République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Kasdi Merbah- OUARGLA

Faculté des Nouvelles Technologies de l'Information et de la Communication

Département d'Electronique et des Télécommunications

Thèse

Présentée en vue de l'obtention du **diplôme** de **DOCTORAT** LMD

En : Automatique

Spécialité : Automatique et système

Par : Gasmi Elhadi

Sujet

Observation à échantillonnage événementiel d'un système cyber-physique

Soutenue publiquement, le 18/09/2024, devant le jury composé de :

SAMAI Djamel	Pr	Université de Ouargla	Président
SID Mohamed Amine	Pr	Université de Setif	Directeur de thèse
HACHANA Oussama	Pr	Université de Ouargla	Co-directeur de thèse
SARI Bilal	Pr	Université de Setif	Examinatrice
ROUABAH Boubakeur	MCA	Université de Ouargla	Examinateur
MAHBOUB Mohamed Abdelbasset	MCA	Université de Ouargla	Examinateur

Democratic and Popular Republic of Algeria

Ministry of Higher Education and Scientific Research

University Kasdi Merbah, Ouargla

Faculty of new information and communication technologies

Department of Electronics & Telecommunications

Dissertation submitted to the Department of Electronics and Telecommunications in Candidacy for the Degree of **Doctor** 3rd Cycle LMD

In : Automatics

Speciality : Automatics & systems

By : Elhadi GASMI

Title

Event-trigger state estimation in cyber-physical system

Presented publicly on 09/18/2024, in front of jury members:

Djamel SAMAI	Pr	University of Ouargla	President
Mohamed Amine SID	Pr	University of Setif	Supervisor
Oussama HACHANA	Pr	University of Ouargla	Co-Supervisor
Bilal SARI	Pr	University of Setif	Examiner
Boubakeur ROUABAH	MCA	University of Ouargla	Examiner
Mohamed Abdelbasset MAHBOUB	MCA	University of Ouargla	Examiner

Academic Year: 2023 / 2024

Dedication

This thesis is dedicated to my family, whose unwavering support and encouragement have been my guiding light throughout this journey.
To my parents, for their endless love and belief in me, which have been the foundation of my strength and perseverance.
To my mentors and advisors, for their invaluable guidance and wisdom, which have been instrumental in shaping this work.
And to all my friends and colleagues, for their support and camaraderie, which have made this endeavor an enriching experience. Thank you for inspiring me to reach for the stars.

Acknowledgements

I would like to express my deepest gratitude to everyone who has supported and guided me throughout the course of this thesis.

First and foremost, I am profoundly grateful to my advisor, Pr Sid Mohamed Amine, for their invaluable guidance, insightful feedback, and continuous encouragement and for granting me the opportunity to successfully complete this PhD research. Your expertise and dedication have been instrumental in the completion of this work. I am also thankful to my co-supervisor, Pr Oussama Hachana for his invaluable assistance, feedback, and insightful comments.

I would also like to thank the members of my thesis committee, Pr Djamel SAMAI, Pr Bilal SARI, Dr Mohamed Abdelbasset MAHBOUB, and Dr Boubakeur ROUABAH, for their time, valuable suggestions, and constructive criticisms.

I would like to express my sincere thanks and gratitude to Pr Simo Särkkä for his invitation and hospitality during my short visit to Finland. My experience at Aalto University was rich and fruitful thanks to his guidance and valuable advice. I learned a lot through the fruitful questions and discussions, and the opportunities I had to explore new fields during my meetings with his research team, and I look forward to maintaining contact with you and your team in the future.

A special thanks to my colleagues and friends in the Department, whose camaraderie, collaboration, and moral support have made this journey more enjoyable and intellectually stimulating.

I am immensely grateful to my family for their unconditional love, patience, and unwavering support. To my parents, thank you for instilling in me the values of perseverance and dedication. To my siblings, thank you for always being there for me.

Thank you all for being a part of this journey.

Abstract

This thesis focuses on event-triggered state estimation problems within the context of cyber-physical systems (CPSs), aiming to develop new event-triggered estimators for nonlinear and Gaussian/non-Gaussian systems. Event-triggered state estimation has been a prominent area of systems research for several decades, with successful applications in diverse fields such as signal processing, target tracking, and navigation systems. This approach offers a promising solution to data traffic congestion by facilitating aperiodic, event-triggered information exchange between sensors and estimators.

The motivation for this research stems from the resource limitations inherent in CPS applications, such as wireless sensor networks, and the increased computational burden associated with calculating optimal state estimates under event-triggering conditions. In this work, we address several practical challenges encountered in the field and endeavor to advance the state of the art in event-triggered state estimation.

In the first part, we provide a brief introduction to the problem of systems under event-triggering conditions and outline the main theory using probabilistic inference, where the problem is addressed with Bayesian state estimation.

In the second part, we present the necessary theory for event-triggered state estimation, where the Gaussian assumption are discussed to approximate the posterior Probability Density Function (pdf). Here, the problem is reduced to one of the approximated nonlinear type of Kalman filters.

In the next part of this research, we assume that the posterior pdf is no longer to be Gaussian. Therefore, we develop an event-triggered particle filter to approximate the non-Gaussian posterior. The pdfs are approximated based on Monte Carlo simulations using a set of particles and weights. However, computing the particle weights based on the event trigger condition can lead to a computational burden. To address this issue, a Bayesian constraint is developed.

Finally, we study the effect of packet dropouts on the performance of state estimators, specifically focusing on particle filters. Packet dropouts, caused by imperfect communication channels, are unavoidable when information is transmitted through a communication network. We first develop a nonlinear particle filter to reduce the estimation error. Using a special form of the sequential Monte Carlo algorithm, the posterior distribution is approximated, and the corresponding minimum mean-squared error is derived. By contrasting the error covariance matrix with the posterior Cramér-Rao lower bound, the estimator's performance is assessed.

Keywords: Event Trigger, Remote estimation, Particle filter, Gaussian distribution, Nonlinear filtering, Packet Dropout.

ملخص

تركز هذه الرسالة على مشاكل تقدير الحالة المستندة إلى الحدث في سياق الأنظمة الفيزيائية السيبرانية وتهدف إلى تطوير مقدرات حالة جديدة مستندة إلى الحدث لأنظمة خطية\غير خطية وأنظمة قوسيه و الغير قوسيه. كان تقدير الحالة المستندة إلى الحدث مجالاً بارزًا في بحوث الأنظمة لعدة عقود، مع تطبيقات ناجحة في مجالات متنوعة مثل معالجة الإشارات، تتبع الأهداف، وأنظمة الملاحة. يقدم هذا النهج حلاً واعدًا لازدحام حركة البيانات من خلال تسهيل تبادل المعلومات المستند إلى الحدث بشكل غير دوري بين أجهزة الاستشعار والمقدرات. ينبع الدافع لهذا البحث من القيود الموجودة في موارد تطبيقات الأنظمة الفيزيائية السيبرانية، مثل شبكات الاستشعار اللاسلكية، والعبء الحسابي المتزايد المرتبط بحساب تقديرات الحالة المثلى في ظل ظروف التشغيل المستندة إلى الحدث. في هذا العمل، نتناول عدة تحديات عملية موجودة في هذا المجال ونسعى إلى تطوير حالة الفن في تقدير الحالة المستندة إلى الحدث.

في الجزء الأول، نقدم مقدمة موجزة لمشكلة الأنظمة في ظل ظروف التشغيل المستندة إلى الحدث ونحدد النظرية الرئيسية باستخدام الاستدلال الاحتمالي، حيث يتم تناول المشكلة باستخدام تقدير الحالة البايزية. في الجزء الثاني، نقدم النظرية اللازمة لتقدير الحالة المستندة إلى الحدث، والتي يتم تقريبها بناءً على فرضية القوسيه للتوزيعات الاحتمالية اللاحقة. هنا، يتم تقليل المشكلة إلى شكل خالي من المشتقات من مرشحات كالمن.

