Karlsruher Institut für Technologie

Fakultät für Informatik
Institut für Anthropomatik und Robotik
Lehrstuhl für Intelligente Sensor-Aktor-Systeme (ISAS)
Prof. Dr.-Ing. Uwe D. Hanebeck

Intelligente Sensor-Aktor-Systeme

Bachelorarbeit

Event-based State Estimation in Multisensor Systems.

Teresa Giner Ruiz

26. August 2019

Referent:	Prof. Dr.-Ing. Uwe D. Hanebeck
Korreferent:	Prof. Dr.-Ing. Tamim Asfour
Betreuer:	Dr.-Ing. Benjamin Noack

Abstract

Nowadays sensors are implemented in countless of actual scenarios ranging from security to entertainment applications. They generate a huge amount of transmissions within the network they belong to, resulting in a costly communication effort. In order to optimize the transmission process, an event-based system – instead of the conventional periodic approach – should be used. In this sense, several challenges appear when multiple sensors are involved at the same time, where new issues about their event-criteria arise, i.e., how could sensors compare their observations to make a transmission decision. To this effect, communication between sensor nodes is to be studied seeking to utilize information in a profitable way. In this work, different multisensor network structures are to be compared, i.e., star, chain, and hierarchical topologies. Finally, quality will be deeply discussed in terms of estimation's quality degradation due to the proposed joint trigger criteria as compared to independent event triggers.

Eidesstattliche Erklärung

Hiermit erkläre ich, die vorliegende Bachelorarbeit selbstständig angefertigt zu haben. Die verwendeten Quellen sind im Text gekennzeichnet und im Literaturverzeichnis aufgeführt.

Mª Teresa Giner

Karlsruhe, 26. August 2019

Teresa Giner Ruiz

Contents

List of Figures

Notation

Conventions

x	Scalar
$\boldsymbol{x}$	Random variable
$\hat{x}$	Mean of random variable $\boldsymbol{x}$.
$\underline{x}$	Column vector
$\underline{\boldsymbol{x}}$	Random vector
$\underline{\hat{x}}$	Mean of random vector $\underline{\boldsymbol{x}}$.
$\mathbf{A}$	Matrix
$(.)_k$	Quantity at time step k.
$\mathbb{R}$	Set of real numbers.
$\sim$	Distribution operator. E.g., $\boldsymbol{x} \sim \mathcal{U}$ means $\boldsymbol{x}$ is distributed according to $\mathcal{U}$.

Abbreviations

MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error
NEES	Normalized Estimation Error Squared
AWGN	Additive White Gaussian Noise

CHAPTER 1

Introduction

Our century has evolved towards technology involving Internet and telecommunications in every single step that we take. Our daily life could not be anymore understood without new technologies, not only playing an important role in our lives but also being indispensable: what would we do right now without them?

Networks are the actual foundation for our modern digitalized world, working silently in the background as a huge and apparently infinite infrastructure. Communications, Internet, everything runs over networks. This work especially focuses on its information generators, the actual agents that produce thousands of millions of data units: sensors.

Sensors are the starting points of networks, the information collecters. They measure real world values to be further on processed on control elements or intelligent units. Sensors can communicate between each other, with other network elements, or even with humans. The goal of this research is to improve the communication between sensors and its processing units, reducing the transmission rate in order to save constrained resources such as computing power, channel bandwidth, or energy.

As we know, changes in real world do not happen periodically which means that transmitting sensor's information in this way leads to an unjustified waste of the above mentioned resources. Therefore an event-based approach was proposed in [1], where transmissions only take place when significant real changes occur. This new idea changes completely previous research directions and turns to a completely different strategical point of view: how to accurately define an event, such that no important information is missed and actions are still taken in time. For this purpose, various triggers for data transmission were examined, which are to represent the largest possible information gain at low communication rate for the control centre, without significantly worsening the result of the state estimation.

At times when no event is triggered and therefore no transmission is received at the control centre, still some information, called negative information, can be used since the absence of new information reflects that the new sensor's measurement is not as relevant as to be considered an event.

Not only is the trigger paradigm to be studied but also the sensors' topology. In [7], the difference between a centralized, a distributed, and a decentralized scheme was studied, with the purpose of showing the optimal system according to the underlying measurements dependency structure. Whether to implement the complete algorithm on each node or to distribute the workload among them is a decision to consider, as well as which fusion strategy to chose. Sensor network's research always pursues to exploit all information available in the system and therefore the sensor nodes' structure is the main issue to be studied in this work.

In this paper, approaches to the extension of the single sensor event-based estimation method from [11] will be investigated, which enable a better triggering decision by means of a node's cooperation and maximizing the usage of information. First, this requires the modeling of an ideal communication channel between a sensor and a control centre, which is supposed to have no packages loss. Furthermore, a trigger method is investigated to achieve the most effective data transmission possible, resulting in a low estimation error at low average transmission rate.

CHAPTER 2

Basics

This research is based on state estimation principles, event-based systems, and basic network knowledges. In this section, the basics of each topic will be explain in order to fully comprehend the scope of the present thesis. Further details can be found in the literature presented at the end of the document.

2.1 State Estimation

Stochastic processes can be used to describe real-world dynamic systems, where random events make an impact on the evolution of the state variables throughout time. These hidden and not directly detectable variables are constantly being measured by sensors, obtaining inaccurate and uncertain observations corresponding to the actual system value. By combining both the measurements and the known system model, a conclusion about the actual value of the system can be drawn. State estimation principles and the basic ideas about the Kalman filter implementation are described below in this section.

According to [5], given a discrete-time signal $\underline{z}_k$ we can distinguish the real signal term $\underline{s}_k$ and the noise term $\underline{n}_k$ in every measurement taking place, which is expressed as

$$\underline{z}_k = \underline{s}_k + \underline{n}_k$$

understanding $\underline{z}_k$ like an estimation for the real value $\underline{s}_k$. A suitable filter should then be developed such that it minimizes the error $\underline{e}_k$ between both variables defined as

$$\begin{aligned} \underline{e}_k &= \underline{z}_k - \underline{s}_k \,, \\ \underline{e}_k^2 &= (\underline{z}_k - \underline{s}_k)^{\mathrm{T}} \cdot (\underline{z}_k - \underline{s}_k). \end{aligned}$$

This constitutes the *Minimum Mean Squared Error* (MMSE) filter. This filter satisfies the known orthogonality principle which states that the estimation error $\underline{e}_k$ and the signal $\underline{z}_k$ are orthogonal, i.e., it holds

$$\underline{e}_k = \underline{z}_k - \hat{\underline{z}}_k \,,$$
$$\mathrm{E}\left\{\underline{e}_k \, \underline{z}_k^{\mathrm{T}}\right\} = 0 \,, \forall k \,.$$

According to Wiener in [12], it is in fact equivalent to the optimal filter denoted as

$$\mathrm{E}\left\{\underline{s}_k \, \underline{z}_i^{\mathrm{T}}\right\} = \sum_{j=0}^{k} h_{k,j} \mathrm{E}\left\{\underline{z}_j \underline{z}_i^{\mathrm{T}}\right\} \quad i = 0, 1, \ldots, k$$

where $h_{n,j}$ denote the filter coefficients to be determined. However, this solution is constrained to a non-recursive causal system.

Additionally, a non-recursive filter may then be easily developed in order to minimize $\underline{e}_k$, what leads to the commonly used inverse of the autocorrelation matrix $\mathbf{R}_{zz}$ of the measurements' noise as a filter.

$$\underline{h}_{\mathrm{MMSE}} = \mathbf{R}_{zz}^{-1} \, \underline{r}_{ss} \,,$$

is commonly known as the Wiener-Hopf equation, where $\underline{r}_{ss}$ alludes to the autocorrelation vector of the reference signal $\underline{s}$. However, this approach implies a high computational effort as the number of measurements N increase. Thus, a recursive solution would be desirable. In addition to this, not all measurements are usually available from the beginning, which leads to the need of an update of the state estimate among time.

In the following, a recursive approach is presented achieving an efficient state estimation with constant computational effort.

2.1.1 Kalman Filter

The method of estimating the state, published by R. E. Kalman in 1960 [4], has been widely applied in a large variety of fields like navigation, dead reckoning (i.e., object tracking), and signal processing. Nowadays, it is proven to be the optimal estimator in the linear Gaussian case, although this is not an assumption to be done for the filter to be applicable. Otherwise, it constitutes the best linear estimator, since there are non-linear approaches that may produce better results.

The Kalman filter estimations are updated using a weighted average of the present input observation and the previously calculated state obtaining the desired recursive approach. It is based on the one hand on a process system described by a linear model, such as

$$\underline{\boldsymbol{x}}_{k+1} = \mathbf{A}_k \, \underline{\boldsymbol{x}}_k + \mathbf{B}_k \, \underline{\boldsymbol{w}}_k \,, \tag{2.1}$$

whereas on the other hand, the observation model, i.e., the sensor's measurements, is linked to the actual state $\underline{\boldsymbol{x}}_k$ by means of

$$\underline{\boldsymbol{z}}_k = \mathbf{C}_k \, \underline{\boldsymbol{x}}_k + \underline{\boldsymbol{v}}_k \,, \tag{2.2}$$

where both process and measurement noise terms $\underline{\boldsymbol{w}}_k$ and $\underline{\boldsymbol{v}}_k$ are supposed to be mutually uncorrelated and independent of the initial state for any arbitrary k, as well as normally distributed with 0 mean and covariance matrices $\mathbf{W}$ and $\mathbf{R}$, respectively. Likewise, $\mathbf{A}_k$ and $\mathbf{C}_k$ are the process and the observation matrices and are supposed to be detectable throughout the network. $\mathbf{B}_k$, for its part, is used to map the system noise $\underline{\boldsymbol{w}}_k$ to each of the individual components of the state variable $\underline{\boldsymbol{x}}_k$. In the following sections, research is restricted to time-invariant systems and hence these matrices will be recalled as $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$.

The initial state $\underline{x}_0$ has a mean value of $\hat{\underline{x}}_0$ and a covariance matrix $\mathbf{P}_0$. For the first and second moments of the noise processes the following considerations apply

$$\begin{aligned}
\mathrm{E}\{\underline{\boldsymbol{v}}_k\} &= 0 \,, \\
\mathrm{E}\{\underline{\boldsymbol{w}}_k\} &= 0 \,, \\
\mathrm{E}\big\{\underline{\boldsymbol{v}}_k \, \underline{\boldsymbol{v}}_j^{\mathrm{T}}\big\} &= \mathbf{R} \, \delta_{kj} \,, \\
\mathrm{E}\big\{\mathbf{B}_k \, \underline{\boldsymbol{w}}_k \, \underline{\boldsymbol{w}}_j^{\mathrm{T}} \, \mathbf{B}_k^{\mathrm{T}}\big\} &= \mathbf{Q} \, \delta_{kj} \,, \\
\mathrm{E}\big\{\underline{\boldsymbol{v}}_k \, \underline{\boldsymbol{w}}_j^{\mathrm{T}}\big\} &= 0 \,, \\
\forall k, j \,.
\end{aligned}$$

The goal of the remote estimator is to determine an optimal estimate $\hat{\underline{x}}_k$ of the actual state $\underline{x}_k$ in the *Minimum Mean-Square Error* (MMSE) sense based on the history of the different measurements $\underline{z}_0$, $\underline{z}_1$, ..., $\underline{z}_k$.The key of the recursive method lies in the fact that, although the current estimation $\hat{\underline{x}}_k$ does depend on all past state estimations $\hat{\underline{x}}_0$, $\hat{\underline{x}}_1$, . . ., $\hat{\underline{x}}_{k-1}$ and on all measurements available until now $\underline{z}_0$, $\underline{z}_1$, ..., $\underline{z}_k$, the Kalman filter provides an estimation directly from the last computed $\hat{\underline{x}}_{k-1}$ and the current observation $\underline{z}_k$.

