon. In: 45-th IEEE Conf. on Decision and Control. San Diego, USA, pp. 1351 – 1357.
- Mahler, R., 2011. General Bayes filtering of quantized measurements. In: 14-th Int. Conf. on Information Fusion. Chicago, USA, pp. 346 – 352.
- Majda, A., Kleeman, R., Cai, D., 2002. A mathematical framework for quantifying predictability through relative entropy. *Methods and Applications of Analysis* 9 (3), 425–444.
- Mallick, M., Coraluppi, S., Carthel, C., 2001. Advances in asynchronous and decentralized estimation. In: IEEE Aerospace Conf. Big Sky, USA, pp. 1873 – 1888.
- Marck, J. W., Kester, L., Bergmans, J., Iersel, M. v., Foeken, E. v., 2008. Effective and efficient communication of information. In: IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems. Seoul, Korea, pp. 315 – 320.
- Marck, J. W., Sijs, J., 2010. Relevant sampling applied to event-based state-estimation. In: 4-th Int. Conf. on Sensor Technologies and Applications. Venice, Italy, pp. 618–624.
- Mardia, K. V., Kent, J. T., Bibby, J. M., 1979. Multivariate analysis. Academic press, London.
- Miskowicz, M., 2006. Send-on-delta concept: An event-based data-reporting strategy. *Sensors* 6, 49–63.
- Miskowicz, M., 2007. Asymptotic effectiveness of the event-based sampling according to the integral criterion. *Sensors* 7, 16–37.

- Mo, Y., Sinopoli, B., 2008. A characterization of the critical value for Kalman filtering with intermittent observations. In: 47-th IEEE Conf. on Decision and Control. Cancun, Mexico, pp. 2692–2697.
- Montgomery, D., Runger, G., 2007. Applied statistics and probability for engineers. John Wiley and Sons.
- Mosquera, C., Jayaweera, S. K., 2008. Entangled Kalman filters for cooperative estimation. In: 5-th IEEE Sensor Array and Multichannel Signal Processing Workshop. Darmstadt, Germany, pp. 2725 – 2728.
- Mutambara, A., H.F., D.-W., 2000. Fully decentralized estimation and control for a modular wheeled mobile robot. *Int. Journal of Robotic Research* 19 (6), 582–596.
- Nguyen, V., Suh, Y., 2007. Improving estimation performance in networked control systems applying the send-on-delta transmission method. *Sensors* 7, 2128–2138.
- Niehse, W., 2002. Information fusion based on fast covariance intersection filtering. In: 5-th Int. Conf. on Information Fusion. Annapolis, USA, pp. 901–905.
- Olfati-Saber, R., 2007. Distributed Kalman filtering for sensor networks. In: 46-th IEEE Conf. on Decision and Control. New Orleans, USA, pp. 5492 – 5498.
- Olfati-Saber, R., 2009. Kalman-consensus filter: Optimality, stability, and performance. In: 48-th IEEE Conf. on Decision and Control. Shanghai, China, pp. 7036 – 7042.
- Olfati-Saber, R., Shamma, J. S., 2005. Consensus filters for sensor networks and distributed sensor fusion. In: 44-th IEEE Conf. on Decision and Control 2005 and 2005 European Control Conf. Seville, Spain, pp. 6698 – 6703.
- Papp, Z., Sijs, J., Lagioia, M., 2009. Sensor network for real-time vehicle tracking on road networks. In: 5-th Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing. Melbourne, Australia, pp. 85 – 90.
- Regazonni, C. S., 1994. Distributed extended Kalman filtering network for estimation and tracking of multiple objects. *Electronics Letters* 30 (15), 1202–1203.
- Regazzoni, C. S., Tesei, A., 1996. Distributed data fusion for real-time crowding estimation. *Signal Processing* 53, 47 – 63.
- Ren, W., Beard, R., Kingston, D., 2005. Multi-agent Kalman consensus with relative uncertainty. In: American Control Conf. Portland, USA, pp. 1865 – 1870.

- Ribeiro, A., Giannakis, G. B., Roumeliotis, S. I., 2006. SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations. *IEEE Trans. on Signal Processing* 54 (12), 4782 – 4795.
- Ribeiro, A., Schizas, I. D., Roumeliotis, S. I., Giannakis, G. B., 2010. Kalman filtering in wireless sensor networks: Reducing communication cost in state-estimation problems. *IEEE Control Systems Magazine* 4, 66–86.
- Ristic, B., Arulampalam, S., Gordon, N., 2004. *Beyond the Kalman filter: Particle filter for tracking applications*. Artech House.
- Roy, S., Hashemi, R., Laub, A., 1991. Square root parallel Kalman filtering using reduced order local filters. *IEEE Trans. on Aerospace and Electronic Systems* 27 (2), 276–289.
- Sawo, F., Beutler, F., Hanebeck, U., 2008. Decentralized state estimation of distributed phenomena based on covariance bounds. In: 17-th IFAC World Congress. Seoul, Korea, pp. 9014 – 9020.
- Schlosser, M. S., Kroschel, K., 2007. Performance analysis of decentralized Kalman filters under communication constraints. *Journal of Advances in Information Fusion* 2 (2), 65 – 76.
- Shannon, C. E., Weaver, W., 1949. *The mathematical theory of communication*. The University of Illinois Press, Urbana, Illinois.
- Sijs, J., Lazar, M., 2009. On event-based state estimation. In: *Hybrid Systems: Computation and Control*. Vol. 5469 of *Lecture Notes in Computer Science*. Springer Verlag, pp. 336–350.
- Sijs, J., Lazar, M., 2011a. Distributed Kalman filtering with global covariance. In: *American Control Conf.* San Francisco, USA, pp. 4840 – 4845.
- Sijs, J., Lazar, M., 2011b. Empirical case-studies of state fusion via ellipsoidal intersection. In: 14-th Int. Conf. on Information Fusion. Chicago, USA, pp. 1709 – 1716.
- Sijs, J., Lazar, M., 2012a. Event based state estimation with time synchronous updates. *IEEE Trans. on Automatic Control* *in press*.
- Sijs, J., Lazar, M., 2012b. State fusion with unknown correlation: Ellipsoidal intersection. *Automatica* *in press*.
- Sijs, J., Lazar, M., Heemels, W., 2010a. On integration of event-based estimation and robust MPC in a feedback loop. In: 13-th ACM Int. Conf. on Hybrid Systems: Computation and Control. pp. 31–40.

- Sijs, J., Lazar, M., Van de Bosch, P., Papp, Z., 2008. An overview of non-centralized Kalman filters. In: IEEE Int. Conf. on Control Applications. San Antonio, USA, pp. 739–744.
- Sijs, J., Lazar, M., v.d. Bosch, P., 2010b. State fusion with unknown correlation: Ellipsoidal intersection. In: American Control Conf. Baltimore, USA, pp. 3992 – 3997.
- Sijs, J., Papp, Z., 2012. Intelligent Sensor Networks: Across Sensing, Signal Processing, and Machine Learning. Chapter: Self organizing state estimation for sensor networks. Publisher: Taylor & Francis LLC, CRC Press *in press*.
- Sijs, J., Papp, Z., Booij, P., 2011. Nonlinear estimation with a network of heterogenous algorithms. In: 14-th Int. Conf. on Networking, Sensing and Control. Delft, The Netherlands, pp. 433 – 438.
- Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M., Sastry, S., 2004. Kalman filter with intermittent observations. IEEE Trans. on Automatic Control 49, 1453–1464.
- Solodovnikov, V., 1960. Statistical dynamics of linear automatic control systems. Soviet state publishing house for physico-mathematical literature.
- Sorenson, H. W., Alspach, D. L., 1971. Recursive Bayesian estimation using Gaussian sums. Automatica 7, 465–479.
- Speranzon, A., Fischione, C., Johansson, K. H., Sangiovanni-Vincentelli, A., 2008. A distributed minimum variance estimator for sensor networks. IEEE Journal on Selected Areas in Communications 26 (4), 609–621.
- Speyer, J., 1979. Computation and transmission requirements for a decentralized Linear-Quadratic-Gaussian control problem. IEEE Trans. on Automatic Control 24 (2), 266–269.
- Stankovic, S. S., Stankovic, M. S., Stipanovic, A. M., 2009. Consensus based overlapping decentralized estimator. IEEE Trans. on Automatic Control 54 (2), 410–415.
- Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., Estrin, D., 2004. Habitat monitoring with sensor networks. ACM Communications 47, 34–40.
- Tabuada, P., 2007. Event-triggered real-time scheduling for stabilizing control tasks. IEEE Trans. on Automatic Control 52, 1680–1685.
- Tahbaz Salehi, A., Jadbabaie, A., 2010. Consensus over ergodic stationary graph processes. IEEE Trans. on Automatic Control 55, 225–230.

- Vadigepalli, R., Doyle, F. J., 2003. A distributed state estimation and control algorithm for plantwide processes. *IEEE Trans. on Control Systems Technology* 11 (1), 119 – 127.
- van Arem, B., van Driel, C., Visser, R., 2006. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Trans. on Intelligent Transportation Systems* 7, 429–436.
- Vashentsev, A. Y., 2004. External ellipsoidal estimation of the union of two concentric ellipsoids and its applications. *Computational Mathematics and Modelling* 15 (2), 110–122.
- Wang, X., Lemmon, M., 2009. Self-triggered feedback control systems with finite-gain L_2 stability. *IEEE Trans. on Automatic Control* 45, 452–467.
- Wang, Y., Li, X., 2009. A fast and fault-tolerant convex combination fusion algorithm under unknown cross-correlations. In: 12-th Int. Conf. on Information Fusion. Seattle, USA, pp. 571–578.
- Welch, G., Bishop, G., 1995. An introduction to the Kalman filter.
- Wikipedia, Ants. Ant, source: <http://en.wikipedia.org/wiki/ant>.
- Woodbury, M., 1950. Inverting modified matrices. Memo. rep. 4.2, Statistical Res. Group, Princeton.
- Xiao, L., Boyd, S., 2004. Fast linear iterations for distributed averaging. *Systems and Control Letters* 53 (1), 65–78.
- Xiao, L., Boyd, S., Lall, S., 2005. A scheme for robust distributed sensor fusion based on average consensus. In: 4-th Int. Symp. on Information processing in sensor networks. Los Angeles, California, USA, pp. 63 – 70.
- Xu, Y., Hespanha, J., 2004. Optimal communication logics for networked control systems. In: 43-rd IEEE Conf. on Decision and Control. Paradise Island, Bahamas, pp. 3527 – 3532.
- Zhuo, Y., Li, J., 2008. Data fusion of unknown correlations using internal ellipsoidal approximations. In: 17-th IFAC World Congress. Seoul, Korea, pp. 2856–2860.

\mathcal{A}

Matrix properties

- Suppose that $A, B, C, D \succeq 0$. If $A \preceq B$ and $B \preceq C$, then $A \preceq C$ (Proposition 8.1.1 (Bernstein, 2005)). If $A \preceq B$ and $C \preceq D$, then $A + C \preceq B + D$ (Proposition 8.1.2 (Bernstein, 2005)).
- Suppose that $A \succeq 0$. Then it holds that $\lambda_{\min}(A)I \preceq A \preceq \lambda_{\max}(A)I$ (Corollary 8.4.2 (Bernstein, 2005)).
- Suppose that $A, B \succeq 0$. If $A \preceq B$, then $CAC^\top \preceq CBC^\top$ for any C of suitable size (Proposition 8.1.2 (Bernstein, 2005)).
- Suppose that $A, B \succ 0$. If $A \preceq B$, then $A^{-1} \succeq B^{-1}$ (Proposition 8.5.5 (Bernstein, 2005)).
- Suppose that A and B are symmetric matrices. Then it holds that $\lambda_{\max}(A + B) \leq \lambda_{\max}(A) + \lambda_{\max}(B)$ (Corollary III.2.3 (Bhatia, 1997)).
- Suppose that $A \succ 0$. Then $\lambda_q(A) = \sigma_q(A)$ (p.5 (Bhatia, 1997)).
- Suppose $A \succeq 0$ and let $\alpha \in \mathbb{R}$. Then it holds that $\lambda_q(\alpha A) = \alpha \lambda_q(A)$ (In case one can find a solution for the equality $(\lambda I - A)x = 0$ for any $\|x\|_2 = 1$, then the corresponding λ and x are defined as an eigenvalue with corresponding eigenvector of A respectively. Therefore if $\nu_q(A)$ is defined as the q -th eigenvector of A then $(\lambda_q(A)I - A)\nu_q(A) = 0 \Rightarrow (\alpha \lambda_q(A)I - \alpha A)\nu_q(A) = 0$).
- If A is symmetric, then CAC^\top is also symmetric for any C of suitable size.
- For any given matrices A, B of suitable size, it holds that $\sigma_{\max}(AB) \leq \sigma_{\max}(A)\sigma_{\max}(B)$ (Theorem III.4.5 (Bhatia, 1997)).