في الجزء التالي من هذا البحث، نفترض أن التوزيع الاحتمالي لم يعد قوسي وبالتالي، نقوم بتطوير مرشح جسيمي مستند إلى الحدث لتقريب التوزيع الغير القوسي. يتم تقريب التوزيعات الاحتمالية بناءً على محاكاة قانون كارلو باستخدام مجموعة من الجسيمات والأوزان.

أخيرًا، ندرس تأثير فقدان الحزم على أداء مقدرات الحالة، مع التركيز بشكل خاص على المرشحات الجسيمية. يكون فقدان الحزم، الناجم عن قنوات الاتصال غير المثالية، أمرًا لا مفر منه عند نقل المعلومات عبر شبكة الاتصال. نقوم أولاً بتطوير مرشح جسيمي غير خطي لتقليل خطأ التقدير. باستخدام شكل خاص من خوارزمية مونتي كارلو التسلسلية، يتم تقريب التوزيع اللاحق، ويتم اشتقاق الحد الأدنى لخطأ التربيع الوسطي. من خلال مقارنة مصفوفة التغاير للخطأ مع الحد الأدنى لصيغة كرامر، يتم تقييم أداء المقدّر.

الكلمات المفتاحية: التشغيل المستند إلى الحدث، التقدير عن بُعد، مرشح الجسيمات، توزيع غاوسي، التصفية غير الخطية، فقدان الحزم.

Contents

List of Figures

List of Symbols

Symbols	Description
$\mathbb{R}$	Real numbers
$\mathbb{R}^n$	Real-valued column vectors of length n
$\mathbb{R}^{nxm}$	Real-valued column vectors of length n x m
$\mathbb{I}_n$	Identity matrix of size n x n
$(.)^T$	Matrix transpose
$(.)^{-1}$	Inverse Matrix
$N(.)$	Normal distribution
$Log(.)$	Logarithm
e_i	Unit vector in the direction of the coordinate axis i.
$\delta(.)^T$	Dirac delta function
$\int_{(.)}$	Integration over set (.)
$p(.)$	Probability density function (pdf)
$\mathbb{E}[.\|.]$	Conditional expectation
$Cov[.\|.]$	Conditional covariance
$p(.\|.)$	Conditional pdf
$n(\mu, \sigma)$	Normal distribution with mean μ and covariance σ
x_k	State
z_k	Measurement in the sensor
w	State noise
v	Measurement noise
Q	State noise covariance
R	Measurement noise covariance
$\bar{z}$	last transmitted Measurement
y_k	Measurement in the sensor
$f(.)$	The state function
$g(.)$	The measurement function
π_0	Initial probability density function
γ_k	Trigger variable
λ_k	Packet loss variable
Ξ_k	Trigger set
$\bar{\Xi}_k$	Non-Trigger set
ξ	Trigger threshold
$P_.$	Estimated covariance

P^{xy}	Cross covariance
$K_{.}$	Kalman gain
$\frac{\partial(.)}{\partial x}$	Jacobean for x
ω	Particle weights
ψ^i	Sigma point
Δ	Second partial derivative operator
j	Fisher information matrix
M	Number of particle
Ω	Constraint set
$\phi(.)$	Constraint function
$L_{(.)}$	Decision variable
$m_*(.)$	Trigger decision variable
$\bar{L}_*(.)$	Non-trigger decision variable

Abbreviations

pdf Probability Density Function

DSSM Dynamic State-Space Model

AN Agent Network

SN Sensor Network

KF Kalman Filter

EKF Extended Kalman Filter

CPS Cyber-Physical System

NCS Network Control System

EbSE Event-based State Estimation

ETSE Event-Trigger State Estimation

PF Particle Filter

ETPF Event Trigger Particle Filter

SOD Send On Delta

IBT Innovation Level Based Event Trigger

ETKF Event Trigger Kalman Filter

ETEKF Event Trigger Extended Kalman Filter

CKF Cubature Kalman Filter

ETCKF Event Trigger Cubature Kalman Filter

ETCKFPD Event-triggered Cubature Kalman Filter with Packet Dropout

ETUKF Event Trigger Unscented Kalman Filter

ETUKF Event Trigger Unscented Kalman Filter with Packet Dropout

SIR Sequential Importance Sampling Resampling

RMSE Root Minimum Mean Square Error

EKF Extended Kalman Filter

UKF Unscented Kalman Filter

CKF Cubature Kalman Filter

SIB Stochastic Innovation-Based

MCS Monte Carlo Simulation

SIS Sequential Importance Sampling

PCRLB Posterior Cramér-Rao Lower Bound

Chapter 1

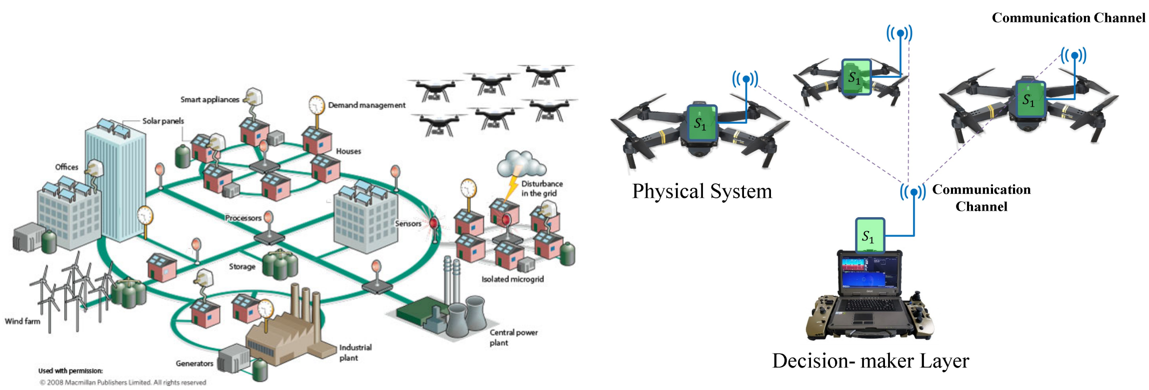
Introduction

1.1 Overview

This chapter will first describe the broad problem domain addressed by the body of work presented in this thesis. After this, a compact literature overview of related work in the field is given. Finally, a summary of the specific contributions of the work presented in this dissertation as well as a overview of the thesis itself is provided.

1.2 Cyber-physical systems

Cyber-Physical Systems (CPSs) are often regarded as "smart" systems that interact with humans through various interfaces, enhancing the physical world by integrating communication, computation, and control. Essentially, CPSs can be envisioned as (wireless) networked systems comprising numerous distributed, interconnected, and autonomously functioning nodes, all managed and regulated by computer-based algorithms within the cyber realm. In other words, CPSs represent a seamless integration of physical components, computational processes, software elements, and networking technologies, each functioning on a distinct scale.



(a) (b)

Figure 1.1: Cyber-Physical Systems (a) Societal-Scale CPS and indistry 4.0 [4] (b) Networked group of drones as CPS.

CPSs are revolutionizing networks of physical and computational components, playing a pivotal role in the design and development of next-generation airplanes [1], fully autonomous vehicles [2], and hybrid vehicles [3]. On the other hand, a specialized branch known as human-in-the-loop CPSs is being explored, where brain signals control physical objects to deliver futuristic smart services, potentially enhancing various aspects of our quality of life.

In the near future, CPSs are expected to permeate every facet of our world, bringing about unmistakable and significant changes. Just as the rapid advancement of the internet has transformed our interaction with information, CPSs will further extend the boundaries of information technology through their interactions between the physical world and humans. The impact of CPSs on social structures, the economy, and society is undeniable, as they represent a new generation of systems that combine computing and communication efficiencies with physical and engineered systems.