The implementation consists of tracking the pair of parameters $(\hat{\underline{x}}_k, \mathbf{P}_k)$, where $\mathbf{P}_k$ is the error covariance matrix corresponding to the estimate at time k, defined like

$$\mathbf{P}_k \; = \mathrm{E}\left\{(\hat{\underline{x}}_k - \underline{x}_k)(\hat{\underline{x}}_k - \underline{x}_k)^{\mathrm{T}}\right\} .$$

Its trace yields the mean-squared estimation error. In essence, the Kalman filter scheme is compound of the combination of measurement information with prior information at each time k, executed as a two-step algorithm: prediction and filtering. The prediction step employs previous information based on the process model (2.1), while the update step makes use of the measurement model (2.2).

The prediction result $(\hat{\underline{x}}_k^-, \mathbf{P}_k^-)$ is computed according to

$$\begin{aligned} \hat{\underline{x}}_k^- &= \mathbf{A}\,\hat{\underline{x}}_{k-1}\,, \\ \mathbf{P}_k^- &= \mathbf{A}\,\mathbf{P}_{k-1}\,\mathbf{A}^{\mathrm{T}} + \mathbf{Q}\,. \end{aligned} \tag{2.3}$$

These two values serve later on as the prior information used by the filter in order to update an estimation with the aid of a new measurement $\underline{z}_k$,

$$\begin{aligned} \hat{\underline{x}}_k &= \hat{\underline{x}}_k^- + \mathbf{K}_k\,(\underline{z}_k - \mathbf{C}\,\hat{\underline{x}}_k^-)\,, \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\,\mathbf{C})\,\mathbf{P}_k^-\,(\mathbf{I} - \mathbf{K}_k\,\mathbf{C})^{\mathrm{T}} + \mathbf{K}_k\,\mathbf{R}\,\mathbf{K}_k^{\mathrm{T}} \\ &= (\mathbf{P}_k^{-1} + \mathbf{C}^{\mathrm{T}}\,\mathbf{R}^{-1}\mathbf{C})^{-1} \\ &= (\mathbf{I} - \mathbf{K}_k\,\mathbf{C})\,\mathbf{P}_k^-\,. \end{aligned} \tag{2.4}$$

In (2.4), the Kalman gain corresponds to the expression

$$\mathbf{K}_k = \mathbf{P}_k^-\,\mathbf{C}^{\mathrm{T}}\,[\mathbf{C}\,\mathbf{P}_k^-\mathbf{C}^{\mathrm{T}} + \mathbf{R}]^{-1}$$

and ensures an optimal combination of both current measurement and prior prediction, suitably weighted by the Kalman gain $\mathbf{K}_k$ according to their certainty. In other words, values with lower uncertainty will be weighted heavier, what allows the model to dynamically trust the prediction model or the new measurement differently.

The final result is a new estimate which lies between the prior prediction and the measured state and has a better estimated uncertainty than either of them alone. To sum up, the implementation of both steps in real systems provide the accomplishment of reliable estimations through inaccurate variables, which provides the Kalman filter of an enormous helpfulness in countless applications.

2.1.2 Information Form

The steps described in Section 2.1.1 can be stated in different formulations regarding its purpose. The inverse covariance filter, commonly known as the information filter, is based on the conversion of both parameters ($\hat{\underline{x}}_k$, $\mathbf{P}_k$) to a pair of terms that directly refer to the information gained across the network, i.e., the information vector and the information matrix ($\hat{\underline{y}}_k$, $\mathbf{Y}_k$). These are related to the original states and error covariance matrix by means of

$$\begin{aligned}\hat{\underline{y}}_k &= \mathbf{P}_k^{-1}\hat{\underline{x}}_k,\\ \mathbf{Y}_k &= \mathbf{P}_k^{-1}.\end{aligned} \tag{2.5}$$

On the other hand, sensors will then have to transform their raw measurements $\underline{z}_k$ into their own information term and information matrix properly, as shown in the following equation

$$\begin{aligned}\underline{i}_k &= \mathbf{C}_k^{\mathrm{T}} \cdot \mathbf{R}_k^{-1} \cdot \underline{z}_k,\\ \mathbf{I}_k &= \mathbf{C}_k^{\mathrm{T}} \cdot \mathbf{R}_k^{-1} \cdot \mathbf{C}_k.\end{aligned} \tag{2.6}$$

If we now rewrite the prediction step in terms of the information vector and the information matrix

$$\begin{aligned}\mathbf{Y}_k^- &= \mathbf{A}\mathbf{Y}_{k-1}^{-1}\mathbf{A}^{\mathrm{T}} + \mathbf{B}\mathbf{W}\mathbf{B}^{\mathrm{T}},\\ \hat{\underline{y}}_k^- &= \mathbf{Y}_k^-\mathbf{A}\mathbf{Y}_{k-1}^{-1}\hat{\underline{y}}_{k-1}^-,\end{aligned} \tag{2.7}$$

then the update step becomes a trivial sum

$$\begin{aligned}\mathbf{Y}_k &= \mathbf{Y}_{k-1} + \mathbf{I}_k,\\ \hat{\underline{y}}_k &= \hat{\underline{y}}_{k-1} + \underline{i}_{k-1},\end{aligned} \tag{2.8}$$

where both Equations (2.7) and (2.8) are equivalent to the ones in the previous section (2.3) and (2.4) respectively. The major among various reasons to study this formulation of the Kalman filter in this research is the decrease of complexity when handling new measurements in the update step (2.8). Furthermore, since mathematically it is still equivalent to a linear system such as the one described in Equation (2.1), linear properties are still preserved and assumptions taken into consideration up to this point can be still made. Nevertheless, the time step prediction entails a higher complexity and thus it will be still carried out by means of the state variable.

Due to the further focus on multisensor network scenarios, this new form of expressing the Kalman filter is considered to introduce a high gain from a computational point of view since this time multiple observations will have to be treated simultaneously. In Section 4, a deep discussion of the use of the information form for each of the different proposals is explained and presents the major advantages and eases that the information form brings to this research.

In Section 2.2, an introduction to event-based transmission is displayed, where unlike so far new measurements do not take place on every time step and therefore a modification to the previous set up scenario must be done.

2.2 Event-based Transmission

Since the goal of this work is to optimize the transmission between sensors inside a network, an event-based approach must be taken into consideration. Any system concerning an event-triggering technique reduces considerably the communication rate, what allows these systems to still achieve good results by means of fewer transmissions. However, estimation performance may loose therefore accuracy, since new sensor information is not available on each time step but only on those when events take place. In this section, an overview on existing methods for event-transmission is provided, with different trigger criteria already studied and their corresponding state estimation modifications overcoming the lack of information at non-event time frames.

Event-based transmission is specifically applied between the sensor and the estimator, where measurements are typically periodically sent. As real variables do not change continuously in time, sending its measurements at periodic time steps inevitably leads to a low information gain per transmission from the receiver point of view, where the state estimation will then be performed. To overcome this drawback, the sensor is extended by a so-called trigger, an additional transmission procedure which decides whether the present observation should be sent or not. For computational effort purposes, sensors are supposed to keep only their last sent measurement, as well as no acknowledgment information or other bilateral communication are considered in this work. Hence, the transmission decision is made at each time step based only on internal information of the transmitter without the use of any additional information from the estimator, from now on also indistinctly referred to as receiver.

Among all trigger implementations previously studied, two main groups can be distinguished: deterministic and stochastic triggers. In the following subsections

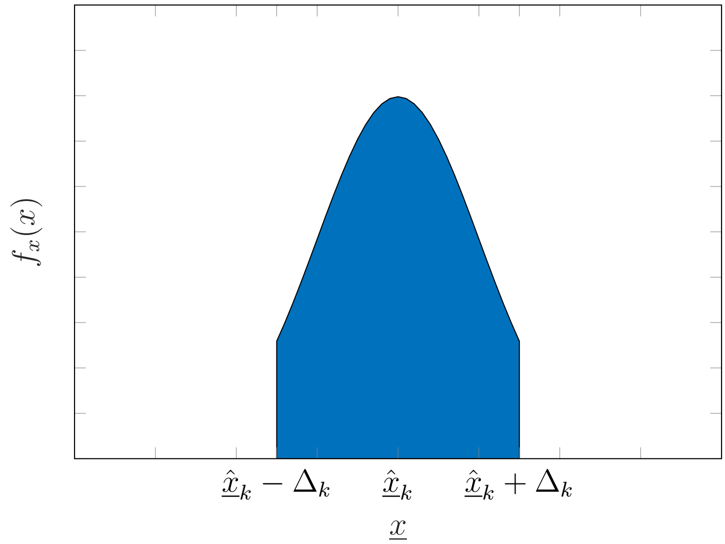

Figure 2.1: Violation of the Gaussion distribution of the state variable. Image from Eva J. Schmitt in [9].

the most common formulations are to be explained.

2.2.1 Deterministic Trigger

Deterministic triggers are the simplest event-based approach where a fixed threshold is designed to make the decision and is constant during all time steps. This implies on the one hand that the same measurements values will always give out the exact same transmission decision, independent of the context and on the other hand that there is no degree of uncertainty whether a transmission will take place or not. Since using only the new measurement value to make the transmission decision would be inefficient, depending on how the compared variable against the threshold is defined we can distinguish different deterministic trigger types. It is worth stressing that the incorporation of a deterministic trigger in a network automatically infers the violation of the desired Gaussian distribution in the state variable due to the fact that not all values of the state variables can be achieved by means of a fixed decision boundary. Figure 2.1 shows how the state variable distribution would look like in this scenario. For all these reasons, a stochastic trigger will be used further in this research and its fundamentals can be found in Section 2.2.2.

In this section, three main approaches are to be studied according to Dawei Shi, Ling Shi, and Tongwen Chen in [10] and the principles explained in [11] are to be presented.

Send-on-Delta

The Send-on-Delta trigger computes the difference between the current sensor observation $\underline{z}_k$ and the last sensor measurement sent when an event occurred $\underline{z}_l$, denoting l as the last event time step. In this sense, this trigger focuses on the difference of measurement values throughout time in order to make the decision.

The trigger condition can be expressed as

$$k_e = \min\{k > l \mid \|z_k - z_l\|_2 > \Delta\}\,, \tag{2.9}$$

where k_e denotes an actual event time step, Δ is the threshold to be exceeded in order for a transmission to take place, and the Euclidean norm is adopted as a distance measure, although any other standard may be used. It is important to point out that for the Send-on-Delta trigger as well as for all of the following explained deterministic triggers, Δ remains invariant through time.