B

Proofs corresponding to Chapter 2

B.1 Proof of Lemma 2.4.2

Statement lemma:

The proposed EBSE satisfies $P[k] \preceq \varsigma[k]\Theta[k]$, for all $t_k \in \mathbb{T}$.

The following result is used in this proof of Lemma 2.4.2, for which

$$\begin{aligned}\hat{y}[k] &:= \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} \hat{y}_q[k], \\ Y[k] &:= \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} ((\hat{y}[k] - \hat{y}_q[k])(\hat{y}[k] - \hat{y}_q[k])^\top).\end{aligned}\tag{B.1}$$

Lemma B.1.1 For all synchronous instants $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$ there exists a $m[k] := (\hat{y}^-[k] - \hat{y}^+[k])^\top (\hat{y}^-[k] - \hat{y}^+[k]) \lambda_{\min}^{-1}(R[k])$, such that $Y[k] \preceq m[k]R[k]$.

The proof of Lemma B.1.1 is found in Section B.1.1. Further, let us recall the approximation of $P[k]$ in the EBSE algorithm in Section 2.3.3, i.e.,

$$P[k] = \Theta[k] + \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} ((\hat{x}[k] - \hat{\theta}_q[k])(\hat{x}[k] - \hat{\theta}_q[k])^\top).\tag{B.2}$$

Combining the expressions of $\hat{x}[k]$ and $\hat{\theta}_q$, as presented in the same algorithm, with the above $\hat{y}[k]$, one can derive that $\hat{x}[k] - \hat{\theta}_q[k] = \Theta[k]C^\top R^{-1}[k](\hat{y}[k] - \hat{y}_q[k])$. Substituting this expression into (B.2), while taking the above definition of $Y[k]$ into account, implies that (B.2) can be rewritten as follows:

$$P[k] = \Theta[k] + \Theta[k]C^\top R^{-1}[k]Y[k]R^{-1}[k]C\Theta[k].\tag{B.3}$$

Substituting $m[k]$ into the definition of $\varsigma[k]$ given in (2.4.1), implies that $\varsigma[k] = 1$ if $t_k \in \mathbb{T}_e$ and $\varsigma[k] = m[k] + 1$ if $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$. Lemma 2.4.2 is thus proven by considering each of these sample instants separately.

Let us start with an event instant $t_k \in \mathbb{T}_e$, for which $P[k] \preceq \Theta[k]$ should hold. Notice that at these instants the EBSE receives a measurement, yielding $N = 1$ and $\hat{y}_1[k] = y[k]$, which further implies that $Y[k] = 0$. Substituting $Y[k] = 0$ into (B.3) directly results in the first inequality, i.e.,

$$P[k] = \Theta[k], \quad \forall t_k \in \mathbb{T}_e.\tag{B.4}$$

If $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$, i.e., a time-synchronous instant, then $P[k] \preceq (m[k]+1)\Theta[k]$ should hold. The result of Lemma B.1.1, i.e., $Y[k] \preceq m[k]R[k]$, gives that $C^\top R^{-1}[k]Y[k]R^{-1}[k]C \preceq C^\top R^{-1}[k](m[k]R[k])R^{-1}[k]C$ and thus

$$\Theta[k] + \Theta[k]C^\top R^{-1}[k]Y[k]R^{-1}[k]C\Theta[k] \preceq \Theta[k] + \Theta[k](m[k]C^\top R^{-1}[k]C)\Theta[k].$$

Since the left hand side of the above inequality is equal to $P[k]$ in (B.3), one obtains that $P[k] \preceq \Theta[k](\Theta^{-1}[k] + m[k]C^\top R^{-1}[k]C)\Theta[k]$, which after substituting $\Theta^{-1}[k] = P^{-1}[k^-] + C^\top R^{-1}C$, see (2.16b), yields

$$P[k] \preceq \Theta[k](P^{-1}[k^-] + (m[k]+1)C^\top R^{-1}[k]C)\Theta[k] \quad (\text{B.5a})$$

$$\preceq (m[k]+1)\Theta[k](P^{-1}[k^-] + C^\top R^{-1}[k]C)\Theta[k], \quad (\text{B.5b})$$

$$= (m[k]+1)\Theta[k], \quad \forall t_k \in \mathbb{T}_s \setminus \mathbb{T}_e, \quad (\text{B.5c})$$

as $m[k] \in \mathbb{R}_{>0}$. Notice that (B.5c) together with (B.4) complete the proof. \square

B.1.1 Proof of Lemma B.1.1

Notice that the lemma considers synchronous instants $t_k \in \mathbb{T}_s \setminus \mathbb{T}_e$, at which $\mathcal{H}[e|t_k]$ is given and thus also the values of $\hat{y}_q[k]$, for all $q \in \mathbb{Z}_{[1,N]}$, $R[k]$, $\hat{y}^-[k]$ and $\hat{y}^+[k]$. The inequality $Y[k] \preceq m[k]R[k]$ is proven in two steps. Firstly, a bound on the covariance $Y[k]$ is given in terms of $\hat{y}^-[k]$ and $\hat{y}^+[k]$ and secondly, the connection between $R[k]$ and $m[k]$ is derived.

Fact 8.7.38 of (Bernstein, 2005) gives that $yy^\top \preceq y^\top y I$ holds for any $y \in \mathbb{R}^n$. Hence, $Y[k] = \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} ((\hat{y}[k] - \hat{y}_q[k])(\hat{y}[k] - \hat{y}_q[k])^\top)$ that was defined in (B.1) satisfies

$$Y[k] \preceq \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} ((\hat{y}[k] - \hat{y}_q[k])^\top (\hat{y}[k] - \hat{y}_q[k]))I. \quad (\text{B.6})$$

The vector $\hat{y}[k]$ is constructed from a linear combination of all the vectors within $\cup_{q \in \mathbb{Z}_{[1,N]}} \{\hat{y}_q[k]\}$, see also (B.1). As a result, the Euclidian distance $\|\hat{y}[k] - \hat{y}_q[k]\|_2 = \sqrt{(\hat{y}[k] - \hat{y}_q[k])^\top (\hat{y}[k] - \hat{y}_q[k])}$ is always less or equal than $\|\hat{y}^-[k] - \hat{y}^+[k]\|_2$, since $\|\hat{y}^-[k] - \hat{y}^+[k]\|_2$ is always larger than the Euclidian distance between any two vectors within $\cup_{q \in \mathbb{Z}_{[1,N]}} \{\hat{y}_q[k]\}$. Substituting this inequality into (B.6) gives that also the following inequality holds:

$$\begin{aligned} Y[k] &\preceq \sum_{q=1}^N \frac{\omega_q[k]}{\sum_{q=1}^N \omega_q[k]} ((\hat{y}^-[k] - \hat{y}^+[k])^\top (\hat{y}^-[k] - \hat{y}^+[k]))I \\ &= ((\hat{y}^-[k] - \hat{y}^+[k])^\top (\hat{y}^-[k] - \hat{y}^+[k]))I. \end{aligned} \quad (\text{B.7})$$

The second step starts with $I \preceq \lambda_{\min}^{-1}(R[k])R[k]$, which follows from the fact that $\lambda_{\min}(R[k])I \preceq R[k]$. Substitution of this inequality in (B.7) gives

that $Y[k] \preceq m[k]R[k]$ with $m[k] := (\hat{y}^-[k] - \hat{y}^+[k])^\top (\hat{y}^-[k] - \hat{y}^+[k]) \lambda_{\min}^{-1}(R[k])$. Moreover, the fact that $\mathcal{H}[e|t_k]$ is bounded and $R[k] \succ 0$, further implies that $(\hat{y}^-[k] - \hat{y}^+[k])^\top (\hat{y}^-[k] - \hat{y}^+[k]) < \infty$ and $\lambda_{\min}^{-1}(R[k]) < \infty$ hold, due to which $0 \leq m[k] < \infty$. Hence, the proof is completed. \square

B.2 Proof of Theorem 2.4.3

Statement lemma:

Consider the proposed EBSE and the hRDE of (2.19) and let $P[0] = \bar{\Sigma}[0]$. Then $P[k] \preceq \bar{\Sigma}[k]$ holds for all $t_k \in \mathbb{T}$.

The proof of this lemma starts by rewriting the formula of the hRDE into one expression, after which $P[k] \preceq \bar{\Sigma}[k]$ is proven, for all $t_k \in \mathbb{T}$.

To rewrite $\bar{\Sigma}[k]$ of the hRDE, let us introduce $\bar{\varsigma}[k] \in \{1, \varsigma_{\max}\}$ and $D[k] \in \mathbb{R}^n, t_k \in \mathbb{T}$, according to the following characterization:

$$\bar{\varsigma}[k] := \begin{cases} \varsigma_{\max} & \text{if } t_k \in \mathbb{T}_s, \\ 1 & \text{if } t_k \in \mathbb{T}_e \setminus \mathbb{T}_s, \end{cases}, \quad D[k] := \begin{cases} C^\top R_{\max}^{-1} C & \text{if } t_k \in \mathbb{T}_s, \\ 0 & \text{if } t_k \in \mathbb{T}_e \setminus \mathbb{T}_s. \end{cases}$$

Then substituting these variables into the $\bar{\Sigma}[k]$ of the hRDE, i.e., into (2.19), and restating the result of Lemma 2.4.2 for $P[k]$, gives that the following holds for all $t_k \in \mathbb{T}$, i.e.,

$$\begin{aligned} P[k] &\preceq \varsigma[k] \left((A(\tau_k)P[k-1]A^\top(\tau_k) + Q(\tau_k))^{-1} + C^\top R^{-1}[k]C \right)^{-1}, \\ \bar{\Sigma}[k] &= \bar{\varsigma}[k] \left((A(\tau_k)\bar{\Sigma}[k-1]A^\top(\tau_k) + Q(\tau_k))^{-1} + D[k] \right)^{-1}. \end{aligned} \quad (\text{B.8})$$

An important property that is used in the second part of this proof, yields

$$\bar{\varsigma}[k] \geq \varsigma[k] \quad \text{and} \quad D[k] \preceq C^\top R^{-1}[k]C, \quad \forall t_k \in \mathbb{T}. \quad (\text{B.9})$$

A derivation of (B.9) is presented in the following two cases that each correspond to a certain type of sample instant. To that end, recall that $R[k] \preceq R_{\max}$ and $\varsigma[k] \leq \varsigma_{\max}$ hold for all $t_k \in \mathbb{T}$ (see Property 2.4.1).

1. If $t_k \in \mathbb{T}_e \setminus \mathbb{T}_s$, then $\varsigma[k] = 1$, $\bar{\varsigma}[k] = 1$ and $D[k] = 0$ implies that $\bar{\varsigma}[k] = \varsigma[k]$ and $D[k] \preceq C^\top R^{-1}[k]C$.
2. If $t_k \in \mathbb{T}_s$, then $\bar{\varsigma}[k] = \varsigma_{\max}$ and $D[k] = C^\top R_{\max}^{-1}C$ implies that $\bar{\varsigma}[k] \geq \varsigma[k]$ and $D[k] \preceq C^\top R^{-1}[k]C$.