However, these exciting benefits come with a downside: the effects on the physical world are uncertain, presenting numerous challenges and research opportunities within the CPS domain.

Recent advancements in distributed signal processing techniques for CPSs are swiftly integrating into various aspects of our daily lives. A notable innovation in this field is the production of small, affordable electronic devices (sensors) with various modalities, capable of measuring a wide range of physical characteristics, such as light intensity, temperature, optical back scatter (OBS), seismic activity and fluorescence. These advanced sensors can communicate with each other and other devices, enable them to efficiently collect and store data for subsequent processing as required. Depending on the application, these sensors can transmit their observations through wired or wireless means.

By integrating numerous local processing nodes (sensors), networked systems such as sensor networks, agent networks (AN) [5], Robotic Networks [6], Camera Networks [7] and Networks of Unmanned Aerial Vehicles (UAV) [8] the CPS system are created, see in figure 1.1. These evolving sensor technologies play a crucial role in the design and functionality of these networks, facilitating their growth and enhancing their capabilities. Cyber-physical systems typically facilitate information exchange between physical and cyber components via wired or wireless communication channels. Wireless communication offers advantages like reduced wiring costs but necessitates careful management of communication resources. Generally the cps system can be structured from three layer. The physical layer that contain the physical system with sensor that sense the variation state of the system and actuator that apply any control. The second layer is the layer of transport, which represent communication channel with wired or wireless network and different nodes. the last layer is the Decision maker layer where the observation treated and the signal Processed and state estimate to um with system decision and intelligent control as presented in figure 1.2.

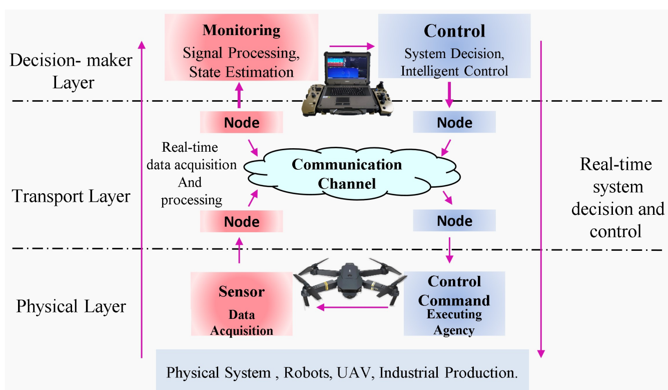


Figure 1.2: The block diagram of the CPS layer.

For instance, in the physical layer, the energy management is critical due to the wireless sensors and processors often operate on limited battery energy. Additionally, in the cps system, the state of the system tracked by one or N sensor. Thus, a large amount of data is transmitted from sensor to the decision layer via transport layer, using band-limited networks for communication between sensors, state estimators, controllers, and actuators can pose challenges such as transmission delays and packet dropouts. In the traditional approach, signals are transmitted periodically or according to a time-triggered schedule, meaning activities occur at predetermined intervals. While this approach is straightforward to implement, it often have the following drawbacks, the sensor periodically transmit the senor measurement consumes the power resources. In addition the amount of sensor data transmitted excessive bandwidth for limited communication channel which can lead

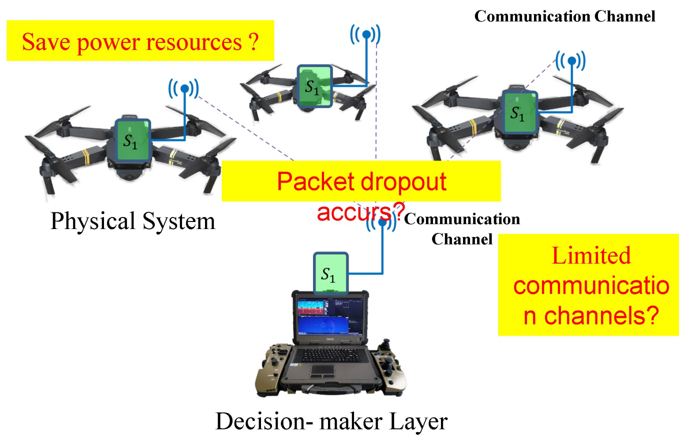


Figure 1.3: Block diagram for CPS drawbacks.

to measurement delay and packet dropout see figure 1.3.

To address the aforementioned problem and its drawbacks, communication should occur only when necessary information needs to be exchanged between system components. This can be achieved by using intelligent sensors to reduce data, thereby saving power and communication resources. Additionally, a new estimation technique needs to be designed to manage the challenges associated with reduced data see figure 1.4. Event-triggered data transfer and state estimation presents a promising solution to optimize communication efficiency in such scenarios.

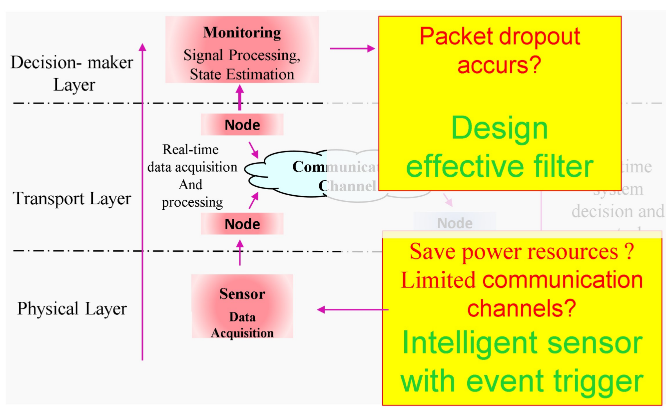


Figure 1.4: CPS with intelligent sensors.

1.3 Event-trigger State Estimation

An Event-Trigger State Estimation (ETSE) approach has been developed to address real-time communication and power constraints (such as limited bandwidth, limited battery capacity, and time delays [13,14]). Unlike the traditional time-based approach where the sensor communicate its observation periodically, in event-trigger approach the sensors measurement transmitted if the specified triggering conditions are violated which significantly reduce the number of transmission tasks. This approach not only enhances efficiency but also optimizes resource utilization in CPS. The ETSE is constructed from two categories: the first is the event trigger approach in the physical layer, which focuses on developing an event trigger mechanism for intelligent sensors. In this approach, the measurement transition is judged by an event trigger condition as shown in figure 1.5. Recently, several event-triggered sampling techniques have been developed in the literature, including [9–12]: Innovation Level Based Event Trigger (IBT), Send On Delta (SOD), covariance-based, stochastic, among others. The second category is the event trigger estimator in the decision maker layer, which provides an estimated state with acceptable performance.The integration of event-based sampling with state estimation techniques, termed as Event-based State Estimation (EbSE), introduces several design constraints. These include effectively managing missing information and addressing induced non-linear behaviors. Over the past decade, ETSE has garnered significant attention in the literature.

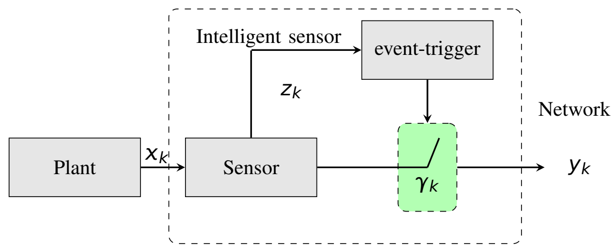


Figure 1.5: Intelligent sensor with an event trigger mechanism.