Predictive Sampling

Instead of considering the difference only between measurement values, a comparison can also be made in term of predictions. Predictive Sampling consists of evaluating the difference between the current measurement $\underline{z}_k$ and a predicted measurement $\hat{\underline{z}}_k$, computed whether by a parallel Kalman filter implemented at the sensor node or by a transmission taking place from the receiver back to the sensor. However, neither of them are desirable in this research given the requirements previously explained, since the desired resource saving and low computation effort would be violated. For this reason this approach was not finally implemented. Again as a distance measure, the Euclidean norm is selected.

The trigger condition would in this case look like

$$k_e = \min\{k > l \mid \|\underline{z}_k - \hat{\underline{z}}_l\|_2 > \Delta\}\,.$$

2.2.2 Stochastic Trigger

Stochastic triggers solve the Gaussianity violation problem. They preserve the state variables distribution function due to the uncertainty introduced when deciding to trigger. This approach simply removes the certainty of a transmission according only to the current measurement. In contrast to deterministic triggers, no hard decision must be made but instead the threshold is this time implemented by a random variable, which changes for each trigger on every time step. In this section, further details of the stochastic trigger are described according to [3, 13].

Let $\boldsymbol{\xi}_k$ be a uniformly distributed random variable between [0,1]. For each time step k, the condition for a trigger to transmit information towards the estimator is defined by

$$\boldsymbol{\xi}_k > \phi(\underline{z}_k)\,,$$

from where a binary variable can be derived in order to describe the trigger behaviour on each time step k, formulated as

$$\gamma_k = \begin{cases} 1, & \boldsymbol{\xi}_k > \phi(\underline{\delta}_k)\,, \\ 0, & \boldsymbol{\xi}_k \leq \phi(\underline{\delta}_k)\,. \end{cases} \tag{2.10}$$

As it is shown in Equation (2.10), the threshold is now compared against $\phi(\underline{\delta}_k)$, employed as a decision function and characterized by an unnormalized Gaussian density function designated by

$$\phi(\underline{\delta}_k) = \exp\left(-\frac{1}{2}\underline{\delta}_k^{\mathrm{T}}\mathbf{Z}^{-1}\underline{\delta}_k\right)\,. \tag{2.11}$$

Analyzing in detail Equation (2.11), two remarkable variables stand out. On the one side, the matrix $\mathbf{Z}$ introduces an additional degree of freedom in the system, designed with the aim of finding the tradeoff between the communication rate and the estimation performance. Furthermore, it plays the role of an additive measurement noise at those time steps when no new measurement arrives at the receiver in absence of an event. It is important to highlight that the greater the parameter $\mathbf{Z}$ is, the smaller the event-rate will be and hence, the lower the estimation quality will be. In other words, higher values of $\mathbf{Z}$ give out worse system performance in terms of estimate quality since the trigger condition becomes harder to fulfill.

On the other side, the parameter $\underline{\delta}_k$ is suitably defined in accordance with the desired system's behaviour. In stochastic triggering, the vectorial difference between two variables will be used. Further in my research, different configurations of the variable $\underline{\delta}_k$ are to be considered depending on the topology of the network, seeing that according to the algorithm implemented, the parameter $\underline{\delta}_k$ is forced to be defined in a concrete variable space. For the nonce, the modification to the Send-on-Delta deterministic implementation explained in Section 2.2.1 is to be adapted.

Let first define the specific case when the stochastic trigger uses a Send-on-Delta trigger. In this case, $\underline{\delta}_k$ is defined as

$$\underline{\delta}_k = \underline{z}_k - \underline{z}_l\,, \tag{2.12}$$

what leads to the uncertainty of the transmission decision by means of $\underline{z}_k$. In this sense, the probability of sending the new measurement towards the estimator can be seen as

$$\mathrm{P}\left\{\gamma_k = 1 \,\middle|\, \underline{\delta}_k\right\} = 1 - \phi(\underline{\delta}_k)\,.$$

In other words, the decision function $\phi(\underline{\delta}_k)$ can directly be interpreted as the probability of non-sending, for a given $\underline{\delta}_k$.

$$\mathrm{P}\left\{\gamma_k = 0 \,\middle|\, \underline{\delta}_k\right\} = \phi(\underline{\delta}_k)\,,$$

where $\phi(\underline{\delta}_k)$ lies between $(0,1]$. To conclude with the Send-on-Delta case, increasing values of $\underline{\delta}_k$ lead to an increasing probability of transmission. For $|\underline{\delta}_k| \to \infty$ it holds

$$\mathrm{P}\left\{\gamma_k = 1 \,\middle|\, \underline{\delta}_k\right\} \to 1\,.$$

If we now extend these results to the general form of a stochastic trigger, the parameter $\underline{\delta}_k$ would actually depend of the choice of $\underline{c}$ in the following expression

$$\underline{\delta}_k = \underline{z}_k - \underline{c}\,,$$

where no restrictions hold. In order to determine $\underline{c}$, the probability of not transmitting at a determined time step k can be computed as

$$\begin{aligned}
\mathrm{P}\left\{\gamma_k = 0 \,\middle|\, \underline{x}_k\right\} &= \int\limits_{-\infty}^{\infty} \mathrm{P}\left\{\gamma_k = 0, \underline{y}_k \,\middle|\, \underline{x}_k\right\} \mathrm{d}\,\underline{y}_k \\
&= \int\limits_{-\infty}^{\infty} \mathrm{P}\left\{\gamma_k = 0 \,\middle|\, \underline{y}_k, \underline{x}_k\right\} \cdot \mathrm{P}\left\{\underline{y}_k \,\middle|\, \underline{x}_k\right\} \mathrm{d}\,\underline{y}_k \\
&= const. \cdot \int\limits_{-\infty}^{\infty} \exp\left(-0.5(\underline{y}_k - \underline{c})^{\mathrm{T}}\,\mathbf{Z}^{-1}\,(\underline{y}_k - \underline{c})\right) \cdot \\
&\qquad \exp\left(-0.5(\underline{y}_k - \mathbf{C}_k\,\underline{x}_k)^{\mathrm{T}}\,\mathbf{R}^{-1}\,(\underline{y}_k - \mathbf{C}_k\,\underline{x}_k)\right) \mathrm{d}\,\underline{y}_k \\
&= const. \cdot \exp\left(-0.5(\underline{c} - \mathbf{C}\,\underline{x}_k)^{\mathrm{T}}(\mathbf{Z} + \mathbf{R})^{-1}(\underline{c} - \mathbf{C}\,\underline{x}_k)\right)\,.
\end{aligned}$$

A Gaussian distribution for the probability of non-transmitting is obtained by any choice of $\underline{c}$. This, in fact, constitutes the principle used later on in Section 3.2 where the state estimation is derived for the stochastic trigger scenario.

Finally, it is important to point out that the parameter $\underline{c}$ should be easily reproduced at the receiver, since its value is used in the estimation process at those time steps when there is no measurement transmitted, i.e., in absence of events.

In conclusion, given the basic fundamentals of state estimation in Section 2.1 and of event-based transmission mode in Section 2.2 the following Ssction deals with the complete scenario and the adaptions needed in order to estimate state variables in presence of a stochastic trigger. Later on in Section 4 everything explained will be extended to a multiple sensor approach, and finally in Section 5 numerical results will be shown, as well as a final conclusion about all topologies developed. Finally, Section 6 will sum up the main points of this work and will arise new issues for future research.

CHAPTER 3

State Estimation with Event-based Transmission

According to the fundamentals presented in Chapter 2, the combination of both state estimation and event-based transmission is to be developed in the following. The scheduled procedure (i.e., predict and update steps) has to be appropriately adapted taking all previous exposed facts into consideration.

3.1 Deterministic Trigger

Given the basics of deterministic triggering explained in Section 2.2.1, this section seeks to apply them to the original state estimation formulation, so that equations include event-based intrinsics. To this effect and according to the principles presented in [11], we define $\mathbb{T}_\mathrm{p}$ which contains all periodic time steps with a sampling time τ_s. It can be expressed as

$$\mathbb{T}_\mathrm{p} = \left\{ n\tau_\mathrm{s} \,\middle|\, k \in \mathbb{Z}_+ \right\} \,,$$

where $\mathbb{T}_\mathrm{p} \subset \mathbb{R}$. In this sense, time steps when events happen $\mathbb{T}_\mathrm{e}$ are also contained in $\mathbb{T}_\mathrm{p}$ and therefore it holds

$$\mathbb{T}_\mathrm{e} \subset \mathbb{T}_\mathrm{p} \,.$$

On the other side, at those time steps where the event criteria is not fulfilled and thus no transmission takes place the measurement values lie within a bounded subset, connoted as

$$\mathbb{H}(\mathrm{e}, k) \subset \mathbb{R}^m \,,$$

where m refers to the dimension of the measurement vector. We can now rewrite the trigger condition described in Equation (2.9) for the Send-on-Delta trigger in terms of $\mathbb{H}(\mathrm{e}, k)$ as

$$k_e = \min\{k > l \,|\, \underline{z}_k \notin \mathbb{H}(\mathrm{e}, k)\} \,.$$

This results in two possible situations at the estimator: the use of an actual new measurement or the use of an implicit previous measurement from a past time step, i.e.,

$$\underline{z}_k \in \begin{cases} \left\{\underline{z}_{k_e}\right\}, & k \in \mathbb{T}_\mathrm{e}\,, \\ \mathbb{H}(\mathrm{e}, k), & k \in \mathbb{T}_\mathrm{p} \setminus \mathbb{T}_\mathrm{e}\,. \end{cases} \tag{3.1}$$

Both cases are approximated by an elliptic quantity and given

$$\underline{z}_k \in \begin{cases} \mathbb{L}_{(\underline{z}_{k_e}\,,\ \epsilon\mathbf{I})}, & k \in \mathbb{T}_\mathrm{e}\,, \\ \mathbb{L}_{(\hat{\underline{z}}_{\mathrm{e},k}\,,\ H_{\mathrm{e},k})}, & k \in \mathbb{T}_\mathrm{p} \setminus \mathbb{T}_\mathrm{e}\,, \end{cases}$$

for $\epsilon \to 0$ the elliptic quantity $\mathbb{L}_{(\underline{z}_{k_e}, \epsilon\mathbf{I})}$ would exactly be the measurement value $\underline{z}_{k_e}$ of the event taking place. Moreover, the general expression of an ellipsoid with average $\underline{\mu}$ and a variance $\mathbf{\Sigma}$, can be represented by

$$\mathbb{L}_{(\underline{\mu}, \mathbf{\Sigma})} := \left\{\underline{x} \in \mathbb{R}^m \,\middle|\, (\underline{x} - \underline{\mu})^\mathrm{T}\, \mathbf{\Sigma}^{-1}\, (\underline{x} - \underline{\mu}) \leq 1\right\} \,.$$