The second part proves the main inequality $P[k] \preceq \bar{\Sigma}[k]$, for all $t_k \in \mathbb{T}$, by induction and starts from $k = 0$.

Since $P[0] = \bar{\Sigma}[0]$, the formulas of (B.8) result in

$$\begin{aligned} P[1] &\preceq \varsigma[1] \left((A(\tau_1)P[0]A^\top(\tau_1) + Q(\tau_1))^{-1} + C^\top R^{-1}[1]C \right)^{-1}, \\ \bar{\Sigma}[1] &= \bar{\varsigma}[1] \left((A(\tau_1)\bar{\Sigma}[0]A^\top(\tau_1) + Q(\tau_1))^{-1} + D[1] \right)^{-1}. \end{aligned}$$

Inequality $D[1] \preceq C^\top R^{-1}[1]C$ of (B.9) gives that $(F + C^\top R^{-1}[1]C)^{-1} \preceq (F + D[1])^{-1}$, for any suitable $F \succ 0$, including $F = (A(\tau_1)P[0]A^\top(\tau_1) + Q(\tau_1))^{-1}$. Together with the above update of $P[1]$ and $\bar{\Sigma}[1]$, this further implies that $\zeta^{-1}[1]P[1] \preceq \bar{\zeta}^{-1}[1]\bar{\Sigma}[1]$. Moreover, since $\zeta^{-1}[1] \geq \bar{\zeta}^{-1}[1]$ (see (B.9)), the latter inequality will only hold if $P[1] \preceq \bar{\Sigma}[1]$.

The next step of induction shows that $P[k] \preceq \bar{\Sigma}[k]$, if $P[k-1] \preceq \bar{\Sigma}[k-1]$. To that end, let us define $F := (A(\tau_k)P[k-1]A^\top(\tau_k) + Q(\tau_k))^{-1}$ and $G := (A(\tau_k)\bar{\Sigma}[k-1]A^\top(\tau_k) + Q(\tau_k))^{-1}$. Then starting from $P[k-1] \preceq \bar{\Sigma}[k-1]$ one can derive that $G \preceq F$, which together with $D[k] \preceq C^\top R^{-1}[k]C$ of (B.9) induces $G + D[k] \preceq F + C^\top R^{-1}[k]C$ and thus $(F + C^\top R^{-1}[k]C)^{-1} \preceq (G + D[k])^{-1}$. Notice that the update formulas of (B.8) characterize $\zeta^{-1}[k]P[k] \preceq (F + C^\top R^{-1}[k]C)^{-1}$ and $\bar{\zeta}^{-1}[k]\bar{\Sigma}[k] = (G + D[k])^{-1}$, which further implies that $\zeta^{-1}[k]P[k] \preceq \bar{\zeta}^{-1}[k]\bar{\Sigma}[k]$. Moreover, since $\zeta^{-1}[k] \geq \bar{\zeta}^{-1}[k]$ (see (B.9)), the latter inequality can only hold if $P[k] \preceq \bar{\Sigma}[k]$, which completes the proof. \square

B.3 Proof of Lemma 2.4.4

Statement lemma:

Consider the hRDE of (2.19) and the wRDE of (2.18). At synchronous instants $t_k \in \mathbb{T}_s$ it holds that $\lambda_{\max}(\bar{\Sigma}[k]) = \lambda_{\max}(\Sigma[k])$, while $\lambda_{\max}(\bar{\Sigma}[k]) \leq \alpha^2(\tau_s)\lambda_{\max}(\Sigma[k-a(k)]) + \beta^2(\tau_s)$ holds at event instants $t_k \in \mathbb{T}_e \setminus \mathbb{T}_s$.

The following result is used in the proof of Lemma 2.4.4, for which $\mathbf{a}(k)$ denotes the number of sample instants in between the current instant $t_k \in \mathbb{T}$ and the first preceding synchronous instant $t_{k-a(k)} \in \mathbb{T}_s$ (see Section 2.4.1).

Lemma B.3.1 Consider the hRDE of (2.19) and let $\bar{\tau}_k := t_k - t_{k-a(k)}$. Then $\bar{\Sigma}[k^-] = A(\bar{\tau}_k)\bar{\Sigma}[k-a(k)]A^\top(\bar{\tau}_k) + Q(\bar{\tau}_k)$ holds for all $t_k \in \mathbb{T}_s$.

The proof of this lemma is presented in Section B.3.1

The proposed statement of Lemma 2.4.4 is then proven if: (i) $\bar{\Sigma}[k] = \Sigma[k]$ holds for all $t_k \in \mathbb{T}_s$ and (ii) $\lambda_{\max}(\bar{\Sigma}[k]) \leq \alpha^2(\tau_s)\lambda_{\max}(\Sigma[k-a(k)]) + \beta^2(\tau_s)$ holds for all $t_k \in \mathbb{T}_e \setminus \mathbb{T}_s$.

The proof of implication (i), i.e., $\bar{\Sigma}[k] = \Sigma[k]$ for all $t_k \in \mathbb{T}_s$, starts by applying Lemma B.3.1. Since $t_k, t_{k-a(k)} \in \mathbb{T}_s$, it follows that $\bar{\tau}_k = \tau_s$ is equal to the sampling time. Hence, the results of this lemma imply that the update of $\bar{\Sigma}[k]$ given in (2.19), for all $t_k \in \mathbb{T}_c$, becomes

$$\begin{aligned} \bar{\Sigma}[k^-] &= A(\tau_s)\bar{\Sigma}[k-a(k)]A^\top(\tau_s) + Q(\tau_s), \\ \bar{\Sigma}[k] &= \varsigma_{\max}(\bar{\Sigma}^{-1}[k^-] + C^\top R_{\max}^{-1}C)^{-1}. \end{aligned} \tag{B.10}$$

Since the above update of $\bar{\Sigma}[k]$ for all $t_k \in \mathbb{T}_s$ is equivalent to $\Sigma[k]$ of the wRDE, as shown in (2.18), $\bar{\Sigma}[k] = \Sigma[k]$ holds for all $t_k \in \mathbb{T}_s$ if $\bar{\Sigma}[0] = \Sigma[0]$.

The second implication analyzes $\bar{\Sigma}[k]$ in between the synchronous instants, i.e., for all $t_k \in \mathbb{T}_e \setminus \mathbb{T}_s$. At these instants, the update formula of the

hRDE gives that $\bar{\Sigma}[k] = \bar{\Sigma}[k^-]$, see (2.19). An expression of $\bar{\Sigma}[k^-]$ in terms of $\bar{\Sigma}[k-a(k)]$ is found in Lemma B.3.1, due to which the update of the hRDE at an event instant is conform to

$$\bar{\Sigma}[k] = A(\bar{\tau}_k)\bar{\Sigma}[k-a(k)]A^\top(\bar{\tau}_k) + Q(\bar{\tau}_k), \quad \forall t_k \in \mathbb{T}_e \setminus \mathbb{T}_s. \quad (\text{B.11})$$

The first implication established $\bar{\Sigma}[k] = \Sigma[k]$, for all $t_k \in \mathbb{T}_s$. By definition we have that $t_{k-a(k)} \in \mathbb{T}_s$ holds for $\bar{\Sigma}[k-a(k)]$ of (B.11). As such, $\bar{\Sigma}[k-a(k)] = \Sigma[k-a(k)]$ can be substituted into (B.11), which after taking the largest eigenvalues on both sides of the equation gives that

$$\lambda_{\max}(\bar{\Sigma}[k]) = \lambda_{\max}(A(\bar{\tau}_k)\Sigma[k-a(k)]A^\top(\bar{\tau}_k) + Q(\bar{\tau}_k)) \quad (\text{B.12a})$$

$$\leq \lambda_{\max}(A(\bar{\tau}_k)\Sigma[k-a(k)]A^\top(\bar{\tau}_k)) + \lambda_{\max}(Q(\bar{\tau}_k)) \quad (\text{B.12b})$$

$$= \sigma_{\max}(A(\bar{\tau}_k)\Sigma[k-a(k)]A^\top(\bar{\tau}_k)) + \lambda_{\max}(Q(\bar{\tau}_k)) \quad (\text{B.12c})$$

$$\leq \sigma_{\max}^2(A(\bar{\tau}_k))\sigma_{\max}(\Sigma[k-a(k)]) + \lambda_{\max}(Q(\bar{\tau}_k)), \quad (\text{B.12d})$$

holds for all $t_k \in \mathbb{T}_e \setminus \mathbb{T}_s$. Notice that only $A(\bar{\tau}_k)$ and $Q(\bar{\tau}_k)$ depend on t_k . Therefore, finding the maximum value of (B.12d), for all $t_k \in \mathbb{T}_e \setminus \mathbb{T}_s$, results on finding the largest values of both $\sigma_{\max}(A(\bar{\tau}_k))$ and $\lambda_{\max}(Q(\bar{\tau}_k))$, for all $\bar{\tau}_k \in \mathbb{R}_{(0,\tau_s)}$. The definition of $\alpha(\tau_s)$ and $\beta(\tau_s)$ in Section 2.4.1 imply that $\alpha(\tau_s) \geq \sigma_{\max}(A(\bar{\tau}_k))$ and $\beta(\tau_s) \geq \sqrt{\lambda_{\max}(Q(\bar{\tau}_k))}$, for all $\bar{\tau}_k \in \mathbb{R}_{(0,\tau_s)}$. Therefore substituting these variables into (B.12d) gives that

$$\lambda_{\max}(\bar{\Sigma}[k]) \leq \alpha^2(\tau_s)\lambda_{\max}(\Sigma[k-a(k)]) + \beta^2(\tau_s) \quad \forall t_k \in \mathbb{T}_e \setminus \mathbb{T}_s, \quad (\text{B.13})$$

which completes this proof. \square

B.3.1 Proof of Lemma B.3.1

The proof employs the following condition, which was established in Chapter 1 to guarantee a correct discretization of the process, i.e.,

Condition B.3.2 The process model (2.1) satisfies $A(\tau_1+\tau_2) = A(\tau_1)A(\tau_2)$ and $Q(\tau_1+\tau_2) = A(\tau_1)Q(\tau_2)A^\top(\tau_1) + Q(\tau_1)$, for any $\tau_1, \tau_2 \in \mathbb{R}_{>0}$.

The goal is to prove the following equation:

$$\bar{\Sigma}[k^-] = A(\bar{\tau}_k)\bar{\Sigma}[k-a(k)]A^\top(\bar{\tau}_k) + Q(\bar{\tau}_k), \quad \forall t_k \in \mathbb{T}_s, \quad (\text{B.14})$$

where $\mathbf{a}(k)$ characterizes the first preceding synchronous instant $t_{k-a(k)} \in \mathbb{T}_s$, for any $t_k \in \mathbb{T}$. Notice that $\bar{\Sigma}[k^-]$ of (B.14) is independent on the number of sample instants in between t_k and $t_{k-a(k)}$. Hence, the lemma is proven by induction, for which the original formula of $\bar{\Sigma}[k^-]$ is recalled for clarity, i.e.,

$$\bar{\Sigma}[k^-] := A(\tau_k)\bar{\Sigma}[k-1]A^\top(\tau_k) + Q(\tau_k), \quad \forall t_k \in \mathbb{T}. \quad (\text{B.15})$$

The first step of induction is to prove (B.14) in case $\mathbf{a}(\mathbf{k}) = 1$, i.e., $\bar{\Sigma}[\mathbf{k}-1] = \bar{\Sigma}[\mathbf{k}-\mathbf{a}(\mathbf{k})]$ and $\tau_{\mathbf{k}} = \bar{\tau}_{\mathbf{k}}$. Substituting these values into $\bar{\Sigma}[\mathbf{k}^-]$ of (B.15), results in $\bar{\Sigma}[\mathbf{k}^-] = A(\bar{\tau}_{\mathbf{k}})\bar{\Sigma}[\mathbf{k}-\mathbf{a}(\mathbf{k})]A^\top(\bar{\tau}_{\mathbf{k}}) + Q(\bar{\tau}_{\mathbf{k}})$ and completes the first step.