For instance, Ribeiro et al. [15] derived an optimal estimator using a binary indicator bit based on the measurement's sign to approximate Root Minimum Mean Square Error (RMSE). Sijs and Lazar [16] developed a stochastic state estimator adaptable to different event-sampling strategies. The work in [10] presents an event-based sensor data scheduler that balances communication rate and estimation quality effectively. Additionally, an innovation-based event-triggered communication algorithm for maximum likelihood state estimation is proposed in [21]. In this thesis, our work focuses on the first category, where we develop a new event trigger estimator under certain system constraints and environmental assumptions. We consider constrained CPS systems with limited bandwidth and power resources. Specifically, we address the challenge of how the remote estimator can accurately estimate the true state, such as determining the position and orientation of a mobile robot.

To estimate the true state under these constraints, we design a remote estimator based on information about the system state and the data provided by the sensors. We formulate the system model as a problem with a dynamic state space model representing the CPS system. By employing probability theory, we reason about the true state under conditions of uncertainty with measurements and the information of the trigger mechanism.

1.4 Problem formulation

We consider the nonlinear dynamic state-space model of a CPS system

$$x_{k+1} = f(x_k) + w_k \tag{1.1}$$

$$z_k = g(x_k) + v_k \tag{1.2}$$

With x_k is the state vector where the initial state x_0 sampled from π_0, z_k is the sensor measurement, $f(.)$ and $g(.)$ are nonlinear functions for the state and observation, w_k and v_k are uncorrelated white noises that follows sampled from p_w and p_v, respectively.

1.4.1 Event trigger mechanism

We consider the constraint system with limited communication and power resources. Thus, an event-triggered mechanism [9] equipped in the sensor to reduce the communication burden and energy consumption as illustrated in figure 1.5. Explicitly, after the measurement is collected by the sensor, an event trigger condition decides whether the sensor communicate its measurement into the located remote estimator. Let γ_k be the decision variable

$$\gamma_k = \begin{cases} 1, & \text{the measurement is transmited} \\ 0, & \text{the measurement is not transmited.} \end{cases} \tag{1.3}$$

If $\gamma_k = 1$, the sensor transmit the current measurement to the located remote estimator. Otherwise, no measurement is transmitted. Therefore, the triggered measurement in the estimator y_k can be represented as

$$y_k = \begin{cases} z_k, & \gamma_k = 1, \\ \{z_k \in \Xi_k\}, & \gamma_k = 0. \end{cases} \tag{1.4}$$

Where z_k is the measurement in the sensor, and Ξ_k is the trigger information (condition), For the non trigger case ($\gamma_k = 0$), the remote estimator have knowledge that the current measurement z_k belongs to the trigger condition Ξ_k that falls within the set of all possible measurement values of the sensor at time instance k that satisfy the non trigger case. Here, $\bar{z}$ represents the last transmitted measurement. The commonly used deterministic event triggered conditions that define the trigger set Ξ_k are given as following sections.

1.4.1.1 Send-on-Delta event trigger

The SOD condition stands as the most frequently employed event-triggered condition. However, in SOD condition the sensor measurement z_k only triggered if the distance between the current measurement z_k and the last transmitted one $\bar{z}$ more the threshold ξ. This concept of the SOD condition can be formulated as described in (1.5).

$$\Xi_k = \{y_k \in \mathbb{R}^m | (z_k - \bar{z})^T (z_k - \bar{z}) \leq \xi\} \tag{1.5}$$

1.4.1.2 Innovation level based event trigger

The IBT the sensor trigger its measurement if a disparity between the current observation z_k and its one-step prediction $\mathbb{E}[y_k | y_{1:k-1}]$, reflecting the level of innovation within the current measurement.

Hence, the mathematical representation of the IBT is delineated in (1.6).

$$\Xi_k = \{z_k \in \mathbb{R}^m | (z_k - \mathbb{E}[\, y_k | y_{1:k-1}])^T (z_k - \mathbb{E}[\, y_k | y_{1:k-1}]) \leq \xi\} \quad (1.6)$$

In the literature, both the SOD and IBT mechanisms exhibit certain drawbacks. The SOD mechanism imposes stringent requirements on the system, potentially failing to meet high accuracy demands for state estimation. Conversely, the IBT mechanism necessitates data return from the remote filter to the sensor, thereby increasing communication bandwidth. However, this issue can be mitigated by equipping the sensor with a local filter. Using local filter on the sensor can solve the problem of feedback from the remote estimator but require more computation resources and more cost. Thus, in our thesis we assume that the CPS unit computation is limited and we only use the SOD mechanism as the sensor trigger condition, and we design our filter based in this triggering strategy.

By using one of the SOD event trigger mechanism , all the information received at the remote estimator become as follow

$$y_{1:k} = \{\gamma_1 z_1, ...\gamma_k z_k, \gamma_1, ..., \gamma_k\} \quad (1.7)$$

In next section, using the probabilistic inference will compute the information about the state x_k given all the observation $y_{1:k}$.

1.4.2 Probabilistic Inference

Using probability theory, the ETSE problem can be considered a special case of estimating the hidden variables of the system in an optimal and consistent manner, given noisy observations. This is illustrated in Figure 1.6, which shows the system when the measurement is triggered, and Figure 1.7, which shows the system when the measurement is not triggered. In the context of sequential (recursive) probabilistic inference within discrete-time nonlinear dynamic systems, the hidden system state x_k, initialized with a probability density $p(x_0)$, evolves over time. Here, k represents the discrete time index, and the system state evolves as an indirectly or partially observed first-order Markov process, following the conditional probability density $p(x_k|x_{k-1})$. The observations y_k are conditionally independent given the state and are generated according to the conditional probability density $p(y_k|x_k)$. The state transition density $p(x_k|x_{k-1})$ is determined by the function f and the process noise distribution $p(w_k)$, while the observation likelihood $p(y_k|x_k)$ is fully determined by the function h and the observation noise distribution $p(v_k)$. The problem statement of sequential

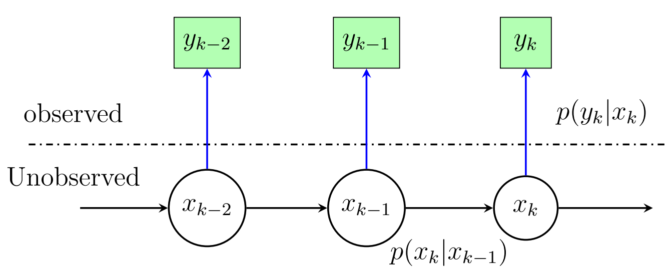


Figure 1.6: Black diagram of probabilistic dynamic state-space model with noisy measurement.

probabilistic inference in the Dynamic State-Space Model (DSSM) framework, as discussed above,

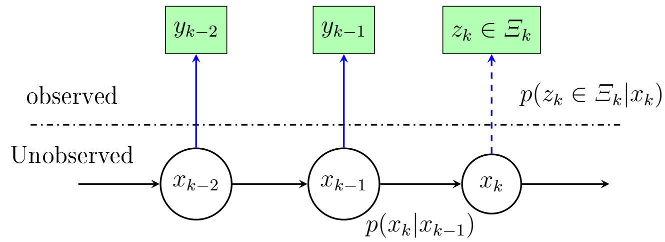


Figure 1.7: Black diagram of probabilistic dynamic state-space model with non trigger measurement.

can now be framed as follows: How do we optimally estimate the hidden system variables in a recursive fashion as incomplete and noisy observations become available online? This issue lies at the heart of numerous real-world applications in various fields such as engineering, bioinformatics, environmental modeling, statistics, finance, econometrics, and machine learning. In a Bayesian framework, the posterior filtering density is updated recursively to incorporate new observations and provide the best possible estimate of the hidden variables. In other word, the posterior described as $p(x_k|y_{1:k})$ which is the pdf of the state x_k given all the observation $y_{1:k} = \{y_1, y_2, ..., y_k\}$

The posterior filtering density constitutes the complete solution to the sequential probabilistic inference problem. It allows us to calculate any "optimal" estimate of the state, such as the conditional mean, by recursively updating the probability distribution of the hidden variables as new, incomplete, and noisy observations become available. This approach ensures that the most accurate and up-to-date estimates of the state are maintained over time with conditional mean

$$\hat{x}_{k|k} = \mathbb{E}[\,x_k|y_{1:k}] = \int_{\mathbb{R}^n} x_k p(x_k|y_{1:k}) dx_k \tag{1.8}$$

The problem statement can thus be reformulated as: How do we recursively compute the posterior density as new observations arrive online?