In order to rewrite the Expression (3.1) as an equation, the noise term $\underline{e}_k$ has to be introduced in the following way,

$$\underline{z}_k + \underline{e}_k = \begin{cases} \underline{z}_{e,k}, & k \in \mathbb{T}_\mathrm{e}\,, \\ \hat{\underline{z}}_{e,k}, & k \in \mathbb{T}_\mathrm{p} \setminus \mathbb{T}_\mathrm{e}\,, \end{cases}$$

where $\hat{\underline{z}}_{e,k}$ is a predicted observation needed due to the lack of an actual measurement. Given this result, the measurement or observation model can be expressed for both stochastic and deterministic noise as

$$
\begin{aligned}
\text{Model:} \quad & \underline{\tilde{z}}_k = \mathbf{C}_k \, \underline{x}_k + \underline{\boldsymbol{v}}_{\mathrm{s},k} + \underline{v}_{\mathrm{d},k} \, , \\
\text{Realization:} \quad & \underline{\tilde{z}}_k = \begin{cases} \underline{z}_{e,k}, & k \in \mathbb{T}_{\mathrm{e}} \, , \\ \underline{\hat{z}}_{e,k}, & k \in \mathbb{T}_{\mathrm{p}} \setminus \mathbb{T}_{\mathrm{e}} \, , \end{cases}
\end{aligned}
$$

where again, as explained above, a stochastic implementation would then preserve the Gaussian distribution of the measurement's noise $\underline{\boldsymbol{v}}_{\mathrm{s},k} \sim \mathcal{N}(0, \mathbf{R})$, where by a deterministic trigger additionally the equality $\underline{v}_{\mathrm{d},k} = \underline{e}$, $\underline{v}_{\mathrm{d},k} \in \mathbb{L}_{(0, \mathbf{E}_{\mathrm{e},k})}$ can be applied, with

$$
\mathbf{E}_{\mathrm{e},k} := \begin{cases} 0, & k \in \mathbb{T}_{\mathrm{e}} \, , \\ \mathbf{H}_{\mathrm{e},k}, & k \in \mathbb{T}_{\mathrm{p}} \setminus \mathbb{T}_{\mathrm{e}} \, . \end{cases}
$$

In the concrete instance of a Send-on-Delta implementation, the quantity $\mathbb{H}(\mathrm{e}, k)$ is represented by an m-dimensional sphere with radius Δ and centered on $z_{k_{e-1}}$, holding for any time step $k_{e-1} < k < k_e$:

$$
\begin{aligned}
\underline{\hat{z}}_{e,k} &= \underline{z}_l \, , \\
\mathbf{H}_{\mathrm{e},k} &= \Delta \cdot \mathbf{I} \, .
\end{aligned}
$$

The *Linear Event-Triggered Estimator* (LETE) is considered to be an extension of the standard Kalman filter, which also considers a deterministic noise approach. In this case, each state estimate $\underline{\hat{x}}_k$ is also associated with a estimation error matrix $\mathbf{X}_k$, in addition to the covariance error matrix $\mathbf{P}_k$.

Aiming to minimize the maximum mean squared error, corresponding to trace$(\mathbf{P}_k + \mathbf{X}_k)$, the LETE equations for the prediction and update state then become on the one hand,

$$
\begin{aligned}
\underline{\hat{x}}_k^- &= \mathbf{A} \, \underline{\hat{x}}_{k-1} \, , \\
\mathbf{P}_k^- &= \mathbf{A} \, \mathbf{P}_{k-1} \, \mathbf{A}^{\mathrm{T}} + \mathbf{Q} \, , \\
\mathbf{X}_k^- &= \mathbf{A} \, \mathbf{X}_{k-1} \, \mathbf{A}^{\mathrm{T}} \, ,
\end{aligned}
\tag{3.2}
$$

and on the other hand,

$$\begin{aligned}
\mathbf{K}_k =& [\mathbf{P}_k^- \mathbf{C}^\mathrm{T} + \frac{1}{1-\omega_k} \mathbf{X}_k \, \mathbf{C}^\mathrm{T}] \\
& \cdot [\mathbf{C}\, \mathbf{P}_k^- \mathbf{C}^\mathrm{T} + \frac{1}{1-\omega_k} \mathbf{C}\, \mathbf{X}_k \, \mathbf{C}^\mathrm{T} + \mathbf{R} + \frac{1}{\omega_k} \mathbf{E}_k]^{-1} , \\
\underline{\hat{x}}_k =& \underline{\hat{x}}_k^- + \mathbf{K}_k \, (\underline{z}_k - \mathbf{C}\, \underline{\hat{x}}_k^-) , \\
\mathbf{P}_k =& (\mathbf{I} - \mathbf{K}_k \, \mathbf{C})\, \mathbf{P}_k^- , \\
\mathbf{X}_k =& \frac{1}{1-\omega_k} (\mathbf{I} - \mathbf{K}_k \, \mathbf{C})\, \mathbf{X}_k^- \, (\mathbf{I} - \mathbf{K}_k \, \mathbf{C})^\mathrm{T} + \frac{1}{\omega_k} \mathbf{K}_k \mathbf{E}_k \mathbf{K}_k^\mathrm{T} .
\end{aligned} \tag{3.3}$$

It is important to point out that the ellipsoid matrix $\mathbf{E}_k$ depends on the chosen trigger approach from the ones presented in Section 2.2. However, to fully define the update step the parameter ω_k has to be determined. Given this one dimensional optimization problem and according to [6], the Mean Squared Error is minimum for $0 < \omega_k < 1$, when

$$\begin{aligned}
& \min_{\omega_k} \mathrm{trace}(\mathbf{P}_k + \mathbf{X}_k) \\
& = \arg\min_{\omega_k} \mathrm{trace} \left((1-\omega_k) \left((1-\omega_k)\, \mathbf{P}_k + \mathbf{X}_k \right)^{-1} + \omega_k \, \mathbf{C} \, (\omega_k \, \mathbf{R} + \mathbf{E}_k)^{-1} \, \mathbf{C} \right)^{-1}
\end{aligned}$$

applies.

To sum up, steps shown in Equations (3.2) and (3.3) describe the LETE procedure, considered to be the extension of the standard Kalman filter to deterministic noise scenarios. In the following section, to complete the state estimation with event-based transmission explanation another modification to the originally periodic approach is presented.

3.2 Stochastic Trigger

If we now turn the direction of the research towards a stochastic triggering technique, several trigger intrinsics change. Up to this point of the work, the deterministic case was fully explained in the previous section. Focusing now on the fundamentals described in Section 2.2.2, an adaption to the Kalman filter steps presented in Section 2.1.1 is to be done.

In time steps when no information is transmitted, a correction must be done since the course of the state variable is no longer accurate. In [1] a method is presented in which a matrix $\mathbf{S}_l$ is employed to reduce the effect of the last measurement innovation, designed in such way that

$$\begin{aligned} \underline{z}_{k,l} &= \underline{z}_k - \mathbf{S}_l \, \underline{z}_l \,, \\ \mathbf{S}_l &= \mathbf{C}\,\mathbf{A}^l\,\mathbf{\Sigma}\,\mathbf{C}^{\mathrm{T}}\,(\mathbf{C}\,\mathbf{\Sigma}\,\mathbf{C}^{\mathrm{T}} + \mathbf{R})^{-1} \,, \end{aligned}$$

where l again connotes the last time step when an event took place and $\mathbf{\Sigma}$ designates the state vector covariance matrix in stationary state, statisfying the Lyapunov Stability equation

$$\mathbf{\Sigma} = \mathbf{A}\,\mathbf{\Sigma}\,\mathbf{A}^{\mathrm{T}} + \mathbf{Q},$$

limiting its practice to stable systems though.

Finally, the state estimation proposed in Section 2.2 is to be properly modified in order to include the fundamentals of a stochastic Send-on-Delta trigger, looking to achieve a Kalman filter implementation for such scenarios. On the one hand, the prediction step may not be adapted, since it does not make use of the new measurement, and therefore no modification is needed when the observation is not sent. Thus, the prediction step is still equivalent to the original formulation in (2.3) as

$$\begin{aligned} \underline{\hat{x}}_k^- &= \mathbf{A}\,\underline{\hat{x}}_{k-1} \,, \\ \mathbf{P}_k^- &= \mathbf{A}\,\mathbf{P}_{k-1}\,\mathbf{A}^{\mathrm{T}} + \mathbf{Q} \,. \end{aligned} \tag{3.4}$$

On the other hand, however, the update step incorporates the trigger decision variable γ_k to identify the two possible scenarios taking place described in Formula (2.10).

$$\begin{aligned} \underline{\hat{x}}_k &= \underline{\hat{x}}_k^- + \mathbf{K}_k\,(\gamma_k\,\underline{\eta}_k - \underline{\hat{\eta}}_k^-) \,, \\ \underline{\eta}_k &= \underline{z}_k - \underline{z}_l \,, \\ \underline{\hat{\eta}}_k^- &= \mathbf{C}_k\,\underline{\hat{x}}_k^- - \underline{z}_l \,, \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\,\mathbf{C})\,\mathbf{P}_k^- \,, \\ \mathbf{K}_k &= \mathbf{P}_k^-\,\mathbf{C}^{\mathrm{T}}\,[\mathbf{C}\,\mathbf{P}_k^-\,\mathbf{C}^{\mathrm{T}} + \mathbf{R} + (1-\gamma_k)\,\mathbf{Z}]^{-1} \,. \end{aligned} \tag{3.5}$$

In conclusion, Formulae (3.4) and (3.5) constitute the state estimation event-based algorithm by means of the Kalman filter.

Chapter 2 and Chapter 3 deal with the state-of-art single sensor scenario, where the previous fundamentals explained are applied to a single sensor network to achieve an optimal state estimation employing a Kalman filter. In the following chapter, the extension to a multiple sensor case is presented where different approaches are taken into consideration, i.e., star, ring, and chain topologies are developed and then tested in a final evaluation in Chapter 5.

CHAPTER 4

Multisensor Event-based State Estimation

Sensor elements in real networks rarely work on their own since readings from a unique noisy device brings little to a desirable robust system. To overcome the drawbacks of noise in observations, the combination of different sensors' data is preferred. Besides, being able to foresee future system states by means of present values allows the end user to achieve a better service quality, by dynamically adjusting the network to the current demand for example.

The way in which the different devices of a multisensor network are organized influences significantly the global system performance. When turning from a single to a multisensor scenario, issues about how to communicate the sensors to the central unit arise. Treating all sensors independently leads to an undesired waste of resources since combining the measurements from different sensors in a yet unknown way could easily lead to a quality improvement or a network resource saving.

In this section, three main topologies are to be studied and evaluated in terms of the *Minimum Squared Error*(MSE) for a given event-rate interval between [0,1]. Each of the next proposed approaches are based only on different sensors' distributions, keeping the other network variables constant. First, an elementary approach is presented in Section 4.1, named the star topology, where sensors only transmit to the central unit and no communication between them is considered. This scenario will be used as the basis for all future comparisons due to its simplicity to implement it. Later on in this section, other two prime network structures are to be developed, which aim to improve the mentioned star topology performance by involving communication between the sensor nodes and by trying to exploit common information between them and to overcome degradation introduced by the existence of noise in the system. On the one hand, the chain topology explained in Section 4.3 simply concatenates sensors one after the other one with the purpose of combining all measured information before processing it in the control unit. In this sense, each node would not only receive a measurement $\underline{z}_k^i$ as an input but also the previous

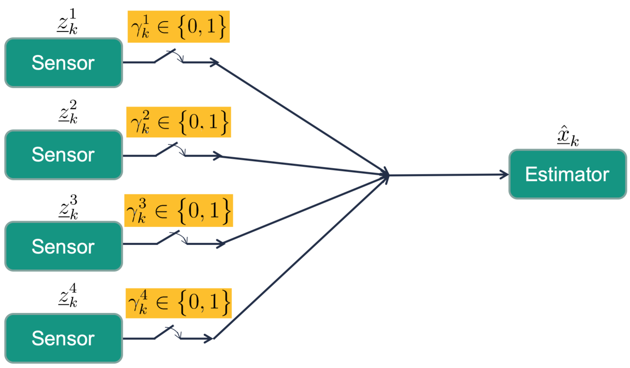

Figure 4.1: Star topology network.

sensor's information in case it was sent $\underline{z}_k^{i-1}$, where i is the number of the sensor in the network. Furthermore, even when the preceding sensor does not transmit, information can still be used since the absence of events allows the reuse of the last received transmission. Hence, in this case the ith-sensor would receive $\underline{z}_l^{i-1}$. How to possibly combine both terms in a smart manner will be addressed in two different ways further on in this work.