The second step of induction is to prove (B.14) for any $t_{\mathbf{k}} \in \mathbb{T}$ and $\mathbf{a}(\mathbf{k}) > 1$, given that (B.14) holds for $t_{\mathbf{k}-1}$. Note that $\mathbf{a}(\mathbf{k}) = \mathbf{a}(\mathbf{k}-1) + 1$ implies $\bar{\Sigma}[\mathbf{k}-\mathbf{a}(\mathbf{k})] = \bar{\Sigma}[\mathbf{k}-1-\mathbf{a}(\mathbf{k}-1)]$. Substituting this result into the formula of (B.14) at $t_{\mathbf{k}-1}$, i.e., $\bar{\Sigma}[(\mathbf{k}-1)^-] = A(\bar{\tau}_{\mathbf{k}-1})\bar{\Sigma}[\mathbf{k}-1-\mathbf{a}(\mathbf{k}-1)]A^\top(\bar{\tau}_{\mathbf{k}-1}) + Q(\bar{\tau}_{\mathbf{k}-1})$, induces

$$\bar{\Sigma}[(\mathbf{k}-1)^-] = A(\bar{\tau}_{\mathbf{k}-1})\bar{\Sigma}[\mathbf{k}-\mathbf{a}(\mathbf{k})]A^\top(\bar{\tau}_{\mathbf{k}-1}) + Q(\bar{\tau}_{\mathbf{k}-1}). \quad (\text{B.16})$$

Since $\mathbf{a}(\mathbf{k}) > 1$, we have that $t_{\mathbf{k}-1} \in \mathbb{T}_e \setminus \mathbb{T}_s$. Further, (2.19) gives that the hRDE update at this event instant is $\bar{\Sigma}[(\mathbf{k}-1)] = \bar{\Sigma}[(\mathbf{k}-1)^-]$. Therefore, one can rewrite (B.15) into $\bar{\Sigma}[\mathbf{k}^-] = A(\tau_{\mathbf{k}})\bar{\Sigma}[(\mathbf{k}-1)^-]A^\top(\tau_{\mathbf{k}}) + Q(\tau_{\mathbf{k}})$, which after substitution of (B.16) results in

$$\begin{aligned} \bar{\Sigma}[\mathbf{k}^-] &= A(\tau_{\mathbf{k}})(A(\bar{\tau}_{\mathbf{k}-1})\bar{\Sigma}[\mathbf{k}-\mathbf{a}(\mathbf{k})]A^\top(\bar{\tau}_{\mathbf{k}-1}) + Q(\bar{\tau}_{\mathbf{k}-1}))A^\top(\tau_{\mathbf{k}}) + Q(\tau_{\mathbf{k}}) \\ &= A(\tau_{\mathbf{k}})A(\bar{\tau}_{\mathbf{k}-1})\bar{\Sigma}[\mathbf{k}-\mathbf{a}(\mathbf{k})]A^\top(\bar{\tau}_{\mathbf{k}-1})A^\top(\tau_{\mathbf{k}}) + A(\tau_{\mathbf{k}})Q(\bar{\tau}_{\mathbf{k}-1})A^\top(\tau_{\mathbf{k}}) + Q(\tau_{\mathbf{k}}). \end{aligned}$$

Based on the results of Condition B.3.2, after substituting $\tau_1 = \tau_{\mathbf{k}}$ and $\tau_2 = \bar{\tau}_{\mathbf{k}}$, the above formula can be rewritten as follows:

$$\bar{\Sigma}[\mathbf{k}^-] = A(\tau_{\mathbf{k}} + \bar{\tau}_{\mathbf{k}-1})\bar{\Sigma}[\mathbf{k}-\mathbf{a}(\mathbf{k})]A^\top(\tau_{\mathbf{k}} + \bar{\tau}_{\mathbf{k}-1}) + Q(\tau_{\mathbf{k}} + \bar{\tau}_{\mathbf{k}-1}). \quad (\text{B.17})$$

As $\tau_{\mathbf{k}} + \bar{\tau}_{\mathbf{k}-1} = \bar{\tau}_{\mathbf{k}}$, $\bar{\Sigma}[\mathbf{k}^-]$ in (B.17) is similar to (B.14), which completes the second step of induction and the proof of Lemma B.3.1. \square

C

Proofs corresponding to Chapter 3

C.1 Proof of Lemma 3.2.1

Statement lemma:

Let $\Delta_{KL} \in \mathbb{R}_+$ be given and let $\alpha(t)$ be conform to (3.5). Then $0 < \Delta_{KL} - \alpha(t) \leq \Delta_{KL} + \frac{1}{2}n$, for all $t \in \mathbb{T}$.

For clarity of expression, let us remove the time-index t , due to which α of (3.5) gives that $\alpha = \frac{1}{2}(\log |P_2| |P_1|^{-1} + \text{tr}(P_2^{-1}P_1) - n)$. Further, (3.5) directly proves that $0 < \Delta_{KL} - \alpha$, since $(\hat{x}_1 - \hat{x}_2)^\top P_2^{-1}(\hat{x}_1 - \hat{x}_2) \geq 0$ and $d(p_1(x)||p_2(x)) < \Delta_{KL}$. Then, the above statement is proven if $\Delta_{KL} - \alpha(t) \leq \Delta_{KL} + \frac{1}{2}n$, or similarly, if $\alpha \geq -\frac{1}{2}n$. The expression of α gives that this is guaranteed when: (i) $\log |P_2| |P_1|^{-1} \geq 0$ and (ii) $\text{tr}(P_2^{-1}P_1) \geq 0$ hold.

To prove implication (i), let us recall from (3.4) that $P_1^{-1} = P_2^{-1} + C^\top V^{-1}C$. The fact that $V \succ 0$ implies $C^\top V^{-1}C \succeq 0$ and thus $P_1^{-1} \succeq P_2^{-1}$. The result of Corollary 8.4.10 of (Bernstein, 2005), i.e., for all $A \succeq B$ it holds that $|A| \geq |B|$, further gives that $|P_1|^{-1} \succeq |P_2|^{-1}$. Hence, $|P_2| |P_1|^{-1} \geq 1$ and thus $\log |P_2| |P_1|^{-1} \geq 0$.

Implication (ii) is proven by observing that $P_1, P_2^{-1} \succ 0$. Applying Lemma 2.2 of (Belmega et al., 2009), i.e. for any $A, B \succeq 0$ it holds that $\text{tr}(AB) \geq 0$, induces $\text{tr}(P_2^{-1}P_1) \geq 0$, which completes the proof. \square

C.2 Proof of Lemma 3.2.2

Statement lemma:

Let $C \in \mathbb{R}^{m \times n}$ of (3.1) be such that $\text{rank}(C) = m$ and let $\Upsilon(t)$ satisfy (3.6). Then, $\Upsilon(t) \succ 0$ holds for all $t \in \mathbb{T}$.

For clarity of expression, let us remove the time-index t . Further, let us state Proposition 8.1.2 of (Bernstein, 2005) that will be used in this proof, i.e., for any $B \succ 0$ and $A \in \mathbb{R}^{m \times n}$ it holds that $ABA^\top \succ 0$ if $\text{rank } A = m$. Then, from the fact that $P_2^{-1} \succ 0$, Proposition 8.1.2 gives that $P_1 P_2^{-1} P_1 \succ 0$, $C P_1 P_2^{-1} P_1 C^\top \succ 0$ and thus $V^{-1} C P_1 P_2^{-1} P_1 C^\top V^{-1} \succ 0$ holds as well. This latter inequality further implies that the expression of Υ in (3.6), i.e., $\Upsilon = V^{-1} C P_1 P_2^{-1} P_1 C^\top V^{-1}$, satisfies $\Upsilon \succ 0$, which completes this proof. \square

C.3 Proof of Lemma 3.4.2

Before stating the lemma, let us first recall the considered formulas of (3.10) and (3.13), respectively, i.e.,

$$V(Ax + Bu(\hat{x})) - V(x) + a\|x\|^c - \vartheta(\|\varpi\|) \leq 0, \quad (\text{C.1})$$

$$V(A\hat{x} + Bu(\hat{x})) - V(x) + a\|x\|^c \leq 0, \quad (\text{C.2})$$

$$V(A(\hat{x} - \varpi_j) + Bu(\hat{x})) - V(x) + a\|x\|^c - \varsigma_j \leq 0, \quad \forall j \in \mathbb{Z}_{[1, N]}.$$

Then an equivalent statement of the lemma is the following:

Let Assumption 3.4.1¹ be satisfied and let \hat{x} and x be given. If there exist a $u(\hat{x})$ and ϖ_j , for all $j \in \mathbb{Z}_{[1, N]}$, such that (C.2) holds, then (C.1) holds for the same $u(\hat{x})$, with $\vartheta(\|\varpi\|) := \eta\|\varpi\|$ and $\eta := \max_{i=1, \dots, M} \{\|\zeta T_i (WT_i)^{-1}\|\}$.

This lemma is proven by merging the two inequalities of (C.2) into one inequality, which is then related to the original (C.1).

To accomplish the merge, note that for any $\varpi \in \mathbb{W}$ there exists a simplex \mathcal{S}_i , such that $\varpi \in \mathcal{S}_i = \text{Co}\{0, \varpi_{q(i,1)}, \dots, \varpi_{q(i,l)}\}$. Hence, for some weights $\kappa_0, \kappa_1, \dots, \kappa_l \in \mathbb{R}_+$ that satisfy $\sum_{r=\mathbb{Z}_{[0, l]}} \kappa_r = 1$, any $\varpi \in \mathcal{S}_i$ can be established as a convex combination $\varpi = \sum_{r=\mathbb{Z}_{[0, l]}} \kappa_r \varpi_{q(i,r)} + \kappa_0 0 = \sum_{r=\mathbb{Z}_{[1, l]}} \kappa_r \varpi_{q(i,r)}$. Substituting $WT_i = (\varpi_{q(i,1)} \cdots \varpi_{q(i,l)})$ gives that $\varpi = WT_i (\kappa_1 \cdots \kappa_l)^\top$ and thus

$$(\kappa_1 \ \kappa_2 \ \cdots \ \kappa_l)^\top = (WT_i)^{-1} \varpi.$$

Further, as for any combination $(i, r) \in \mathbb{Z}_{[1, M]} \times \mathbb{Z}_{[1, l]}$ there exists a $j \in \mathbb{Z}_{[1, N]}$, such that $\varpi_j = \varpi_{q(i,r)}$, the second inequality of (C.2) can be rewritten as

$$V(A(\hat{x} - \varpi_{q(i,r)}) + Bu(\hat{x})) - V(x) + a\|x\|^c - \varsigma_{q(i,r)} \leq 0, \quad \forall i \in \mathbb{Z}_{[1, M]}, r \in \mathbb{Z}_{[1, l]}.$$

Multiplying the above inequality with the corresponding κ_r does not affect its outcome. Similarly, the first inequality of (C.2) can be multiplied with κ_0 . Then, for any $\omega \in \mathcal{S}_i$, the following set of inequalities are satisfied:

$$\kappa_0 (V(A\hat{x} + Bu(\hat{x})) - V(x) + a\|x\|^c) \leq 0,$$

$$\sum_{r \in \mathbb{Z}_{[1, l]}} \kappa_r \left(V(A(\hat{x} - \varpi_{q(i,r)}) + Bu(\hat{x})) - V(x) + a\|x\|^c - \varsigma_{q(i,r)} \right) \leq 0.$$

Addition of the above two inequalities, with $\varpi_{q(i,r=0)} := 0$, for all $i \in \mathbb{Z}_{[1, M]}$, while taking into account that $\sum_{r=\mathbb{Z}_{[1, l]}} \kappa_r = 1$, results in

$$\sum_{r=\mathbb{Z}_{[0, l]}} \kappa_r V(A(\hat{x} - \varpi_{q(i,r)}) + Bu(\hat{x})) - V(x) + a\|x\|^c - \sum_{r=\mathbb{Z}_{[1, l]}} \kappa_r \varsigma_{q(i,r)} \leq 0. \quad (\text{C.3})$$

¹The assumption states that $V(\cdot)$ is a continuous, convex Lyapunov function.