1.5 Recursive Event-Triggering Bayesian estimation

In this section, we introduce the recursive event-triggering Bayesian estimation as an optimal solution to the event trigger posterior pdf. However, with the dynamic state-space model of the system and Bayes' rule, the arrived observations and the information contained in the event trigger condition are used to update the recursive pdf.

1.5.1 Recursive Bayesian estimation

We assume that the state vector x_k and the measurement y_k are Markovian, as detailed in Appendix 2. The recursive update of the posterior density can be expanded and factored into the following form [32, p. 5]:

$$
\begin{aligned}
p(x_k|y_{1:k}) &= \frac{p(y_{1:k}|x_k)p(x_k)}{p(y_{1:k})} \\
&= \frac{p(y_k, y_{1:k-1}|x_k)p(x_k)}{p(y_k, y_{1:k-1})} \\
&= \frac{p(y_k|y_{1:k-1}, x_k)p(x_k)p(y_{1:k-1}|x_k)}{p(y_k|y_{1:k-1})p(y_{1:k-1})}
\end{aligned}
\tag{1.9}
$$

Using of Bayes rule (see Appendix A.1) to the last rights term in equation we have,

$$
\begin{aligned}
p(x_k|y_{1:k}) &= \frac{p(y_k|y_{1:k-1}, x_k)p(x_k)p(x_k|y_{1:k-1})p(y_{1:k-1})}{p(y_k|y_{1:k-1})p(y_{1:k-1})p(x_k)} \\
&= \frac{p(y_k|y_{1:k-1}, x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}.
\end{aligned}
\tag{1.10}
$$

Then using the conditional independence of the observation given the state,

$$
p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}. \tag{1.11}
$$

Considering the previous posterior pdf $p(x_{k-1}|y_{1:k-1})$ has been obtained at time $k-1$, then the a prior conditional pdf is derived as,

$$
p(x_k|y_{1:k-1}) = \int_{\mathbb{R}^n} p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}. \tag{1.12}
$$

Here, $p(y_k|x_k)$ is the likelihood of the observation given the state, and $p(x_k|x_{k-1})$ is the state transition density specifies the knowledge of the model (2.1). The integral represent the prior density updated with the new observation y_k. the pdf $p(y_k|y_{1:k-1})$ is normalizing factor given by,

$$
p(y_k|y_{1:k-1}) = \int_{\mathbb{R}^n} p(y_k|x_k)p(x_k|y_{1:k-1})dx_k. \tag{1.13}
$$

We note that the derivation of the Recursive Bayesian estimation can be obtained recursively in two stages; (I) the time update where we suppose we have the pdf $p(x_{k-1}|y_{1:k-1})$ at time $k-1$, based on the system model in 2.1 the prior density can obtained. (II) the second stage of measurement update can obtained based on the prior in time update and the available measurement y_k and the information based on event trigger $z_k \in \Xi_k$ and the process model in (2.1).

1.5.2 Event-Triggering Bayesian estimation

In this section, the recursive Bayesian equations with of event trigger are presented as time update and measurement update.

1.5.2.1 Time Update

We note that the prior pdf $p(x_{k-1}|y_{1:k-1})$ does not depend on the measurement z_k at time k. Thus we derive the prior pdf as similar to equation 1.12,

$$p(x_k|y_{1:k-1}) = \int_{\mathbb{R}^n} p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}. \tag{1.14}$$

1.5.2.2 Measurement update

In the presence of an event trigger scheduler as described in (1.3), the measurements z_k are not available to the estimator at all time instants k. Thus, the update of the equation depends on whether or not the current sensor measurement z_k has been communicated.

When the event trigger condition is met, the measurement z_k is transmitted, and the posterior density is updated accordingly. However, when the condition is not met, the estimator must rely solely on the predicted state from the time update step and the trigger information. The posterior density update can thus be expressed in two cases:

(a) if $\gamma_k = 1$ the sensor measurement z_k transmitted to the located remote estimator. The observation $y_{1:k}$ is updated to $y_{1:k} = y_{1:k-1} \cup \{z_k\}$, using Bayesian rule, the posterior can be updated as,

$$\begin{aligned} p(x_k|y_{1:k}) &= p(x_k|y_{1:k-1}, y_k) \\ &= \frac{p(z_k|x_k)p(x_k|y_{1:k-1})}{p(z_k|y_{1:k-1})}. \end{aligned} \tag{1.15}$$

(b) if $\gamma_k = 0$ the measurement is non triggered, the posterior update based on the information in the trigger condition where $\{z_k \in \Xi_k\}$. Consequently,

$$\begin{aligned} p(x_k|y_{1:k}) &= p(x_k|y_{1:k-1}, \{z_k \in \Xi_k\}) \\ &= \frac{p(\{z_k \in \Xi_k(\bar{z})\}|x_k)p(x_k|y_{1:k-1})}{p(\{y_k \in \Xi_k\}|y_{1:k-1})}. \end{aligned} \tag{1.16}$$

where

$$p(z_k \in \Xi_k|x_k) = \int_{\Xi_k} p(z_k|x_k)dz_k. \tag{1.17}$$

The multi-dimensional integrals integral equation in (1.9) to (1.13) can be solved and the optimal recursive solution can be tractable. However, for linear and Gaussian system, the Kalman Filter (KF) is the most closed-form recursive solution for this problem. For nonlinear constrained system with event trigger, the system no longer to be Gaussian. Thus, the equation of multi-dimensional integrals of the event trigger recursive solution as in equation (1.14) to (1.16), are intractable and approximate solutions must be used. Next we review various the approximate approach for finding the optimal Bayesian recursive solution.

1.6 Event-Triggering numerical approximation

Over the past few decades, various approximate solutions have emerged to address the event-triggering recursive Bayesian estimation problem across multiple disciplines. These approaches can be broadly categorized into three main groups:

* **Event-Triggering Gaussian Approximation:** based on Gaussian assumptions for the conditional distributions of the states given the available hybrid measurement information, the posterior state estimate can be approximated using an event-trigger mechanism. This approximation approach utilizes linearization with the Extended Kalman Filter Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), or Cubature Kalman Filter (CKF).
  - **Event Trigger Extended Kalman Filter (ETEKF)**, is a variation of the EKF, which is used for state estimation in nonlinear systems. The ETEKF incorporates an event-triggered mechanism that decides when to perform updates based on specific events or conditions, rather than at regular time intervals. This approach aims to reduce computational and communication burdens by only processing measurements that provide significant new information about the system's state [17].
  - **Event Trigger Unscented Kalman Filter (ETUKF)** is an adaptation of the UKF that incorporates an event-triggered mechanism to determine when to perform updates. This approach can significantly reduce computational and communication overhead, especially in systems with limited resources [18, 19].
  - **ETCKF** is an adaptive estimation technique used in CPS and other domains where sensor data is intermittent or triggered by specific events rather than transmitted continuously. It extends the principles of the CKF to optimize state estimation under event-triggered conditions, reducing communication and computational burdens by only updating estimates when triggered by significant events or changes in the system. This approach is valuable in systems with limited resources or where continuous data transmission is impractical or costly [20].
* **Event-Triggering sum Gaussian Approximation :** by approximating the uniform distribution with the sum of a finite number of Gaussian distributions [16].
* **Event-Triggering Sequential Monte-Carlo Approximation :** is an adaptation of the Particle Filter (PF) that incorporates an event trigger technique and uses the Monte Carlo method to approximate the posterior distribution of the state variables with particles and weights [32, 35, 36].