On the other hand and continuing on the same path, if not only were sensors efficiently communicated between each other but also counted with the estimation beforehand, computation taking place across the network would then remain clean and straightforward. This in fact constitutes the main idea of the ring topology explained in Section 4.2, where by closing the mentioned earlier chain in a circle sense, the algorithm could simply start from the predictions and employ the sensor nodes information later on.

4.1 Star Topology

As shown in Figure 4.1, the star topology represents the simplest method to solve the arrangement of several sensors in a network. First of all, sensors in this network structure are only dependent on their own measurement $\underline{z}_k^i$ at each time step k, which frees us from changing the trigger definition described in Section 2.2. In other words, the trigger condition in this basic scenario still relies only on their own new measurement and their last transmission, which must be saved after each new event. This fact constitutes the simplicity of the additional sensors' handling in this topology since the triggering condition can still be defined in the measurement space, according to Equations (2.10) and (2.12).

Therefore, the transmission mode in the present network layout (i.e., periodic or event-based) influences only the information received at the input of the estimator, which will be N units of data, whether brand new observations in those time steps k where events happened or past measurements $\underline{z}_l$ repeated throughout time. This leads to a minor modification to the original Kalman filter formulation explained in Equation (2.4), which in this scenario each sensor information is computed sequentially.

In essence, the estimator takes one by one all observations $\underline{z}_k^i$ measured at the same time, computing an estimate after each one $\hat{\underline{x}}_k^i$ and using it as a new prediction $\hat{\underline{x}}_k^{(i+1)-}$ for the next update $\hat{\underline{x}}_k^{i+1}$. Likewise, the covariance matrix $\mathbf{P}_k^i$ is correspondingly updated. To sum up, $\forall i$ it applies

$$
\begin{aligned}
&ith - Sensor\ computes: \\
&\qquad \mathbf{K}_k^i = \mathbf{P}_k^{i-}\,\mathbf{C}^{\mathrm{T}}\,[\mathbf{C}\,\mathbf{P}_k^{i-}\mathbf{C}^{\mathrm{T}} + \mathbf{R} + \mathbf{Z}(1-\gamma_k^i)]^{-1} \\
&\qquad \hat{\underline{x}}_k^i = (\mathbf{I} - \mathbf{K}_k^i\mathbf{C})\hat{\underline{x}}_k^{i-} + \mathbf{K}_k^i\underline{z}_l^i \\
&\qquad \mathbf{P}_k^i = (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{P}_k^{i-} \\
&ith - Sensor\ updates: \\
&\qquad \hat{\underline{x}}_k^{(i+1)-} = \hat{\underline{x}}_k^i \\
&\qquad \mathbf{P}_k^{(i+1)-} = \mathbf{P}_k^i,
\end{aligned}
\tag{4.1}
$$

where $\underline{z}_l^i$ already incorporates the last trigger decision according to Formula (2.10). Finally, after all N sensor nodes have computed an estimation, a precise favourable final one is obtained which the central Kalman filter node will later on embrace in order to predict the next state value, with regard to algorithm described in Equation (2.3).

If we now focus on the information form described in Section 2.1.2, the same scenario can be computed as

$$
\begin{aligned}
&ith - Sensor\ computes: \\
&\qquad \hat{\underline{y}}_k^i = \hat{\underline{y}}_k^{i-} + \mathbf{C}^{\mathrm{T}}[\mathbf{R} + \mathbf{Z}(1-\gamma_k^i)]^{-1}\underline{z}_l^i \\
&\qquad \mathbf{Y}_k^i = \mathbf{Y}_k^{i-} + \mathbf{C}^{\mathrm{T}}[\mathbf{R} + \mathbf{Z}(1-\gamma_k^i)]^{-1}\mathbf{C} \\
&ith - Sensor\ updates: \\
&\qquad \hat{\underline{y}}_k^{(i+1)-} = \hat{\underline{y}}_k^i \\
&\qquad \mathbf{Y}_k^{(i+1)-} = \mathbf{Y}_k^i
\end{aligned}
\tag{4.2}
$$

applicable $\forall i$, where again $\underline{z}_l^i$ refers to the last transmitted measurement. The Kalman filter node later utilizes the Nth-Sensor estimation to compute a new prediction according to Equation (2.5).

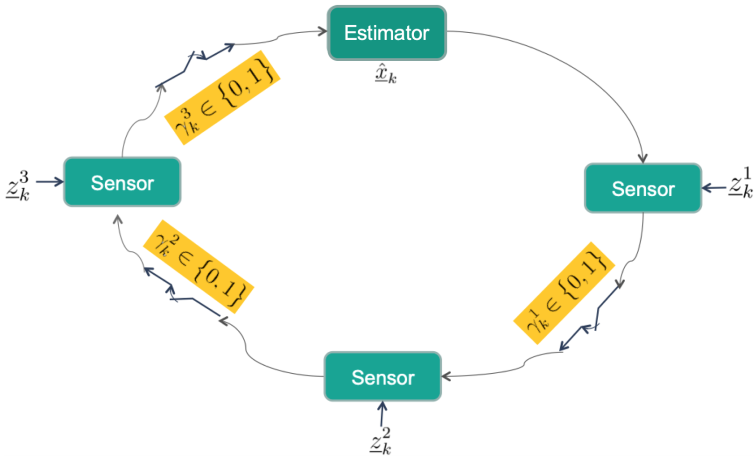

Figure 4.2: Ring topology network.

Up to this point of the work, computation effort differs substantially inside the sensor nodes between both formulations of the star topology formulated in Equation (4.1) and Equation (4.2), endowing the information filter of attractive advantages. Ultimately the pair of parameters $(\underline{y}_k^i, \mathbf{Y}_k^i)$ would be turned back to state space by means of the inverse covariance matrix and finally the original pair $(\underline{x}_k, \mathbf{P}_k)$ would be obtained. Numeric and performance results of this topology are found later on in Section 5.

4.2 Ring Topology

Research at this point of the work turned the filter approach point of view in order to achieve an event-based multisensor topology which could beat the star results. The introduction of transmissions between sensor nodes opens a whole new wide range of possibilities in terms of network structures. Nevertheless, multiple difficulties arise and their handling becomes a great issue to consider as it will be shown later on in this section. Focusing first on the ring topology development, Figure 4.2 shows its network structure where the two main parameters sent across the network are to be changed to $(\underline{i}_k^i, \mathbf{I}_k^i)$ although the information form is still worthwhile. This fact required the estimator node to be instead placed first, so that at each time step k the predicted values $(\hat{\underline{y}}_k^-, \mathbf{Y}_k^-)$ were already computed and used by each sensor, who will later on add its information vector and matrix to the existing estimation.

When first attempted this topology in a periodic transmission formulation, results proclaimed that the ring and the star topologies were mathematically equivalent in absence of trigger elements. This equivalence was not surprising since both scenarios integrate each sensor estimation onto the next. Hence, mathematical elaboration of this method is already described in Section (4.1) taking γ_k as 1 what would result in the following algorithm.

$$\begin{aligned}
&ith - Sensor\ computes: \\
&\qquad \underline{\hat{y}}_k^i = \underline{\hat{y}}_k^- + \underline{i}_k^i \\
&\qquad \mathbf{Y}_k^i = \mathbf{Y}_k^- + \mathbf{I}_k^i \\
&ith - Sensor\ updates: \\
&\qquad \underline{\hat{y}}_k^{(i+1)-} = \underline{\hat{y}}_k^i \\
&\qquad \mathbf{Y}_k^{(i+1)-} = \mathbf{Y}_k^i
\end{aligned} \tag{4.3}$$

Given this ring topology periodic algorithm, it is important to highlight that the only difference with respect to that of the chain topology periodic case presented later on in Section 4.3 is the order in which several terms of a sum are grouped. In other words, by simply moving the brackets in the estimated information vector $\underline{\hat{y}}_k$ computation the ring and chain measurement update steps can be distinguished. This can be shown in the following formulae

$$\begin{aligned}
\underline{\hat{y}}_k &= (((\underline{\hat{y}}_k^- + \underline{i}_k^i) + \underline{i}_k^{i+1}) + ...) + \underline{i}_k^N\,, \\
\underline{\hat{y}}_k &= \underline{\hat{y}}_k^- + ((\underline{i}_k^i + \underline{i}_k^{i+1}) + ... + \underline{i}_k^N)\,.
\end{aligned} \tag{4.4}$$

Finally facing the event-based implementation in the ring multisensor scenario and considering now the concatenation of triggers after each sensor, an alteration to Formulae (4.3) is now incorporated in those time steps when no event is captured by a trigger. The complete execution taking place in this cases can be formulated as

$$\begin{aligned}
&if\ no\ event: \\
&\qquad \mathbf{Y}_k^i = ((\mathbf{Y}_l^i)^{-1} + \mathbf{Z})^{-1} \\
&\qquad \underline{\hat{y}}_k^i = \mathbf{Y}_k^i(\mathbf{Y}_l^i)^{-1}\underline{\hat{y}}_l^i,
\end{aligned} \tag{4.5}$$

where $\underline{\hat{y}}_l^i$ and $\mathbf{Y}_l^i$ designate the information vector and information matrix which were last sent by the ith-sensor on the last event time step.

Thus, given the developed algorithm it is clear that this new topology not only brings worthy advantages when considering numerous sensor nodes, but also employs an inferior computation effort when event-based transmission is studied. To express the

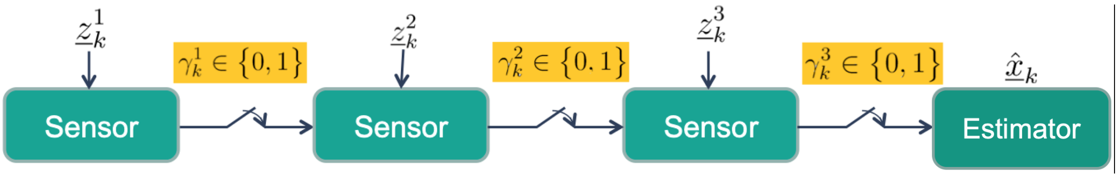

Figure 4.3: Chain topology network.

trigger criteria, a conversion back to the state space needs to be done by modifying Equation (2.12) to

$$\begin{aligned}\underline{\delta}_k &= (\mathbf{Y}_k)^{-}\hat{\underline{y}}_k - (\mathbf{Y}_l^i)^{-1}\hat{\underline{y}}_l \\ &= \hat{\underline{x}}_k - \hat{\underline{x}}_l \,.\end{aligned}$$

This attractive attempt to improve the simple star topology in a way where all different observations were first combined before processing them, resulted in an inferior efficiency, since error variables are growing across the network due to the increasing information's uncertainty on its way towards the estimator. In conclusion, as it will be numerically explained in the following section, handling observations only once before the implementation of the Kalman filter (i.e., the star topology algorithm) is still worthy when comparing it with realizing several computations with observations before estimating (i.e., the ring topology implementation).