Convexity of V gives that $V(\alpha a + (1 - \alpha)b) \leq \alpha V(a) + (1 - \alpha)V(b)$ holds for any $a, b \in \mathbb{R}^n$ and $\alpha \in [0, 1]$. Therefore, if (C.2) is met, then the inequality in (C.3) is also met, which further implies that

$$V\left(A(\hat{x} - \sum_{r=\mathbb{Z}_{[0,1]}} \kappa_r \varpi_{q(i,r)}) + Bu(\hat{x})\right) - V(x) + a\|x\|^c - \sum_{r=\mathbb{Z}_{[1,1]}} \kappa_r \varsigma_{q(i,r)} \leq 0.$$

Since $\sum_{r=\mathbb{Z}_{[0,1]}} \kappa_r \varpi_{q(i,r)} = \varpi$ was already established and formula (3.12) gives that $\zeta T_i = (\varsigma_{q(i,1)} \cdots \varsigma_{q(i,l)})$, the above inequality is equivalent to $V(A(\hat{x} - \varpi) + Bu(\hat{x})) - V(x) + a\|x\|^c - \zeta T_i (\kappa_1 \cdots \kappa_l)^\top \leq 0$. Substituting $x = \hat{x} - \varpi$ and $(WT_i)^{-1} \varpi = (\kappa_1 \cdots \kappa_l)^\top$ into the inequality further yields $V(Ax + Bu(\hat{x})) - V(x) + a\|x\|^c - \zeta T_i (WT_i)^{-1} \varpi \leq 0$, for any $\varpi \in \mathcal{S}_i$ and all $i \in \mathbb{Z}_{[1,M]}$, which can be rewritten by considering the fact that $a^\top b \leq \|a\| \|b\|$ holds for any $a, b \in \mathbb{R}^n$, i.e.,

$$V(Ax + Bu(\hat{x})) - V(x) + a\|x\|^c - \|\zeta T_i (WT_i)^{-1}\| \|\varpi\| \leq 0, \wedge \varpi \in \mathcal{S}_i. \quad (\text{C.4})$$

Notice that (C.4) holds for any simplex \mathcal{S}_i , given that $\varpi \in \mathcal{S}_i$. To enable an equality that is satisfied for any $\varpi \in \mathbb{W} (= \cup_{i \in \mathbb{Z}_{[1,M]}} \{\mathcal{S}_i\})$, one can derive that if (C.4) holds for all $i \in \mathbb{Z}_{[1,M]}$, then the following is also met, i.e.,

$$V(Ax + Bu(\hat{x})) - V(x) + a\|x\|^c - \max_{i \in \mathbb{Z}_{[1,M]}} \{\|\zeta T_i (WT_i)^{-1}\|\} \|\varpi\| \leq 0, \forall \varpi \in \mathbb{W}.$$

The above inequality is equivalent to (C.1) in case $\vartheta(\|\varpi\|) = \eta \|\varpi\|$ and $\eta := \max_{i=1, \dots, M} \{\|\zeta T_i (WT_i)^{-1}\|\}$, which completes the proof. \square

C.4 Proof of Theorem 3.4.3

Before stating the theorem, let us recall the considered formulas of (3.4.2).

$$u(t_k) := \arg_{u \in \mathbb{U}} \min J(\zeta), \quad \forall \varsigma_j \in \mathbb{R}_+ \text{ and } \forall j \in \mathbb{Z}_{[1,M]}, \quad (\text{C.5a})$$

subject to

$$Az + Bu \in \mathbb{X}, \quad \forall z \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k), \quad (\text{C.5b})$$

$$V(A\hat{x}(t_k) + Bu) - V_{\min}(t_k) + a_{\max}(t_k) \leq 0, \quad (\text{C.5c})$$

$$V(A(\hat{x}(t_k) - \varpi_j(t_k)) + Bu) - V_{\min}(t_k) + a_{\max}(t_k) - \varsigma_j \leq 0. \quad (\text{C.5d})$$

Then an equivalent statement of the theorem is the following:

Let a Lyapunov function V , satisfying Assumption 3.4.1², and a cost-function J be given. Further, let a bounded set $\overline{\mathbb{W}}$ be defined, such that $\mathbb{W}(t_k) \subseteq \overline{\mathbb{W}}$ for all t_k , and suppose that (C.5) is feasible for all $\hat{x}(t_k) \in \mathbb{X} \oplus \overline{\mathbb{W}}$ and all t_k . Then the difference inclusion $x(t_k + \tau_s) \in \mathcal{X}(t_k + \tau_s)$, for all $k \in \mathbb{Z}_+$, is ISS in \mathbb{X} for inputs in $\overline{\mathbb{W}}$.

²The assumption states that $V(\cdot)$ is a continuous, convex Lyapunov function.

As $x(t_k) \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$, for all $k \in \mathbb{Z}_+$, note that (C.5b) induces $x(t_{k+\tau_s}) = Ax(t_k) + Bu \in \mathbb{X}$ for all $u \in \mathcal{U}(t_k)$, i.e., the real state satisfies state constraints. Furthermore, since $x(t_k + \tau_s) \in \{\hat{x}(t_k + \tau_s)\} \oplus \mathbb{W}(t_k + \tau_s)$, $x(t_k + \tau_s) \in \mathbb{X}$ and $\mathbb{W}(t_k) \subseteq \overline{\mathbb{W}}$ is a symmetric set, one can further derive that $\hat{x}(t_k + \tau_s) \in \mathbb{X} \oplus \mathbb{W}(t_k + \tau_s) \subseteq \mathbb{X} \oplus \overline{\mathbb{W}}$. Therefore, the statement of the lemma implies that (C.5b) remains feasible also at $k + \tau_s$, due to which $\mathcal{X}(t_k + \tau_s) \subseteq \mathbb{X}$ for any $\varpi(t_k) \in \mathbb{W}(t_k)$ and all $k \in \mathbb{Z}_+$.

The next step is proving ISS. Since $V(x(t_k)) \geq V_{\min}(t_k)$ (see (3.15)) and $a\|x(t_k)\|^c \leq a_{\max}(t_k)$ (see (3.16)), satisfying (C.5c) and (C.5d) implies that the inequalities of (3.13) are met, i.e.,

$$\begin{aligned} V(A\hat{x}(t_k) + Bu) - V(x(t_k)) + a\|x(t_k)\|^c &\leq 0, \\ V(A(\hat{x}(t_k) - \varpi_j(t_k)) + Bu) - V(x(t_k)) + a\|x(t_k)\|^c - \varsigma_j &\leq 0, \quad \forall j \in \mathbb{Z}_{[1, N]}, \end{aligned}$$

for all $x(t_k) \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$ and $u \in \mathcal{U}(t_k)$. Then, from Lemma 3.4.2 it follows that (3.10) holds with $\vartheta(\|\varpi\|) := \eta(t_k)\|\varpi\|$ and $\eta(t_k)$ of (3.14), i.e.

$$V(Ax(t_k) + Bu(t_k)) - V(x(t_k)) + a\|x(t_k)\|^c - \vartheta(\|\varpi(t_k)\|) \leq 0,$$

for all $x(t_k) \in \{\hat{x}(t_k)\} \oplus \mathbb{W}(t_k)$ and $u(t_k) \in \mathcal{U}(t_k)$. It was already proven in (Kellett and Teel, 2005) that satisfying the above inequality, with $V(\cdot)$ being an ISS Lyapunov function, would guarantee ISS of the corresponding control system. Hence, closed-loop ISS is proven if there exists a positive value $\vartheta(\|\varpi(t_k)\|)$, for all t_k , that is derived from (C.5). To that extent, let us define

$$\varsigma^* := \max_{x \in \text{cl}(\mathbb{X}), u \in \text{cl}(\mathbb{U})} \{V(Ax + Bu) - V(x) + a\|x\|^c\},$$

in which $\text{cl}(\mathbb{X})$ and $\text{cl}(\mathbb{U})$ denote the closure of \mathbb{X} and \mathbb{U} , respectively. Since \mathbb{X} , \mathbb{U} and $\overline{\mathbb{W}}$ are assumed to be bounded sets, ς^* exists, and inequality (C.5d) is always satisfied for $\varsigma_j = \varsigma^*$, for all $j \in \mathbb{Z}_{[1, N]}$ at each sample instant t_k , irrespective of x , u and the vertices of $\mathbb{W}(t_k) \subseteq \overline{\mathbb{W}}$. This in turn, via (3.14), ensures the existence of a positive η^* such that $\eta(t_k) \leq \eta^*$ for all t_k and for all $\varpi(t_k) \in \overline{\mathbb{W}}$. Hence, it is proven that inequality (3.10) holds and as $\mathcal{X}(t_{k+1}) \subseteq \mathbb{X}$, closed-loop ISS of $x(t_{k+1}) \in \mathcal{X}(t_{k+1})$ is proven for all for inputs in $\varpi(t_k) \in \overline{\mathbb{W}}$. \square

\mathcal{D}

Proofs corresponding to Chapter 4

D.1 Proof of Corollary 4.4.3

Statement corollary

Let x_1 and x_2 represent two unbiased and consistent estimates of x that are independent. Then, $x_3 := \Omega(x_1, x_2)$ is a Gaussian distributed random vector characterized by

$$P_3 := (P_1^{-1} + P_2^{-1})^{-1}, \quad \hat{x}_3 := P_3 (P_1^{-1} \hat{x}_1 + P_2^{-1} \hat{x}_2), \quad (\text{D.1})$$

such that x_3 is the consistent fused estimate with minimal $\text{tr}(P_3)$.

The proof directly results from the fact that any linear function of Gaussian distributed random vectors is again a Gaussian distributed random vector, see (Edwards, 1979). As such, $x_3 := \Omega(x_1, x_2)$ implies that $x_3 = (P_1^{-1} + P_2^{-1})^{-1} (P_1^{-1} \hat{x}_1 + P_2^{-1} \hat{x}_2)$ is random variable that is Gaussian distributed. The corresponding definitions for the expectation \hat{x}_3 and covariance P_3 of this random variable x_3 , then yield $P_3 = (P_1^{-1} + P_2^{-1})^{-1}$ and $\hat{x}_3 = (P_1^{-1} + P_2^{-1})^{-1} (P_1^{-1} \hat{x}_1 + P_2^{-1} \hat{x}_2)$ and thus satisfy (D.1). Alternatively, since x_1 and x_2 are independent, one could substitute $x_i = x_1$, $x_j = x_2$, $x_f = x_3$ and $P_{ij} = 0_{n \times n}$ into Theorem 4.2.1, which further gives that $P_3 = P_1 - P_1(P_1 + P_2)^{-1}P_1$ and $\hat{x}_3 = \hat{x}_1 - P_1(P_1 + P_2)^{-1}(\hat{x}_1 - \hat{x}_2)$. One can derive that these expressions are equivalent to P_3 and \hat{x}_3 of (D.1), by employing the ‘‘matrix inversion lemma’’ presented in (Woodbury, 1950). The results of this theorem then given that the fusion formulas of (D.1) results in a fused estimate x_3 , which is consistent and has a minimal $\text{tr}(P_3)$. Hence, the proof is completed. \square

D.2 Derivation of Proposition 4.4.3

Before the statement of Proposition 4.4.3 is given, let us first recall that the prior estimates x_i and x_j , characterized by the Gaussian distributions $p(x_i) = G(x_i, \hat{x}_i, P_i)$ and $p(x_j) = G(x_j, \hat{x}_j, P_j)$, were parameterized as follows:

$$\begin{aligned} x_i &= \Omega(x_{ij}, x_{ii}) \quad \text{and} \quad x_j = \Omega(x_{ij}, x_{jj}), \\ \text{with } p(x_{ii}) &:= G(x_{ii}, \theta_i, \Theta_i), \quad p(x_{jj}) := G(x_{jj}, \theta_j, \Theta_j), \\ p(x_{ij}) &:= G(x_{ij}, \gamma, \Gamma). \end{aligned} \quad (\text{D.2})$$