1.7 Research Objectives and Work Overview

Over last decade, the new developed remote estimator based event trigger are required in constraint CPS system. However, under reduced data, this last can guaranteed the estimation performance and save the communication and power resources. In [10], an event trigger Kalman filter for linear system based IBT, where the pdf of the system computed under the Gaussian assumption. The authors in [12], proposed an event-triggered RMSE filter to mountain Gaussian characteristics . However, in real time CPS application the most of system are nonlinear, thus nonlinear event trigger estimator are required. An event-triggered based EKF distributed resilient filter are proposed in [17]. Using the structure of the CKF, an event-triggered nonlinear estimator is proposed in [20]. Furthermore, under the Gaussian assumption and in both linear and nonlinear systems, researchers aim to balance communication constraints and estimation accuracy using different event-trigger estimators with various strategies [22–30].

The Gaussian filters mentioned above can achieve acceptable accuracy in systems with determined non-linearity. However, in highly non-linear CPS applications, the accuracy of these filters degraded. Furthermore, when using event-triggered sensor scheduling algorithms [31], the Gaussian assumptions no longer hold.

In addressing the aforementioned shortcomings, PF have proven to be highly effective [32]. based on Monte Carlo Simulation (MCS) method, these filters perform a numerical approximation for the nonlinear filtering problem.

In the realm of event-trigger state estimation, PF have received attention from researchers, particularly in addressing missed measurements during non-triggering instances and without Gaussian assumption of the posterior pdf. Reference [33] explores the application of event-trigger approach in nonlinear filtering and illustrates it through the use of a particle filter. Sid and Chitraganti expand on this concept with an event-trigger particle filter [34], and further refine it for Bayesian constrained posterior [35], where likelihood computation involves numerical integration. Davar and Mohammadi developed an event-trigger linear particle filter specifically designed with SOD condition [36]. Liu et al. [37] employ the PF as a central event-trigger estimator, using simpler local estimators to manage sensor triggering conditions. In [38], an event-based auxiliary PF is introduced, utilizing the likelihood approximation similar to that in [16], but with improved performance compared to bootstrap filter implementations.

However, packet loss is a typical communication constraint that must be considered in estimation design [39,40] The uncertainties arising from packet loss can significantly degrade the performance of Networked Systems and, in certain scenarios, destabilize system precipitate. Thus, it is imperative to thoroughly evaluate the impact of these uncertainties on the entire system. There has been limited study on the performance of event-trigger estimators under packet loss. In [41], an enhanced EKF under event-trigger sampling and packet loss approximating the non-linear pdf's through a first-order Taylor series around the current mean and covariance. However, the event trigger EKF effectiveness is most pronounced in scenarios with moderate non-linearity.Another approach, the Discrete-time Event-triggered Unscented Kalman Filter (ETUKF), builds upon the Unscented Kalman Filter (UKF) as detailed in [42]. The UKF avoids the limitations of local linearization by approximating statistical properties using carefully chosen points, known as sigma points. Nevertheless, when the system dimension exceeds a certain threshold (typically $n > 3$), the central sigma point can acquire a negative weight, potentially leading to non-positive definite covariance matrices and filtering divergence. To address this issue, the Discrete-time Event-triggered Cubature Kalman Filter with Packet Dropout (ETCKFPD) is proposed in [43]. the ETCKFPD employs a set of deterministic weighted points where all weights are positive and equal, ensuring numerical stability suitable for high-dimensional state space models. However, both ETCKFPD and Event Trigger Unscented Kalman Filter with Packet Dropout (ETUKF) algorithms are constrained by their reliance on unimodal distribution systems and the assumption of Gaussian pdf. In the presence of packet dropout, where sensor measurement information may be incomplete, these estimators may fail to update effectively, resulting in non-Gaussian posterior distributions. Nevertheless, previous studies have neglected to investigate the behavior of event-trigger non-Gaussian posterior of nonlinear systems under packet loss. This oversight has motivated our research, which seeks to bridge this gap in the literature. Specifically, our study aims to tackle the challenging task of approximating non-Gaussian posterior distributions in the presence of event triggering and packet dropout. Building on the discussions mentioned earlier, in this thesis we first aim to combine and compare the recent event trigger Gaussian and non-Gaussian estimator as the state of the art, then we extend our study for investigate the same problem event trigger particle filter setting as discussed in [37], but with an additional focus on the occurrence of packet loss between the sensor and the estimator.

1.8 Contributions and Thesis Outline

The following three main objectives of this thesis were identified:

- Comparative study between recent event trigger Gaussian filters
- An event trigger particle filter for non-Gaussian and nonlinear system is discussed, this filter be alternative to the event trigger Kalman filters based Gaussian assumption.
- We study the effect of the packet loss during data transmission from the system to the remote state estimator in the communication channels, and we design an event-triggered PF with packet loss.

In Chapter 2 an optimal Gaussian approximation for event trigger recursive Bayesian estimation, introduces the most Gaussian filter based Kalman filter. This chapter covers much of the introductory motivation of why a better solution than the Gaussian filter based Kalman filter is needed in the case of event trigger.

In Chapter 3 a non-Gaussian event-triggered approximation based Particle filter approach for nonlinear state space model is developed. take into account the in the presence that the Gaussian no longer be saved, the multi-dimensional integrals integral equation of the posterior density are approximated by Sequential Monte-Carlo Approximation, We show the advantages of employing particle filter over the more established and more explored extended Kalman filters and unscented Kalman filters, frequently used in the literature, to estimate the nonlinear systems. We argue that better estimates can be obtained using the PF and justify our claims in our simulations.

In Chapter 4 a non-Gaussian event-triggered approximation based Particle filter approach for nonlinear state space model with packet loss is developed. Where the SOD mechanism is proposed. random variables obeying the Bernoulli distribution model the packet loss. Subsequently, a specialized form of the sequential Monte Carlo algorithm is employed to approximate the posterior distribution and derive the corresponding minimum mean-squared error. The performance of the estimator is assessed by contrasting the error covariance matrix with the posterior Cramér-Rao lower bound.

In chapter 5 concludes the thesis and future research work.

1.9 Publications

- E. Gasmi, M. A. Sid and O. Hachana, “Event-Triggered State Estimation Using Particle Filtering Approach,” *2nd International Conference on Advanced Electrical Engineering (ICAEE)*, October 2022, Algeria.
- E. Gasmi, M. A. Sid and O. Hachana, “Comparison of event-based state estimation UKF and EKF for Practical WAMS Applications in Smart Grid,” *International Conference on Sustainable Energy and Advanced Materials*, 2021, Algeria.
- E. Gasmi, M. A. Sid and O. Hachana, “Nonlinear event-based state estimation using particle filter under packet loss," *ISA Transactions*, vol. 144, pp. 176-187, 2024.

1.10 Conclusion

In this chapter , we discussed a brief introduction about CPS and event trigger state estimation. However, the event trigger state estimation problem assumed as recursive probabilistic problems as posterior distribution. The recursive posterior simplified using Bayesian method into weighted integral which will be approximated using Gaussian /non-Gaussian approaches in this thesis.