4.3 Chain Topology

Due to the large error accumulated across the ring topology explained in Section 4.2 another sensors' rearrangement was developed. Figure 4.3 shows the structure of the chain topology. Given this scenario, two different implementations of the chain were investigated in this research and in both periodic and event-base transmission mode.

The first topology development is named the information form chain, due to the exclusive transmission of the information vector and information matrix throughout the chain. Thus, it can be completely described in reference to both and its basics are explained in Section 4.3.1. Since there were several issues on its development, the second approach to solve the chain topology is later on explained in Section 4.3.2 , denoted as the measurement fusion chain topology.

4.3.1 Information Form Chain Topology

In absence of triggers, i.e., when periodic transmission is implemented, the system implementation is trivial when employing the information form. Focusing first on the sensor nodes' behaviour, each observation $\underline{z}_k^i$ will need first to be converted into an information vector $\underline{i}_k^i$ according to the transformation explained in (2.6). Then, each information vector and matrix $(\underline{i}_k^i, \mathbf{I}_k^i)$ will contribute to a global summation transmitted across the network connoted hereinafter $(\underline{j}_k^i, \mathbf{J}_k^i)$.

For instance, if only two sensors were considered, we could merely express

$$\begin{aligned}\underline{j}_k^i &= \mathbf{C}_k^{\mathrm{T}} \cdot \mathbf{R}_k^{-1} \cdot \underline{z}_k^i + \mathbf{C}_k^{\mathrm{T}} \cdot \mathbf{R}_k^{-1} \cdot \underline{z}_k^{i-1} \,, \\ \mathbf{J}_k^i &= \mathbf{C}_k^{\mathrm{T}} \cdot \mathbf{R}_k^{-1} \cdot \mathbf{C}_k + \mathbf{C}_k^{\mathrm{T}} \cdot \mathbf{R}_k^{-1} \cdot \mathbf{C}_k \,,\end{aligned}$$

to describe how the ith-sensor accumulates its own information with the one immediately received. If we now generalize this results to N sensors, it holds

$$\begin{aligned}\underline{j}_k^N &= \mathbf{C}_k^{\mathrm{T}} \cdot \mathbf{R}_k^{-1} \cdot \sum_{i=1}^{N} \underline{z}_k^i \,, \\ \mathbf{J}_k^N &= \sum_{i=1}^{N} (\mathbf{C}_k^{\mathrm{T}})^i \cdot (\mathbf{R}_k^{-1})^i \cdot \mathbf{C}_k^i \,,\end{aligned}$$

to the last ongoing parameters. Finally, the estimator located at the end of the chain would compute an estimate as

$$\begin{aligned}\underline{\hat{y}}_k &= \underline{\hat{y}}_k^- + \underline{j}_k^N \,, \\ \mathbf{Y}_k &= \mathbf{Y}_k^- + \mathbf{J}_k^N .\end{aligned}$$

Like in every information form implementation, a last space transformation needs to be executed to finally get the variables $(\underline{x}_k, \mathbf{P}_k)$ in their original state space, so reversing Equation (2.5) we get

$$\begin{aligned}\underline{\hat{x}}_k &= \mathbf{P}_k^{-1} \underline{\hat{y}}_k , \\ \mathbf{P}_k &= \mathbf{Y}_k^{-1} .\end{aligned}$$

The algorithm explained in a periodic transmission scenario was successfully implemented. However, if we now turn the direction of the chain topology research towards an event-based transmission system, severe mathematical issues come into view. The unfamiliar fact that now sensors deal with information terms and not raw measurement values makes the trigger condition definition unfeasible.

Studies during this research development showed that when expecting to convert information vectors back to the measurement space in intermediate nodes, with the purpose of implementing once again the sensor's trigger in terms of measurements singular matrices appeared. This problem when trying to compute the algorithm presented in (4.5) in absence of events made it impossible to simulate an information form chain topology scenario in a desirable event-based transmission mode, and therefore only a periodic transmission result was completely developed.

As to continue deep down the topology impacts in a multisensor network, a shift between the parameters sent from node to node throughout the network needs to be done. Up to this point, we can conclude that the straightforward sharing of information vectors $\underline{i}_k^i$ in an intermediate accumulative term $\underline{j}_k^i$ impedes the system to incorporate triggers between sensors.

To overcome this obstacle, the attempt to communicate directly information estimations $(\underline{y}_k^i,\ \mathbf{Y}_k^i)$ after a local sum is already computed inside each sensor node was presented in Section 4.2, proving that the brackets' order change in the Algorithm (4.4) made it possible to achieve an event-based approach. This simple change of standpoint between transmitting $\underline{i}_k^i$ or $\hat{\underline{y}}_k^i$ allows the fully development of the topology, which in this first case was not successful. In conclusion, computing an estimation step at each sensor leads to a successful practice which in the information space consists of a simple local sum. Additionally, another implementation of the chain topology took place in this research, based on a measurement fusion beforehand. The next section explains its intrinsics and deals with its new formulation, as well as consistent results are later on presented in Chapter 5.

4.3.2 Measurement Fusion Chain Topology

According to Q. Gan and C. J. Harris in [2], if all observation models $\mathbf{C}_k^i$ are the same, where then $\forall i$ it applies

$$\mathbf{C}_k^i \ = \mathbf{C}_k^{i+1} \ = \mathbf{C}_k \, ,$$

a simple solution to a periodic transmission scenario can be derived through the use of data fusion. Looking for an event-based extension and constraining the network model to a single sensor model, we could then implement a measurement fusion after each node transmission. Given for instance two sensors, if the first trigger decides to send information, the following computation would take place on the second sensor

$$\begin{aligned}\tilde{\mathbf{R}}_k^2 &= ((\mathbf{R}_k^1)^{-1} + (\mathbf{R}_k^2)^{-1})^{-1}\,, \\ \tilde{\underline{z}}_k^2 &= \tilde{\mathbf{R}}_k^2((\mathbf{R}_k^1)^{-1}\underline{z}_k^1 + (\mathbf{R}_k^2)^{-1}\underline{z}_k^2)\,,\end{aligned} \tag{4.6}$$

where $(\tilde{\mathbf{R}}_k,\, \tilde{\underline{z}}_k)$ are the fused error covariance matrix and the fused measurement respectively. To proof that in fact it is equivalent to the original multisensor Kalman filtering process, the estimated state variable and error covariance matrix should look like

$$\begin{aligned}\hat{\underline{x}}_k &= \mathbf{P}_k((\mathbf{P}_k^-)^{-1}\underline{x}_k^- + \mathbf{C}_k^{\mathrm{T}}(\mathbf{R}_k^1)^{-1}\underline{z}_k^1 + \mathbf{C}_k^{\mathrm{T}}(\mathbf{R}_k^2)^{-1}\underline{z}_k^2)\,, \\ \mathbf{P}_k^{-1} &= (\mathbf{P}_k^-)^{-1} + \mathbf{C}_k^{\mathrm{T}}(\mathbf{R}_k^1)^{-1}\mathbf{C}_k + \mathbf{C}_k^{\mathrm{T}}(\mathbf{R}_k^2)^{-1}\mathbf{C}_k\,.\end{aligned} \tag{4.7}$$

If we now replace the fused terms defined in Equation (4.6), we get the following expressions:

$$\begin{aligned}\hat{\underline{x}}_k &= \mathbf{P}_k((\mathbf{P}_k^-)^{-1}\underline{x}_k^- + \mathbf{C}_k^{\mathrm{T}}(\tilde{\mathbf{R}}_k^1)^{-1}\tilde{\underline{z}}_k^1 \\ &= \mathbf{C}_k^{\mathrm{T}}(\tilde{\mathbf{R}}_k)^{-1}\tilde{\mathbf{R}}_k((\mathbf{R}_k^1)^{-1}\underline{z}_k^1 + (\mathbf{R}_k^2)^{-1}\underline{z}_k^2)\,, \\ \mathbf{P}_k^{-1} &= (\mathbf{P}_k^-)^{-1} + \mathbf{C}_k^{\mathrm{T}}(\tilde{\mathbf{R}}_k)^{-1}\mathbf{C}_k \\ &= (\mathbf{P}_k^-)^{-1} + \mathbf{C}_k^{\mathrm{T}}((\mathbf{R}_k^1)^{-1} + (\mathbf{R}_k^2)^{-1})\mathbf{C}_k\,,\end{aligned}$$

which in fact are equivalent to Equation (4.7).

Following this example, if the second sensor decides to trigger, the fusion taking place on the third sensor would look like

$$\begin{aligned}\tilde{\mathbf{R}}_k^3 &= ((\tilde{\mathbf{R}}_k^2)^{-1} + (\mathbf{R}_k^3)^{-1})^{-1}\,, \\ \tilde{\underline{z}}_k^3 &= \tilde{\mathbf{R}}_k^3((\tilde{\mathbf{R}}_k^2)^{-1}\tilde{\underline{z}}_k^2 + (\mathbf{R}_k^3)^{-1}\underline{z}_k^3),\end{aligned}$$

involving these time the pair $(\tilde{\mathbf{R}}_k^2,\, \tilde{\underline{z}}_k^2)$, which already incorporates the information of the first sensor according to (4.6). This fact helps to accumulate information all over the network, without the need of complex computation and hence, the measurement fusion chain appears as the best solution considered in this research.

Furthermore, extending this data fusion approach to a triggering scenario and applying the principles of [2] to an event-based approach a great advantage is obtained: in absence of events, when again a modification to the previous implementation has to be made, the original formulation explained in the algorithm (4.2) can simply be reused. The designed matrix $\mathbf{Z}$ can be added to $\tilde{\mathbf{R}}_k$ at these concrete time steps. Thus, if we adapt the star implementation developed in Section 4.1 to Q. Gan and C. J. Harris data fusion approach we obtain:

$$
\begin{aligned}
&ith - Sensor\ computes: \\
&\qquad \underline{\hat{y}}_k^i = \underline{\hat{y}}_k^{i-} + \mathbf{C}^{\mathrm{T}}[\tilde{\mathbf{R}} + \mathbf{Z}(1-\gamma_k^i)]^{-1}\underline{z}_l^i \\
&\qquad \mathbf{Y}_k^i = \mathbf{Y}_k^{i-} + \mathbf{C}^{\mathrm{T}}[\tilde{\mathbf{R}} + \mathbf{Z}(1-\gamma_k^i)]^{-1}\mathbf{C} \\
&ith - Sensor\ updates: \\
&\qquad \underline{\hat{y}}_k^{(i+1)-} = \underline{\hat{y}}_k^i \\
&\qquad \mathbf{Y}_k^{(i+1)-} = \mathbf{Y}_k^i
\end{aligned}
$$

This solution was named the measurement fusion chain topology and its performance resulted as good as that of the star, as it is shown in Chapter 5. This last development was considered the largest advance in this multisensor network research and was successfully implemented in the simulation environment. Nevertheless, its research is still open to new extensions or further issues that may arise in the future.