Further, Corollary 4.4.1 states that $\Omega(\cdot, \cdot)$ of two independent prior estimates is a fusion function that attains the Best Linear Unbiased Estimate. Then,

a statement equivalent to Proposition 4.4.3, yields

Let the prior estimates x_i and x_j be given according to (D.2), such that x_{ii} , x_{ij} and x_{jj} are pair-wise independent. Then, the fused estimate x_f of the prior estimates x_i and x_j , yields $x_f = \Omega(x_i, x_{jj})$, where

$$P_f = (P_i^{-1} + \Theta_j^{-1})^{-1}, \quad \hat{x}_f = P_f (P_i^{-1} \hat{x}_i + \Theta_j^{-1} \hat{x}_{jj}). \quad (\text{D.3})$$

Before deriving (D.3), note that x_j is the fusion result of the two prior estimates x_{ij} and x_{jj} . Hence, instead of fusing x_i and x_j directly, one could perform two fusion steps recursively: first merge x_i with x_{ij} and fuse that result further with x_{jj} to determine x_f . Let us start with the first merging step. From the fact that $x_i = \Omega(x_{ii}, x_{ij})$, one can derive that $\text{cov}(x_i, x_{ij}) = P_i$ is available, due to which the fusion approach presented in Theorem 4.2.1 can be used for merging $p(x_i) = G(x_i, \hat{x}_i, P_i)$ and $p(x_{ij}) = G(x_{ij}, \gamma, \Gamma)$. This theorem gives that fusion of x_i and x_{ij} establishes x_i as their fused estimate (because x_i and x_{ij} are “fully correlated”). The next step is to fuse this result, i.e., x_i , with x_{jj} to determine the desired fused estimate x_f . Note that x_i and x_{jj} are independent, due to which that can be fused by the fusion function $\Omega(\cdot)$ in Corollary 4.4.3 and thus $x_f = \Omega(x_i, x_{jj})$.

D.3 Proof of Lemma 4.4.5

Statement lemma:

Let $P_i, P_j \succ 0$ be given and let $\hat{P}_i, \hat{P}_j, \hat{\Gamma}$ satisfy (4.16) and (4.17), for some nonsingular $T \in \mathbb{R}^{n \times n}$. Then $\Gamma = T \hat{\Gamma} T^\top$.

The transformation presented in (4.16), yields $\hat{P}_i = T^{-1} P_i T^{-\top}$ and $\hat{P}_j = T^{-1} P_j T^{-\top}$. Similarly, let us recall the definitions for $\hat{\Gamma}$ and Γ , as they were presented in (4.17) and (4.15), respectively, i.e.,

$$\begin{aligned} \hat{\Gamma} &= \arg \min_{\hat{\Upsilon} \in \mathbb{R}^{n \times n}} \log |\hat{\Upsilon}|, \\ &\text{subject to } \hat{\Upsilon} \succeq \hat{P}_i, \hat{\Upsilon} \succeq \hat{P}_j, \end{aligned} \quad (\text{D.4})$$

$$\begin{aligned} \Gamma &= \arg \min_{\Upsilon \in \mathbb{R}^{n \times n}} \log |\Upsilon|, \\ &\text{subject to } \Upsilon \succeq P_i, \Upsilon \succeq P_j, \end{aligned} \quad (\text{D.5})$$

and let $\hat{\Upsilon} := T^{-1} \Upsilon T^{-\top}$. Then from the fact that $T^{-\top} := (T^\top)^{-1} = (T^{-1})^\top$ holds for any nonsingular matrix T , see Proposition 2.6.8 of (Bernstein, 2005), one can derive that $\Upsilon = T \hat{\Upsilon} T^\top$, $P_i = T \hat{P}_i T^\top$ and $P_j = T \hat{P}_j T^\top$. As such, for proving the statement it suffices to show that:

- (i) $\arg \min_{\hat{\Upsilon} \in \mathbb{R}^{n \times n}} \log |\hat{\Upsilon}| = \arg \min_{\Upsilon \in \mathbb{R}^{n \times n}} \log |\Upsilon|$;
- (ii) $\hat{\Upsilon} \succeq \hat{P}_i$ and $\hat{\Upsilon} \succeq \hat{P}_j$ induce $\Upsilon \succeq P_i$ and $\Upsilon \succeq P_j$.

Implication (ii) results directly from Proposition 8.1.2 of (Bernstein, 2005), i.e., if $A \succeq B$ then $CAC^\top \succeq CBC^\top$ holds for any suitable C . As such, $\hat{\Upsilon} \succeq \hat{P}_i$ implies $T\hat{\Upsilon}T^\top \succeq T\hat{P}_iT^\top$, which is further equivalent to $\Upsilon \succeq P_i$. Similarly, $\Upsilon \succeq P_j$ is derived from $\hat{\Upsilon} \succeq \hat{P}_j$.

To prove (i) consider Proposition 2.7.3 of (Bernstein, 2005), i.e., $|EF| = |E||F|$ holds for any nonsingular matrices E, F . As such, one can derive that $\log|\Upsilon| = \log|T\hat{\Upsilon}T^\top| = \log|\hat{\Upsilon}| + 2\log|T|$ and thus $\arg \min_{\hat{\Upsilon}} \log|\Upsilon| = \arg \min_{\hat{\Upsilon}} \log|\hat{\Upsilon}|$, which completes the proof. \square

D.4 Proof of Lemma 4.4.6

Statement lemma:

Let $P_i = I_n$ and $P_j = D_j$, for some diagonal matrix $D_j \succ 0$. Then $\Gamma = D_\Gamma$.

The definition of Γ used in this lemma was already recalled in (D.5). Further, the diagonal matrix $D_\Gamma \in \mathbb{R}^{n \times n}$ was defined in (4.18), for some diagonal $D_j \in \mathbb{R}^{n \times n}$, with the following diagonal elements:

$$\lfloor D_\Gamma \rfloor_{qq} = \max\{1, \lfloor D_j \rfloor_{qq}\}, \quad \forall q \in \mathbb{Z}_{[1,n]}. \quad (\text{D.6})$$

Therefore, the proof of this lemma is given by showing that

$$\begin{aligned} D^* &:= \arg \min_{\Upsilon \in \mathbb{R}^{n \times n}} \log|\Upsilon|, \\ &\text{subject to } \Upsilon \succeq D_i, \Upsilon \succeq D_j, \end{aligned} \quad (\text{D.7})$$

implies $D^* = D_\Gamma$. Notice that D_Γ is the obvious solution of (D.7) when $I_n \succeq D_j$ or $I_n \preceq D_j$ holds, as these cases result in $D_\Gamma = I_n$ or $D_\Gamma = D_j$, respectively. To prove the remaining cases, i.e., $I \not\succeq D_j$ and $I \not\preceq D_j$, a result of (Vashentsev, 2004) is recalled by considering the general case of two diagonal matrices $D_i, D_j \succ 0$, rather than $I_n, D_j \succ 0$. A first assumption defines a certain structure of D_i and D_j .

Assumption D.4.1 $\lfloor D_i \rfloor_{qq} < \lfloor D_j \rfloor_{qq}$, for all $q \in \mathbb{Z}_{[m+1,n]}$, and $\lfloor D_i \rfloor_{qq} \geq \lfloor D_j \rfloor_{qq}$, for all $q \in \mathbb{Z}_{[1,m]}$, hold for some $m \in \mathbb{Z}_{(0,n)}$.

Assumption D.4.1 is met for a suitable transformation of D_j , since $I \not\succeq D_j$ and $I \not\preceq D_j$. Further, given m as according to Assumption D.4.1, let us define $\mathcal{T} := \{T \in \mathbb{R}^{m \times n-m} | TT^\top \preceq I_m\}$ as a set of matrices and let the diagonal matrices $Q, R \in \mathbb{R}^{n \times n}$ and $\Sigma(T) \in \mathbb{R}^{n \times n}$ be defined conform to

$$\begin{aligned} \lfloor Q \rfloor_{qq} &:= \min\{\lfloor D_i \rfloor_{qq}, \lfloor D_j \rfloor_{qq}\}, \quad \forall q \in \mathbb{Z}_{[1,n]}, \\ \lfloor R \rfloor_{qq} &:= \sqrt{|\lfloor D_i \rfloor_{qq} - \lfloor D_j \rfloor_{qq}|}, \quad \forall q \in \mathbb{Z}_{[1,n]}, \\ \Sigma(T) &:= \begin{pmatrix} I_m & T \\ T^\top & I_{n-m} \end{pmatrix}, \quad T \in \mathcal{T}. \end{aligned} \quad (\text{D.8})$$

Then with $D(T) := Q + R\Sigma^{-1}(T)R$, for all $T \in \mathcal{T}$, Theorem 2 of (Vashentsev, 2004) states the following

Theorem D.4.2 *Let $D_i, D_j \succ 0$ be given. Then there does not exist a matrix $\bar{D} \neq D(T)$, for any $T \in \mathcal{T}$, such that $D_i \preceq \bar{D} \preceq D(T)$ and $D_j \preceq \bar{D} \preceq D(T)$.*

Combining the result of Theorem D.4.2 with Corollary 8.4.10 of (Bernstein, 2005), i.e., $\log|A| \leq \log|B|$ holds for any $A \preceq B$ and $A, B \succ 0$, gives that $D^* = D(T)$, for some $T \in \mathcal{T}$. As such, we can limit the search of (D.7) from $\Upsilon \in \mathbb{R}^{n \times n}$ to $\Upsilon = D(T)$, for all $T \in \mathcal{T}$, due to which this optimization problem can further be rewritten into

$$D^* = Q + R\Sigma^{-1}(T^*)R \quad \text{with} \quad T^* := \arg \min_{T \in \mathcal{T}} \log|D(T)|.$$

Notice that D_Γ characterized by (D.6) is equal to $Q + R^2$, for any $D_j \succ 0$ and $D_i = I_n$. Therefore, the proof is completed if $D^* = Q + R^2$, or similarly $T^* = 0_{m \times n-m}$, which is proven next.

The property that $\log|A| \leq \log|B|$ holds for any $0 \prec A \preceq B$ (Corollary 8.4.10 of (Bernstein, 2005)) gives that T^* satisfies $Q + R\Sigma^{-1}(T^*)R \preceq Q + R\Sigma^{-1}(T)R$ and thus $\Sigma(T^*) \succeq \Sigma(T)$, for all $T \in \mathcal{T}$. As such, T^* is equivalently obtained via $T^* = \arg \max_{T \in \mathcal{T}} \log|\Sigma(T)|$. Since $\Sigma(T) = \begin{pmatrix} I_m & T \\ T^\top & I_{n-m} \end{pmatrix}$, Fact 2.14.9 of (Bernstein, 2005) gives that $|\Sigma(T)| = |I_m - TT^\top| |I_{n-m}|$ and thus

$$T^* = \arg \max_{T \in \mathcal{T}} \log|I_m - TT^\top| \tag{D.9a}$$

$$= \arg \max_{T \in \mathcal{T}} \sum_{q=1}^m \lambda_q(I_m - TT^\top) \tag{D.9b}$$

$$= \arg \max_{T \in \mathcal{T}} \sum_{q=1}^m 1 - \lambda_q(TT^\top). \tag{D.9c}$$

As $0 \preceq TT^\top \preceq I_m$, it follows that $\lambda_q(TT^\top) \in \mathbb{R}_{[0,1]}$ for all $T \in \mathcal{T}$ and $q \in \mathbb{Z}_{[1,m]}$. Hence, the maximum in (D.9c) is attained at $\lambda_q(TT^\top) = 0$ for all $q \in \mathbb{Z}_{[1,m]}$, i.e., $T = 0_{m \times n-m}$, which completes the proof. \square

D.5 Proof of Theorem 4.4.7

Statement theorem:

For any $P_i, P_j \succ 0$, the covariance Γ conform to (4.15) is equal to $TD_\Gamma T^\top$.