Chapter 2

Event-Trigger Gaussian Approximation Kalman filters derivatives

2.1 Introduction

In this chapter, we will first discuss the event trigger Bayesian problem based Gaussian approximation that reduced to event trigger Kalman filters. Then, we review the Gaussian approximate kalman filters where the update step computed based different event trigger strategy. However, the obtained Gaussian weights will computed based on the proposed approximation method for the system model and we sum up with event trigger Kalman filters algorithm. Finally, we validate the discussed Gaussian filters with simulation experiment.

2.2 Problem Statement

Recalling from chapter 1, the event trigger system with nonlinear system model as follows,

$$\begin{aligned} x_{k+1} &= f(x_k) + w_k \\ z_k &= g(x_k) + v_k. \end{aligned} \tag{2.1}$$

$$y_k = \begin{cases} z_k, & \gamma_k = 1, \\ \{z_k \in \Xi_k\}, & \gamma_k = 0. \end{cases} \tag{2.2}$$

With x_k is the state vector where the initial state x_0 sampled from π_0, z_k is the sensor measurement, $f(.)$ and $g(.)$ are nonlinear functions for the state and observation, w_k and v_k are white Gaussian noises. The SOD [9] are proposed to save the communication channel and power resources. The objective of this chapter is to approximate the event trigger posterior pdf of the state x_k given all the measurement and information $y_{1:k}$ at the remote estimator ($p(x_k|y_{1:k})$) under the Gaussian assumption as a normal distribution with expected mean $\hat{x}_{k|k}$ and covariance $P_{k|k}$ where the posterior pdf approximated as $p(x_k|y_{1:k}) \approx N(x_k|\hat{x}_{k|k}, P_{k|k})$.

Next we review the Gaussian approximation for the event trigger posterior pdf, where the mean and covariance are computed based on the event trigger condition and the approximated Gaussian weights approach.

2.3 Event-Triggering condition

In probabilistic terms, the prior pdf of the state in (1.14) and measurement in (1.16) are assumed to be Gaussian and given as,

$$\begin{aligned} p(x_k|y_{1:k-1}) &= N(x_k|\hat{x}_{k|k-1}, P_{k|k-1}) \\ p(y_k|y_{1:k}) &= N(y_k|\hat{y}_{k|k-1}, P^y_{k|k-1}). \end{aligned} \tag{2.3}$$

where $\hat{x}_{k|k-1}$, $P_{k|k-1}$ and $\hat{y}_{k|k-1}$, $P^y_{k|k-1}$ are the prior state and measurement mean and covariance with state and measurement model given as,

$$\begin{aligned} p(x_k|x_{k-1}) &= N(x_k|f_{k-1}(x_{k-1}), Q_{k-1}) \\ p(z_k|x_k) &= N(z_k|g_k(x_{k-1}), R_{k-1}). \end{aligned} \tag{2.4}$$

Based on the Gaussian assumption, the posterior pdf of the state x_k given all measurements $y_{1:k}$ follow the Gaussian distribution, and it can be expressed as,

$$p(x_k|y_{1:k}) = N(x_k|\hat{x}_{k|k}, P_{k|k}). \tag{2.5}$$

The approximation of the posterior problem in (2.5) depend on the approximation of the prior pdf $p(x_k|y_{1:k-1})$ and the likelihood pdf $p(y_k|x_k)$. The approximate of the prior distribution and predictive likelihood densities are in general difficult, thus, we solve that by approximate the joint density of x_k and y_k given all measurement $y_{1:k-1}$ as follows,

$$\begin{aligned} p(x_k, y_k|y_{1:k-1}) &= p(y_k|x_k)p(x_k|y_{1:k-1}) \\ &\approx N(\begin{bmatrix} x_k \\ y_k \end{bmatrix}; \begin{bmatrix} \mu_k(x_{k-1}), P_k(x_{k-1}) \end{bmatrix}). \end{aligned} \tag{2.6}$$

Where $\mu_k(x_{k-1}), P_k(x_{k-1})$ are the mean and covariance in function of x_{k-1} for the joint distribution. Before proceeding the Gaussian approximation for Eq. (2.6), a useful lemma is recalled from [38].

Lemma 1 *Given the joint distribution of two correlated Gaussian distributed variables $x \in R^n$ and $y \in R^m$,*

$$p(x, y|y_{1:k-1}) = N(\begin{bmatrix} x \\ y \end{bmatrix}; \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}). \tag{2.7}$$

The conditional distribution becomes,

$$p(x|y) = N(x|\mu, P). \tag{2.8}$$

$$\mu = \mu 1 + P_{12}(P_{22})^{-}1(y - \mu 2) \tag{2.9}$$

$$P = P_{11} - P_{12}(P_{22})^{-1}P_{12}^T. \tag{2.10}$$

First, the posterior $p(x_{k-1}|y_{1:k-1})$ is calculated at time $k-1$ from the belief earlier and assumed as Gaussian given as follows,

$$p(x_{k-1}|y_{1:k-1}) \approx N(x_{k-1}; \hat{x}_{k-1|k-1}, \hat{P}_{k-1|k-1}). \tag{2.11}$$

Where $\hat{x}_{k-1/k-1}$, $P_{k-1/k-1}$ are the estimated mean and covariance at time $k-1$. Then, the joint distribution of x_k, x_{k-1} given $y_{1:k-1}$ can be computed as,

$$\begin{aligned} p(x_{k-1}, x_k|y_{1:k-1}) &= p(x_{k-1}|y_{1:k-1})p(x_k|x_{k-1}) \\ &\approx N(x_{k-1}; \hat{x}_{k-1|k-1}, \hat{P}_{k-1|k-1})N(x_k; f(\hat{x}_{k-1|k-1}), Q) \\ &\approx N(\begin{bmatrix} x_{k-1} \\ x_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k-1/k-1} \\ \hat{x}_{k/k-1} \end{bmatrix}, \begin{bmatrix} P_{k-1/k-1} & D_{k/k-1} \\ (D_{k/k-1})^T & P_{k/k-1} \end{bmatrix}. \end{aligned} \tag{2.12}$$

Where $\hat{x}_{k-1/k-1}$, $P_{k-1/k-1}$ are the estimated mean and covariance at time $k-1$. The predicted mean $\hat{x}_{k/k-1}$ and covariance $P_{k/k-1}$ given as,

$$\begin{aligned}\hat{x}_{k/k-1} &= \int f(x_{k-1})N(x_{k-1};\hat{x}_{k-1/k-1},P_{k-1/k-1})dx_{k-1}\\ P_{k/k-1} &= \int f(x_{k-1})f(x_{k-1})^T N(x_{k-1};\hat{x}_{k-1/k-1},P_{k-1/k-1})dx_{k-1} - \hat{x}_{k/k-1}\hat{x}^T_{k/k-1} + Q_k\end{aligned} \tag{2.13}$$

With the cross covariance $D_{k/k-1}$ defined as,

$$D_{k/k-1} = \int (x_{k-1} - \hat{x}_{k-1/k-1})(f(x_{k-1}) - \hat{x}_{k/k-1})^T N(x_{k-1};\hat{x}_{k-1/k-1},P_{k-1/k-1})dx_{k-1} \tag{2.14}$$

By the marginalization over state x_{k-1}, the prior distribution $p(x_k|y_{1:k-1})$ can be approximated as a normal distribution with predicted mean $\hat{x}_{k/k-1}$ and covariance $P_{k/k-1}$ as follow,

$$p(x_k|y_{1:k-1}) \approx N(x_k;\hat{x}_{k|k-1},\hat{P}_{k|k-1}). \tag{2.15}$$

Next, the approximation of the likelihood function depend on the event trigger cases as in (2.2).