4.3.3 Conclusion

To sum up, three different methods of arranging sensor nodes in a wireless network have been studied and their algorithms have been developed in this section. A simulation environment was set up in order to quantify their gains and to represent their estimation's quality in a visual way. Numerical results were obtained as well as a fair comparison took place between them. All variable configurations, as well as the methodology employed, are fully explained in the following section.

On the one hand, the star topology was implemented in both state and information space. Both formulations gave out the same state estimation and their difference relied on their computation effort. On the other hand, the ring topology was only developed by means of the information form of the Kalman filter but its estimation quality resulted unacceptable. For this reason, this solution was rejected later on in this research. Finally, the chain topology was approached from two different points of view: an information form approach which resulted unfeasible and could not be tested, and a measurement fusion approach, which has been chosen to be the biggest advance in this work due to its great performance and simplicity in its computations.

CHAPTER 5

Evaluation

In Section 4, three main different multisensor network topologies have been presented and their event-based state estimation implementations have been developed. Given all these outcomes, a comparison between them is to be done in order to achieve results in terms of estimation quality. In addition, the state-of-art major single sensor scenarios, both periodic and event-based practices, are also to be a basis to be compared with and this research pursues to evaluate the improvements of multiple sensors in networks.

A simulation environment has been established regarding this purpose, where all possible network configurations were elegible and later on plotted considering their efficiency. The simulation takes place in discrete-time steps, at which an event can be triggered and a transmission therefore sent throughout the network. At the end of it, the state estimation takes places periodically at each time step. Sensors' measurements are assumed to be affected by *Additive White Gaussian Noise* (AWGN) and channels are supposed ideal without taking latency or packet losses into consideration.

5.1 Goals

According to event-based systems, different design goals can be pursued depending on which restrictions are decisive in the considered network. Possible scenarios among numerous options are in this section mentioned.

On the one hand, an energy limitation at the sensor nodes implies that no local state estimation should take place at them in order to keep their computing effort as low as possible. In this case, a simple trigger mechanism is suitable.

On the other hand, a limit on the transmission's bandwidth forces the information content per sending to be as high as possible, whereas the number of transmissions will correspondingly be kept minimal. Hence, in this scenario an elaborated trigger is desirable so that a local measurement estimation can be computed on each sensor node before deciding whether to transmit or not.

Additionally and especially in distributed multisensor systems, transmission rate is preferred to be kept also minimal in order to reduce the number of information fusions taking place. In this sense, processing is worthy to be distributed so that it minimizes the number of fusion computations in the central unit.

Regardless of the particular target, reducing the communication rate is in general terms an everlasting ambition which enlarges the convenience of event-based mechanisms and leads to an increasing interest in new advancements following this practice.

5.2 Network Model

Considering the main targets illustrated in Section 5.1, the network constructed in the simulation is characterized by the following features.

First of all, communication between sensors and communication sensor-to-estimator is considered to be wireless. Nevertheless, packet losses are left outside the scope of this work, assuming channels to be ideal. In addition to this, the bandwidth available for transmission is also considered ideally not limited, what does not restrict the number of transmissions per unit of time. Finally, the energy power at the sensor nodes, on the other hand, is advised tight, pursuing therefore the first of the goals mentioned in Section 5.1.

5.3 Results

The three topologies studied in this work and presented in Section 4 were compared using a Monte Carlo simulation and their analytical results can be found in Figure 5.1. There the estimation quality in terms of MSE was studied for a range of transmission rates. In order to set these, different values of $\mathbf{Z}$ were used so that all topologies could be fairly compared. The scenario configured was composed of 15 sensors with equal measurement models and a stochastic Send-on-Delta trigger annexed to each of them. The figure shows how a high number of sensors makes the star topology nearly unaffected by low transmission rates and therefore its curve is nearly constant. This holds due to the independency between the trigger decisions in the star topology.

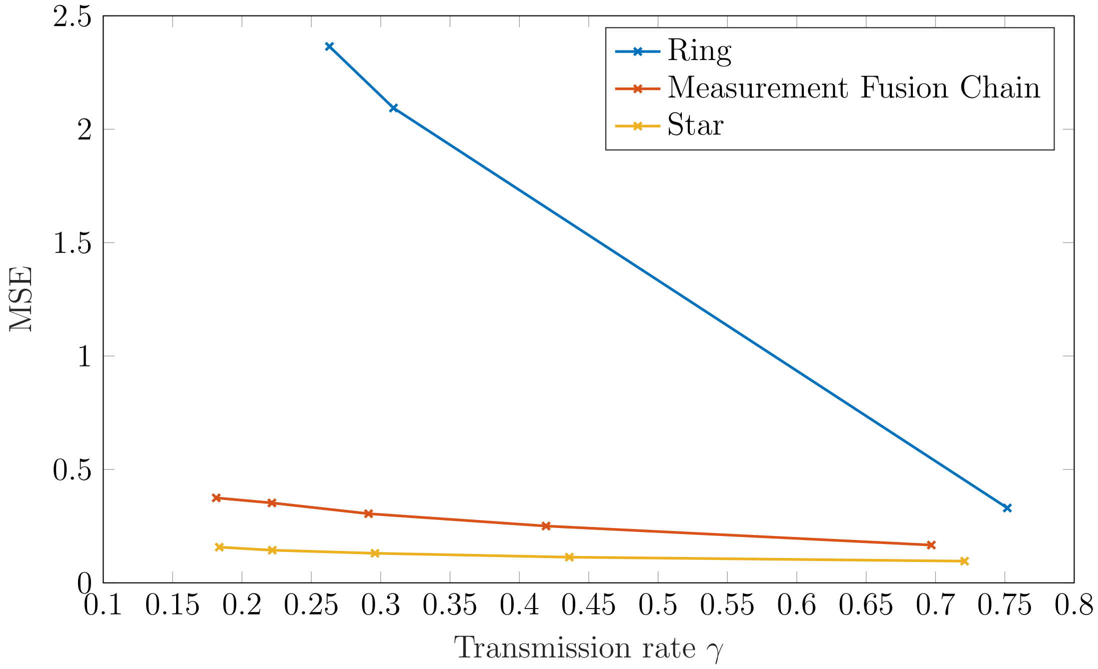

Figure 5.1: Trade off between MSE and transmission rate for each topology.

On the one hand, the measurement fusion chain has a similar performance in comparison to the star. Moreover, when reducing the number of sensors in the network the star topology performance is strongly affected while the chain topology results remain nearly unchanged. This fact turns the yellow and red curves even closer to each other. On the other hand, the ring topology seems unacceptable and has therefore been rejected as a solution to an event-based multisensor topology.

Different reasons could cause such a degradation of the ring topology estimation: first, the concatenation of several sensors decreases their event-rate obtaining a lower transmission rate in the final estimator; while on a star topology decisions are made independently what in fact keeps the transmission rate constant throughout all sensors. When studying each sensor's behaviour separately, the purple curves in Figure 5.2 confirmed the decrease of trigger decisions throughout the ring. The system was set up twice with 20 sensor nodes and the initial transmission rates were configured to 0.39 and 0.325. The figure shows how the event-rates decrease to 0.245 and 0.19 respectively.

In order to overcome this problem and seeking to achieve a better estimation quality, another implementation of the ring topology was developed. When looking for further sensors to trigger more often, the idea was to change dynamically the parameter $\mathbf{Z}$ explained in Section 2.2.2, so that the trigger criteria was easier fulfilled in future sensors. Several parameters were designed and three different implementations are shown in green in Figure 5.2.

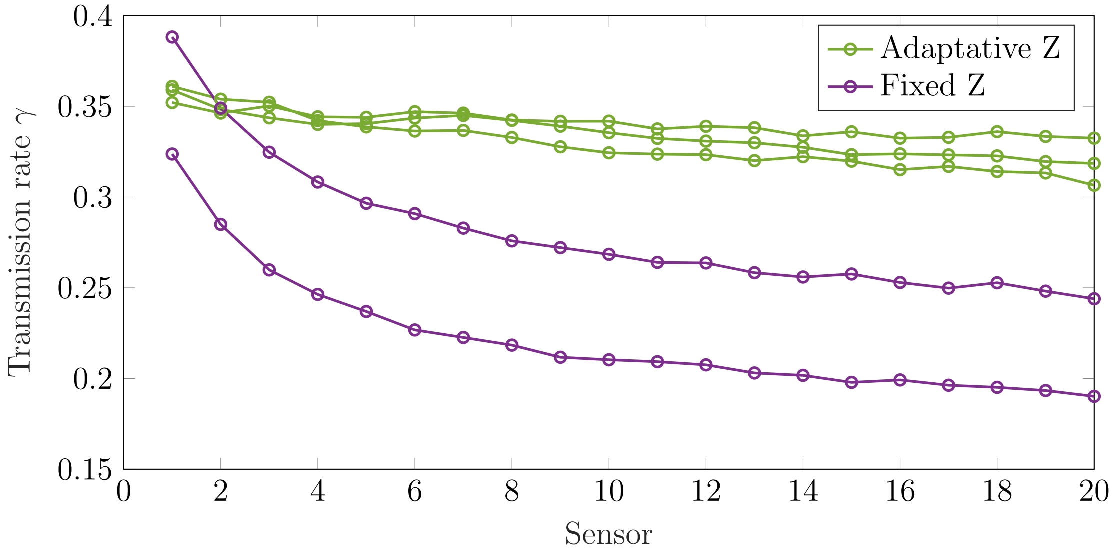

Figure 5.2: Two different implementations of the ring topology.

Even though a nearly constant rate is achieved, when the simulation was again run, results did not improve. Therefore this was rejected as the reason that causes such degradation to the ring topology performance and a closer look to its formula took place. Focusing now on those time steps when no event takes place, which are the truly issue of the ring topology, the following information matrix update takes place

$$\mathbf{Y}_k = ((\mathbf{Y}_l)^{-1} + \mathbf{Z})^{-1},$$

where $\mathbf{Y}_l$, i.e., the last sent information matrix, will always be reused unless a transmission takes place. However, on the same time steps the star topology would instead compute

$$\mathbf{Y}_k = \mathbf{Y}_k^- + \mathbf{C}^\mathrm{T}(\mathbf{R} + \mathbf{Z})^{-1}\mathbf{C}\,,$$

where $\mathbf{Y}_k^-$ is still a new prediction for its value. This means that the star topology still predicts the new information matrix even when no measurement arrives while the ring topology computes it but never uses it since it will always employ $\mathbf{Y}_l$. This has been proven to be the reason to reject the ring topology as a solution for this work.

Turning now to the measurement fusion approach presented in Section 4.3, the same study as shown in Figure 5.1 is plotted in Figure 5.3 but zoomed in for visual purposes. It can bee seen that for low transmission rates the MSE values are not higher than 0.45, which proves that this topology performs nearly as good as the star.