Let us start by assuming that T is nonsingular. This means that T satisfies the hypothesis of Lemma 4.4.5. Therefore, substituting $\hat{P}_i = T^{-1}P_iT^{-\top}$, $\hat{P}_j = T^{-1}P_jT^{-\top}$ and $\hat{\Gamma} = T^{-1}\Gamma T^{-\top}$ into this lemma gives that $\Gamma = T\hat{\Gamma}T^\top$,

if $\hat{\Gamma} = \arg \min_{\hat{\Upsilon} \in \mathbb{R}^{n \times n}} \log |\hat{\Upsilon}|$, subject to $\hat{\Upsilon} \succeq \hat{P}_i$ and $\hat{\Upsilon} \succeq \hat{P}_j$. Since $\hat{P}_i = I_n$ and $\hat{P}_j = D_j$ are diagonal, the result of Lemma 4.4.6 gives that $\hat{\Gamma} = D_\Gamma$.

The lemma is thus proven if $T = S_i D_i^{\frac{1}{2}} S_j$ is nonsingular. Employing the results of Proposition 2.6.5 of (Bernstein, 2005), i.e., AB is nonsingular for any nonsingular matrices A and B , gives that T is nonsingular when all three $S_i, S_j, D_i^{\frac{1}{2}}$ are nonsingular. Note that each column of S_i is an eigenvector of $P_i \succ 0$, due to which Corollary 5.4.8 of (Bernstein, 2005) gives that $S_i^\top = S_i^{-1}$ and thus $S_i S_i^{-1} = I_n$ and $S_i^{-1} S_i = I_n$ hold. These latter two equalities imply that S_i is by definition a nonsingular matrix (see (Bernstein, 2005) for this definition). Similarly, S_j is nonsingular and since $P_i \succ 0$, the diagonal matrices $D_i \succ 0$ and $D_i^{\frac{1}{2}} \succ 0$ are nonsingular as well, which completes the proof. \square

\mathcal{E}

Proofs corresponding to Chapter 5

E.1 Proof of Lemma 5.2.4

Statement lemma:

Let each node i employ the CKF. Then $P_{i_f}[k] \preceq P_j[k]$ and $P_i[k+1] \preceq AP_j[k]A^\top + Q$ hold for all $j \in \mathcal{N}_{i(0,1)}$ and $k \in \mathbb{Z}_+$.

Let us first prove $P_i[k+1] \preceq AP_j[k]A^\top + Q$, for all $j \in \mathcal{N}_{i(0,1)}$ and $k \in \mathbb{Z}_+$, given that $P_{i_f}[k] \preceq P_j[k]$ holds for all $j \in \mathcal{N}_{i(0,1)}$ and $k \in \mathbb{Z}_+$. The algorithm of CKFs (Section 5.2.2) gives that $P_i^{-1}[k+1] = (AP_{i_f}[k]A^\top + Q)^{-1} + C_i^\top V_i^{-1} C_i$. As $V_i^{-1} \succ 0$, and thus $C_i^\top V_i^{-1} C_i \succeq 0$, it follows that $P_i[k+1] \preceq AP_{i_f}[k]A^\top + Q$. Therefore, if $P_{i_f}[k] \preceq P_j[k]$, for all $j \in \mathcal{N}_{i(0,1)}$ and $k \in \mathbb{Z}_+$, then also $P_i[k+1] \preceq AP_j[k]A^\top + Q$ holds for all $j \in \mathcal{N}_{i(0,1)}$ and $k \in \mathbb{Z}_+$.

The second part shows that $P_{i_f}[k] \preceq P_j[k]$, for all $j \in \mathcal{N}_{i(0,1)}$ and $k \in \mathbb{Z}_+$. From the algorithm of CKFs (Section 5.2.2) it follows that this inequality holds if $P_{i(l)} \preceq P_{j(l)}$ and $P_{i(l)} \preceq P_{i(l-1)}$, for all $l \in \mathbb{Z}_{[1,L]}$. Notice that in the l -th fusion-cycle, i.e., with $P_{i(l-1)}$ and $P_{j(l)}$ as the prior covariances, $P_{i(l)}$ represents the resulting fused covariance. Hence, substituting $P_i = P_{i(l-1)}$, $P_j = P_{j(l)}$ and $P_{i_f} = P_{i(l)}$ into Proposition 5.2.1 gives that $P_{i(l)} \preceq P_{j(l)}$ and $P_{i(l)} \preceq P_{i(l-1)}$, which completes the proof. \square

E.2 Proof of Lemma 5.2.5

Statement lemma:

Let each node i employ the CKF. Then for any $c \in \mathbb{Z}_{\geq 1}$ the following holds

$$P_{i_f}[k] \preceq A^c P_j[k-c] (A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top, \quad \forall k \in \mathbb{Z}_{\geq c}, j \in \mathcal{N}_{i(c+1)}. \quad (\text{E.1})$$

Before the proof is presented, let $\mathcal{N}_{h(1)}$ denote the set of direct neighboring nodes for a given node $h \in \mathcal{M}$, in which $\mathcal{M} \subset \mathcal{N}$. Then $\{\cup_{h \in \mathcal{M}} \mathcal{N}_{h(0,1)}\}$ denotes the set of all nodes $h \in \mathcal{M}$ and all nodes j that are a direct neighbor of any node $h \in \mathcal{M}$. The proof of (E.1) proceeds by induction.

Step one of induction employs $c = 1$, i.e., $P_{i_f}[k] \preceq AP_j[k-1]A^\top + Q$ should hold for all $j \in \mathcal{N}_{i(2)}$ and $k \in \mathbb{Z}_{\geq 1}$. Let a node $h \in \mathcal{N}_{i(1)}$ be given, such that $j \in \mathcal{N}_{h(1)}$. Then substituting $i = h$ and $k = k-1$ into Lemma 5.2.4, yields

$$P_h[k] \preceq AP_j[k-1]A^\top + Q, \quad \forall j \in \mathcal{N}_{h(0,1)}, k \in \mathbb{Z}_{\geq 1}. \quad (\text{E.2})$$

Since node $h \in \mathcal{N}_{i(1)}$ is a direct neighbor of node i , Lemma 5.2.4 gives that $P_{i_f}[k] \preceq P_h[k]$ holds for all $h \in \mathcal{N}_{i(1)}$ and $k \in \mathbb{Z}_+$. Combining this inequality with (E.2) induces

$$P_{i_f}[k] \preceq AP_j[k-1]A^\top + Q, \quad \forall j \in \{\cup_{h \in \mathcal{N}_{i(1)}} \mathcal{N}_{h(0,1)}\}, \quad k \in \mathbb{Z}_{\geq 1}. \quad (\text{E.3})$$

The condition $j \in \{\cup_{h \in \mathcal{N}_{i(1)}} \mathcal{N}_{h(0,1)}\}$ means that (E.3) holds for all nodes j , such that the graph distance $d(v_i, v_j) \leq 2$. Hence, $\{\cup_{h \in \mathcal{N}_{i(1)}} \mathcal{N}_{h(0,1)}\} = \{\cup_{q \leq 2} \mathcal{N}_{i(q)}\}$ and since $\mathcal{N}_{i(2)} \subset \{\cup_{q \leq 2} \mathcal{N}_{i(q)}\}$, (E.3) holds for all $j \in \mathcal{N}_{i(2)}$.

The second step of induction is to show that (E.1) holds for any $c \geq 2$, by assuming that (E.1) holds for $c-1$. This latter assumption gives that the following inequality is met, for all nodes $h \in \mathcal{N}_{i(c)}$ and $k \in \mathbb{Z}_{\geq c-1}$, i.e.,

$$P_{i_f}[k] \preceq A^{c-1}P_h[k-(c-1)](A^{c-1})^\top + \sum_{q=0}^{c-2} A^q Q (A^q)^\top. \quad (\text{E.4})$$

Substituting $i = h$ and $k = k-c$ into Lemma 5.2.4, while observing that $k-c+1 = k-(c-1)$, gives that at $P_h[k-(c-1)] \preceq AP_j[k-c]A^\top + Q$ holds for all $j \in \mathcal{N}_{h(0,1)}$ and $k-c \in \mathbb{Z}_+$. Based on this inequality one can rewrite (E.4) into

$$\begin{aligned} P_{i_f}[k] &\preceq A^{c-1}(AP_j[k-c]A^\top + Q)(A^{c-1})^\top + \sum_{q=0}^{c-2} A^q Q (A^q)^\top \\ &= A^c P_j[k-c](A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top, \quad \forall j \in \{\cup_{h \in \mathcal{N}_{i(c)}} \mathcal{N}_{h(0,1)}\}, \quad k \in \mathbb{Z}_{\geq c} \end{aligned}$$

Similar as to the first step, $\mathcal{N}_{i(c+1)} \subset \{\cup_{h \in \mathcal{N}_{i(c)}} \mathcal{N}_{h(0,1)}\}$ implies that the above inequality holds for all $j \in \mathcal{N}_{i(c+1)}$, which completes the proof. \square

E.3 Proof of Lemma 5.2.6

Statement lemma:

Let each node i employ the CKF. Then, $P_i[k] \preceq \Sigma_i[k]$ holds for all $k \in \mathbb{Z}_+$ and $P_i[-1] = \Sigma_i[-1]$, where $\Sigma_i[k]$ satisfies the RDE in (5.4)

The above statement is proven by induction. To clarify the derivation, let us recall the update formulas of P_{i_f} and Σ_i for all $k \in \mathbb{Z}_{\geq 1}$, i.e.,

$$\begin{aligned} P_i^{-1}[k] &= (AP_{i_f}[k-1]A^\top + Q)^{-1} + C_i^\top V_i^{-1} C_i, \\ \Sigma_i^{-1}[k] &= (A\Sigma_i[k-1]A^\top + Q)^{-1} + C_i^\top V_i^{-1} C_i. \end{aligned} \quad (\text{E.5})$$

The first step of induction employs $k = 0$, due to which $P_i[0] \preceq \Sigma_i[0]$ should be proven. Note that this property follows directly from the fact that

P_{i_f} and Σ_i have similar updates, as it is shown in (E.5), and $P_i[-1] = \Sigma_i[-1]$. Or more precisely, $P_i[0] = \Sigma_i[0]$ holds, which completes the first step.

The second step is to prove that $P_i[k] \preceq \Sigma_i[k]$, for all $k \in \mathbb{Z}_{\geq 1}$, if $P_i[k-1] \preceq \Sigma_i[k-1]$ holds. The result of Lemma 5.2.4, i.e., $P_{i_f}[k] \preceq P_i[k]$, gives that $P_i[k-1] \preceq \Sigma_i[k-1]$ induces $P_{i_f}[k-1] \preceq \Sigma_i[k-1]$, which is further rewritten into $(AP_{i_f}[k-1]A^\top + Q)^{-1} \succeq (A\Sigma_i[k-1]A^\top + Q)^{-1}$. Using this inequality in (E.5) implies that $P_i^{-1}[k] \succeq \Sigma_i^{-1}[k]$ and thus $P_i[k] \preceq \Sigma_i[k]$ completes the proof. \square

E.4 Proof of Theorem 5.2.7

Statement lemma:

Let each node i perform the CKF and let $\mathcal{N}_{\text{RDE}} \neq \emptyset$. Then, $P_{i_f}[\infty]$ exists and satisfies $P_{i_f}[\infty] \preceq \bar{\Sigma}_{i,j}[\infty]$, for all $j \in \mathcal{N}_{\text{RDE}}$.