In Figure 5.4, the *Normalized Estimation Error Squared* (NEES), MSE and the mean trace of the error covariance matrix (in the figure named "$\mathbf{P}_E$ mean") are

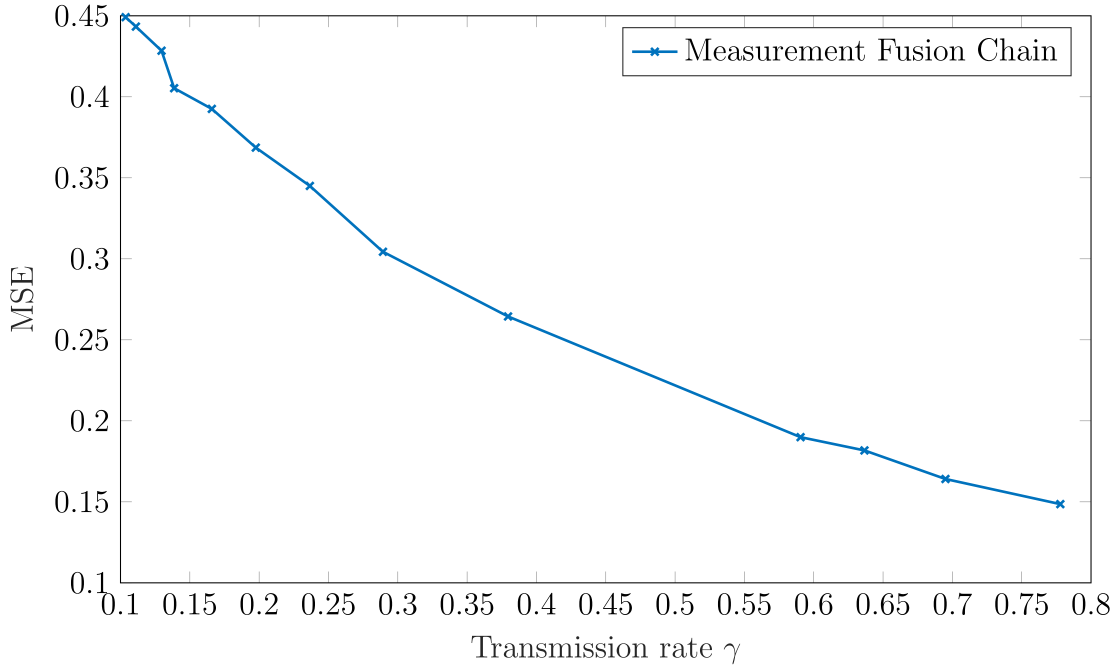

Figure 5.3: MSE analysis of the measurement fusion chain topology.

shown. On the one hand, the NEES in the ith-run is defined according to X. Rong Li in [8] as

$$\text{NEES} = (\underline{x}_k^i - \hat{\underline{x}}_k^i)^{\mathrm{T}} (\mathbf{P}_k^i)^{-1} (\underline{x}_k^i - \hat{\underline{x}}_k^i) \,,$$

which serves as a credibility measure for any estimator. If its mean value is the dimension of the state variable, the filter used is proven to be consistent. In the system implemented the dimension of the state was set to two and Figure 5.4 shows that the mean of the NEES lies around that value.

On the other hand, the MSE and the mean trace of the error covariance matrix are alike and low, both desirable and acceptance tests.

5.4 Conclusion

In conclusion, different sensors arrangements have been developed, implemented, and tested.The star topology has been used as a base for all comparisons due to its simplicity and high efficiency. The ring topology, implemented by means of a

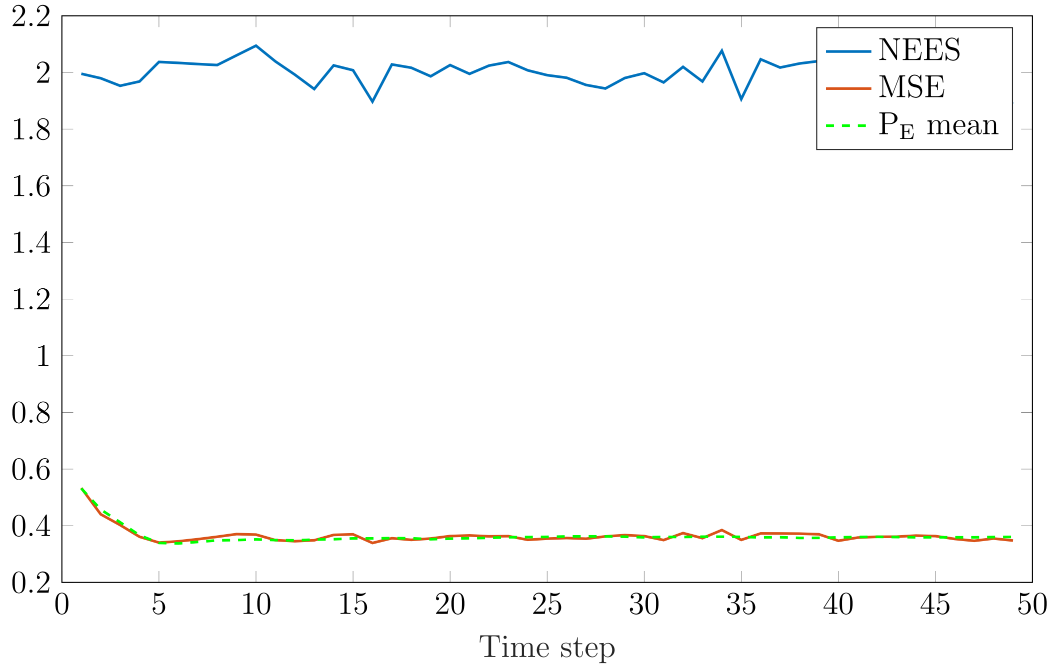

Figure 5.4: Performance of the Measurement Fusion Chain topology.

lower computational effort algorithm, has been finally rejected as an acceptable solution even though it could correctly be used in a wireless network scenario. Finally, the chain topology has proven, on the one hand, that the information space is not suitable for a trigger definition, and therefore an initial chain topology could not be fully developed. However, when fusing data beforehand mathematical equations were again feasible and hence, this last topology has been selected as the best development of this work.

CHAPTER 6

Conclusions

The state-of-art in single sensor networks with event-based transmission has been deeply discussed in this research with the purpose of extending the scenario to a multisensor network. To this effect, different sensor nodes' arrangements have been implemented with the aim of achieving the highest estimation quality with the lowest transmission rate possible. First in Section 4.1, the star topology was explained where each sensor only communicates with the central estimator. In this case, sequential estimations take place using one measurement at each time and getting to a final improved estimation thanks to the multiple observations available. A Kalman filter was employed at the estimator node in two different formulations: in the original proposed in [4] and in the information form. It was proven that both implementations give out the exact same estimation for the same input although the computational effort is lower for the second formulation, what constitutes the main advantage of the information form. Then in Section 4.2 ,the ring topology was described where communication between sensor nodes was introduced, before a combined signal reached the estimator where again a Kalman filter was placed. The ring topology implementation sends the information vector and the information matrix over the network and computes a new estimation on each sensor node. In this sense, it is equivalent to the star topology but with the main difference that in this case, the predicted values computed at the estimator can be used beforehand at each time step. This fact makes it easier for the sensors to compute the developed algorithm and allows the created network structure to work. Finally in Section 4.3, a third topology named the chain topology was described. Changing from the circle structure presented in the ring topology to an open chain, where the estimator is no longer connected to the first sensor in the network, results were strongly improved. Two different implementations were studied in this work: the information form approach and the measurement fusion approach. In the first scenario, instead of transmitting the information vector and information matrix across the network, these were computed only at the estimator and only the new sensors' information terms were sent. However, computations taking place on the sensors were unfeasible and this implementation could not be tested. In the second chain topology development, the measurement fusion published in [2] was employed. Extending it to an event-based transmission network, each sensor computed a fused measurement and a fused error covariance matrix and the final pair was later used by the Kalman filter implemented in the estimator. This last topology was finally successful and therefore, the previous implementation of the chain topology was rejected.

All three systems were implemented in a simulation environment to compare their results. In order to do so in a fair manner, by adjusting the parameter $\mathbf{Z}$ explained in Section 2.2.2 and involved in Formula (2.11) different event-rates were obtained. Each of them gave out a different estimation quality measure. The graphs presented in Chapter 5 showed that the star topology performed best of all three and ring topology the worst.

When studying possible reasons for this last fact, this research concluded that the predicted information matrix was never used by the ring topology while the other two topologies did compute it in the estimation step. This was assumed to cause such a high degradation for the ring topology performance shown in Figure 5.1.

On the other hand, the measurement fusion chain topology achieved high estimation quality even with low event-rates as it can be seen in Figure 5.4. This topology has been considered the greatest development in this research since, even though it does not beat the star topology, its consistency and computational effort are very attractive and the information available in the whole network is used in a smart manner.

Some issues about multisensor event-based state estimation are still open for future studies, such as the development of new topologies which could improve the given results in this work, or the study of different triggering mechanisms, which may be interesting from the network point of view in order to compare not only sensor nodes' arrangement but also trigger intrinsics. Besides, facing real wireless problems, such as latency or packet losses, could be the next step to continue this research.

Bibliography

[1] Marcus Thelander Andren and Anton Cervin. Event-Based State Estimation Using an Improved Stochastic Send-on-Delta Sampling Scheme. In *2016 Second International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*. IEEE, June 2016.

[2] Q. Gan and C. J. Harris. Comparison of two Measurement Fusion Methods for Kalman-Filter-based Multisensor Data Fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1):273–279, Jan 2001.

[3] Duo Han, Yilin Mo, Junfeng Wu, Sean Weerakkody, Bruno Sinopoli, and Ling Shi. Stochastic Event-Triggered Sensor Schedule for Remote State Estimation. *IEEE Transactions on Automatic Control*, 2015.

[4] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35, 1960.

[5] Karl-Dirk Kammeyer and Armin Dekorsy. *Nachrichtenübertragung*. Vieweg+Teubner Verlag, Wiesbaden, 2004.

[6] Benjamin Noack, Florian Pfaff, and Uwe D. Hanebeck. Optimal Kalman Gains for Combined Stochastic and Set-Membership State Estimation. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference On*, pages 4035–4040. IEEE, 2012.

[7] Benjamin Noack, Joris Sijs, Marc Reinhardt, and Uwe Hanebeck. *Treatment of Dependent Information in Multisensor Kalman Filtering and Data Fusion*, pages 169–192. 12 2017.

[8] X Rong Li, Zhanlue Zhao, and Vesselin P. Jilkov. Practical Measures and Test for Credibility of an Estimator. 01 2001.

[9] E. J. Schmitt, B. Noack, W. Krippner, and U. D. Hanebeck. *IEEE Control Systems Letters*.

[10] Dawei Shi, Ling Shi, and Tongwen Chen. *Event-Based State Estimation*, volume 41 of *Studies in Systems, Decision and Control*. Springer International Publishing, Cham, 2016.

[11] Joris Sijs, Benjamin Noack, Mircea Lazar, and Uwe D. Hanebeck. *Time-Periodic State Estimation with Event-Based Measurement Updates*. CRC Press, Boca Raton, FL, 2015.

[12] Norbert Wiener. *Time Series*. Number 9 in MIT. MIT Pr, Cambridge/Mass, 1977.

[13] Junfeng Wu, Xiaoqiang Ren, Duo Han, Dawei Shi, and Ling Shi. Finite-Horizon Gaussianity-Preserving Event-Based Sensor Scheduling in Kalman Filter Applications. *Automatica*, 72:100–107, October 2016.