For clarity of exposition, let us recall the following characterization:

$$\bar{\Sigma}_{i,j}[k] = \begin{cases} \Sigma_j[k] & \text{if } j \in \mathcal{N}_{i(0,1)}, k \in \mathbb{Z}_+, \\ A^c \Sigma_j[k-c](A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top & \text{if } j \in \mathcal{N}_{i(c+1)}, k \in \mathbb{Z}_{\geq c}. \end{cases}$$

Then above statement is proven by showing that $P_{i_f}[k] \preceq \bar{\Sigma}_{i,j}[k]$ holds for all $j \in \mathcal{N}$, due to which $P_{i_f}[\infty] \preceq \bar{\Sigma}_{i,j}[\infty]$ will hold for all $j \in \mathcal{N}_{\text{RDE}}$. Further, in line with the above characterization of $\bar{\Sigma}_{i,j}[k]$, let us distinguish the two cases $j \in \mathcal{N}_{i(0,1)}$ and $j \in \mathcal{N}_{i(c+1)}$, for some $c \in \mathbb{Z}_{\geq 1}$.

In the first case, i.e., $j \in \mathcal{N}_{i(0,1)}$, we have that $\bar{\Sigma}_{i,j}[k] = \Sigma_j[k]$, due to which $P_{i_f}[k] \preceq \Sigma_j[k]$ should hold for all $k \in \mathbb{Z}_+$. The results of Lemma 5.2.4 and Lemma 5.2.6 give that $P_{i_f}[k] \preceq P_j[k]$ and $P_j[k] \preceq \Sigma_j[k]$ hold for all $j \in \mathcal{N}_{i(0,1)}$ and $k \in \mathbb{Z}_+$. Combining these inequalities gives the desired inequality $P_{i_f}[k] \preceq \Sigma_j[k]$, for all $j \in \mathcal{N}_{i(0,1)}$ and $k \in \mathbb{Z}_+$. Let us continue with its asymptotic properties by rewriting this expression as $P_{i_f}[k] - \Sigma_j[k] \preceq 0$, and thus,

$$P_{i_f}[k] - \Sigma_j[k] + \Sigma_j[\infty] - \Sigma_j[\infty] \preceq 0, \quad \forall j \in \{\mathcal{N}_{i(0,1)} \cap \mathcal{N}_{\text{RDE}}\}, k \in \mathbb{Z}_+. \quad (\text{E.6})$$

Proposition 5.2.2 gives that $\lim_{k \rightarrow \infty} (\Sigma_j[k] - \Sigma_j[\infty]) = 0$. Hence, if $k \rightarrow \infty$, then (E.6) becomes $\lim_{k \rightarrow \infty} (P_{i_f}[k] - \Sigma_j[\infty]) \preceq 0$. As $\Sigma_j[\infty]$ is a constant, this inequality is equivalent to $\lim_{k \rightarrow \infty} P_{i_f}[k] - \Sigma_j[\infty] \preceq 0$ and thus,

$$P_{i_f}[\infty] \preceq \Sigma_j[\infty], \quad \forall j \in \{\mathcal{N}_{i(0,1)} \cap \mathcal{N}_{\text{RDE}}\}. \quad (\text{E.7})$$

The second case, i.e., $j \in \mathcal{N}_{i(c+1)}$ and $c \in \mathbb{Z}_{\geq 1}$, implies that $P_{i_f}[k] \preceq A^c \Sigma_j[k-c](A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top$ should hold for all $k \in \mathbb{Z}_{\geq c}$. Note that, for all $c \in \mathbb{Z}_{\geq 1}$ and $k \in \mathbb{Z}_{\geq c}$, the result of Lemma 5.2.5, yields

$$P_{i_f}[k] \preceq A^c P_j[k-c](A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top, \quad \forall j \in \mathcal{N}_{i(c+1)}.$$

When substituting the result of Lemma 5.2.6, i.e., $P_j[k-c] \preceq \Sigma_j[k-c]$, into the above inequality gives for all $c \in \mathbb{Z}_{\geq 1}$ and $k \in \mathbb{Z}_{\geq c}$ that

$$P_{i_f}[k] \preceq A^c \Sigma_j[k-c] (A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top, \quad \forall j \in \mathcal{N}_{i(c+1)}.$$

Note that the above expression yields the desired inequality of this second step. Therefore, let us continue with its asymptotic properties, next, by observing that $c < \infty$ induces $\lim_{k \rightarrow \infty} k-c = \infty$. Then, similarly as to (E.6) and (E.7) one can derive that

$$P_{i_f}[\infty] \preceq A^c \Sigma_j[\infty] (A^c)^\top + \sum_{q=0}^{c-1} A^q Q (A^q)^\top, \quad \forall j \in \mathcal{N}_{i(c+1)} \cap \mathcal{N}_{\text{RDE}},$$

holds for any $c \in \mathbb{Z}_{\geq 1}$, which completes the proof. \square

\mathcal{F}

Proofs corresponding to Chapter 6

F.1 Proof of Lemma 6.3.3

For clarity, let us define $X_i := P_i^{\text{DIF}}$ and $Y_i := P_i^{\text{CIF}}$. Then to prove the inequality $X_i[\infty] \preceq Y_i[\infty]$, for all $i \in \mathcal{N}$, let us prove that $X_i[\mathbf{k}] \preceq Y_i[\mathbf{k}]$ holds for all $i \in \mathcal{N}$ and all $\mathbf{k} \in \mathbb{Z}_+$ by induction. From the update of the DIF and CIF and the fact that $L = 1$ one obtains that

$$\begin{aligned} X_i^{-1}[\mathbf{k}] &= (AX_i[\mathbf{k}-1]A^\top + Q)^{-1} + \sum_{j \in \mathcal{N}_i(0,1)} Z_j[\mathbf{k}], \\ Y_i^{-1}[\mathbf{k}] &= (AY_i[\mathbf{k}-1]A^\top + Q)^{-1} + \sum_{j \in \mathcal{N}_i(0,1)} W_{ij}Z_j[\mathbf{k}]. \end{aligned} \tag{F.1}$$

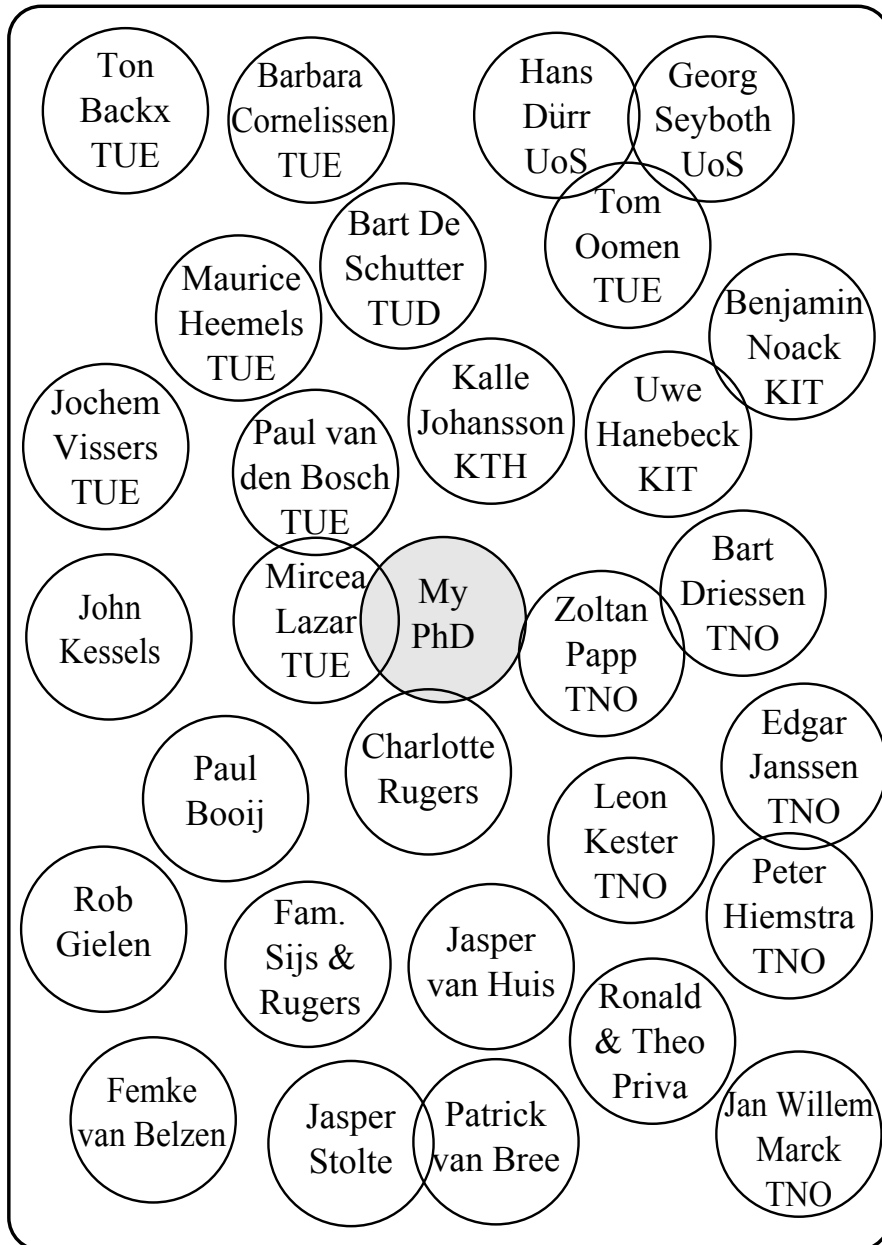
Moreover, since $\sum_{j \in \mathcal{N}_i(0,1)} W_{ij} = 1$, the following inequality also holds

$$\sum_{j \in \mathcal{N}_i(0,1)} Z_j[\mathbf{k}] \succeq \sum_{j \in \mathcal{N}_i(0,1)} W_{ij}Z_j[\mathbf{k}]. \tag{F.2}$$

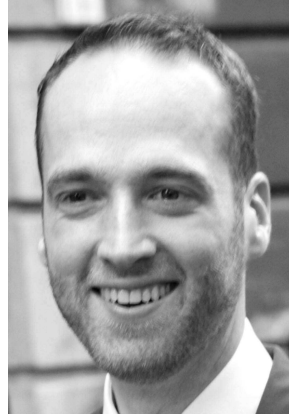
Then the first is step to prove that $X_i[1] \preceq Y_i[1]$. Starting from $X_i[0] = Y_i[0]$ one can derive that $(AX_i[1]A^\top + Q)^{-1} = (AY_i[1]A^\top + Q)^{-1}$. Combining this equality with the inequality of (F.2) gives that the update equations of (F.1) are such that $X_i^{-1}[1] \succeq Y_i^{-1}[1]$ holds and thus $X_i[1] \preceq Y_i[1]$.

The second step is to prove that $X_i[\mathbf{k}] \preceq Y_i[\mathbf{k}]$, if $X_i[\mathbf{k}-1] \preceq Y_i[\mathbf{k}-1]$ holds. Starting from the latter inequality one can derive that $(AX_i[\mathbf{k}-1]A^\top + Q)^{-1} \succeq (AY_i[\mathbf{k}-1]A^\top + Q)^{-1}$. Combining this equality with the inequality of (F.2) gives that the update equations of (F.1) are such that $X_i^{-1}[\mathbf{k}] \succeq Y_i^{-1}[\mathbf{k}]$ and thus $X_i[\mathbf{k}] \preceq Y_i[\mathbf{k}]$, which completes the proof. \blacksquare

Acknowledgements



Curriculum Vitae



Joris Sijs was born on February 19, 1981 in Axel, The Netherlands.

He received the Master of Science degree in the Control Systems Group from the Faculty of Electrical Engineering, Technical University of Eindhoven, The Netherlands in 2006.

Since January 2006 he has been an employee of TNO (Dutch Organization of Applied Sciences). Among others, his main interests are state estimation and control, event sampling strategies, self-organizing systems and popularization of science. He pursued his Ph.D. degree from September 2007 to April 2012 with the Control Systems Group at the Faculty of Electrical Engineering, Eindhoven University of Technology, The Netherlands, under the supervision of Prof. Paul van den Bosch. The research conducted in this period is presented in this dissertation.