When $\gamma_k = 1$, the measurement is triggered. The approximation of the joint distribution of x_k and y_k given $y_{1:k-1}$ as follow,

$$\begin{aligned}p(x_k, y_k, |y_{1:k-1}) &\approx p(y_k|x_{k-1})p(x_k \mid y_{1:k-1})\\ &\approx N(\begin{bmatrix}x_k\\ y_k\end{bmatrix};\begin{bmatrix}\hat{x}_{k/k-1}\\ \hat{y}_{k/k-1}\end{bmatrix},\begin{bmatrix}P_{k/k-1} & P^{xy}_{k,k/k-1}\\ (P^{xy}_{k,k/k-1})^T & P^{yy}_{k-1/k-1}\end{bmatrix}.\end{aligned} \tag{2.16}$$

Where $\hat{x}_{k/k-1}$ and $P_{k/k-1}$ defined in (2.13), and the estimated measurement mean and covariance $\hat{z}_{k|k-1}$ and $P^{yy}_{k|k-1}$, with cross-covariance $P^{zz}_{k|k-1}$ are given as,

$$\begin{aligned}\hat{y}_{k/k-1} &= \int h(x_k)N(x_k;\hat{x}_{k/k-1},P_{k/k-1})dx_k\\ P^{yy}_{k|k-1} &= \int (h(x_k) - \hat{y}_{k/k-1})(h(x_k) - \hat{x}_{k/k-1})^T N(x_k;\hat{y}_{k/k-1},P_{k/k-1})dx_k + R_k\\ P^{xy}_{k|k-1} &= \int (x_k - \hat{x}_{k/k-1})(h(x_k) - \hat{y}_{k/k-1})^T N(x_k;\hat{x}_{k/k-1},P_{k/k-1})dx_k.\end{aligned} \tag{2.17}$$

Using the Gaussian conditioning Lemma 1, by the marginalization over the state y_k given $y_{1:k-1}$, the posterior can be approximated as,

$$p(x_k|y_k, y_{1:k-1}) \approx N(x_k;\hat{x}_{k|k},P_{k/k}). \tag{2.18}$$

with the mean and covariance $\hat{x}_{k|k}$, $P_{k|k}$ can be obtained recursively by,

$$\begin{aligned}\hat{x}_{k/k} &= \hat{x}_{k/k-1} + K_k(y_k - \hat{y}_{k/k-1}),\\ P_{k/k} &= P_{k/k-1} - K_k P^{yy}_{k|k-1}K_k^T.\end{aligned} \tag{2.19}$$

Where K_k, $\hat{y}_{k/k-1}$ is defined as ,

$$K_k = P^{xy}_{k|k-1}(P^{yy}_{k|k-1})^{-1}. \tag{2.20}$$

When $\gamma_k = 0$, in this case the data transmission is absent, and the filter lacks direct knowledge of the measurement. However, it is aware that the measurement at the current time instance is included in the information set based on the event trigger condition that proposed. Thus, the filter can update its state based on this available information. In this we review two Gaussian process update based on the trigger condition.

2.3.1 Send-on-Delta approximation

Under this event trigger strategy, the sensor decide to transmit the current measurement if SOD condition satisfied. Thus, the prior distribution will approximated as same in (2.13). However, with the lack of measurement , the posterior density is approximated based on the event trigger SOD, condition as presented in (1.5). This process is clearly discussed in [20] where the posterior density is approximated as Gaussian $p(x_k|\bar{y}_{1:k}) \approx N(x_k; \hat{x}_{k|k}, \hat{P}_{k|k})$ with estimated state and covariance are given as follows,

$$\begin{aligned}
\hat{x}_{k+1} &= \hat{x}_{k+1|k} + K_{k+1}(\bar{y} - \hat{y}_{k+1|k}) \\
P_{k+1} &= (1+a_1)\left(I - K_{k+1}(P^{xy}_{k+1})^T P^{-1}_{k+1|k}\right) \times P_{k+1|k}\left(I - K_{k+1}(P^{xy}_{k+1})^T P^{-1}_{k+1|k}\right)^T \\
&\quad + (1+a_2)\, K_{k+1}R_{k+1}K^T_{k+1} + (1 + a_1^{-1} + a_2^{-1})K_{k+1}\xi I K^T_{k+1}.
\end{aligned} \tag{2.21}$$

where $\bar{y}$ is the last transmitted measurement and the parameters a_1 and a_2 can be chosen appropriately to reduce such conservatism [44, p. 19], and the filter gain,

$$\begin{aligned}
K_{k+1} =& (1+a_1)P^{xy}_{k+1} \times [(1+a_1)(P^{xy}_{k+1})^T P^{-1}_{k+1|k} P^{xy}_{k+1} + (1+a_2)R_{k+1} \\
&+ (1 + a_1^{-1} + a_2^{-1})\xi I]^{-1}.
\end{aligned} \tag{2.22}$$

The state estimation during the measurement update process is influenced by the most recent transmitted measurement, even in its absence. However, the error covariance matrix of the system state incorporates variables a and threshold ξ, which are associated with the design of the SOD condition.

2.3.2 Innovation based approximation

In this event trigger strategy, the sensor decide to transmit the current measurement if the IBT condition satisfied. Thus, the prior distribution will approximated as same in (2.13). Furthermore, the IBT strategy involves computing an intermediate quantity for determining the trigger variable γ_k. During the update process, it is crucial to compute the one-step prediction measurement and error covariance matrix as shown in (2.17), and subsequently compute the posterior distribution. When $\gamma_k = 0$, indicating that the remote filter does not receive the measurement, the update process adjusts the measurement based on the information provided in (1.6), as discussed in detail in [17]. Thus, the posterior density in the update process can be approximated as follows:

$$p(x_k|\gamma_k = 0, y_{1:k-1}) \approx N(x_k; \hat{x}_k, P_k) \tag{2.23}$$

Where $\hat{x}_k$, $P_{k|k}$ computed recursively by,

$$\begin{aligned} \hat{x}_k &= \hat{x}_{k/k-1} \\ P_{k/k} &= P_{k/k-1} - \Psi(\xi) K_k P^{yy}_{k|k-1} K_k^T, \\ K_k &= P^{xy}_{k|k-1} (P^{yy}_{k|k-1})^{-1}. \end{aligned} \tag{2.24}$$

with $\Psi(\xi)$ is defined as

$$\Psi(\xi) = \frac{2}{\sqrt{2\pi}} \xi \exp -\frac{\xi^2}{2} [1 - D(\xi)]^{-1}. \tag{2.25}$$

where $D()$ is the Q function of standard Gaussian distribution.

2.4 Filter design

We note that to evaluate the Gaussian filter we need to compute the Gaussian weighted integrals in (2.13) and (2.17), If the dynamic function of the system process $f_k(.)$ and measurement $g_k(.)$ are linear, linear transformations can be used to compute this Gaussian weighted integrals. The Gaussian filter reduces to the linear Event Trigger Kalman Filter (ETKF) . However, if the dynamic function of the system is nonlinear. The GA filter is mainly theoretical, as its direct practical application is often challenging. The presence of model non-linearity makes it infeasible and computationally intractable to analytically compute the integrals in equations (2.13) and (2.17). Due to this limitation, certain techniques are necessary, such as linearizes the model or numerical techniques such as unscented transformation, and the three-degree spherical-radial Cubature rule.

Let the system in (2.1) is Gaussian linear state space model where the process transition function $f(x_k) = Ax_k$ and the measurement function $g(x_k) = Cx_k$. In probabilistic terms the model is described as,

$$\begin{aligned} p(x_k|x_{k-1}) &= N(x_k|A_{k-1}x_{k-1}, Q_{k-1}), \\ p(z_k|x_k) &= N(z_k|C_k x_{k-1}, R_{k-1}). \end{aligned} \tag{2.26}$$

Then the parameters of the distributions in (2.15) and (2.16) can be computed with the following even trigger Kalman filter prediction and update steps,

- The prediction step

$$\begin{aligned} \hat{x}_{k/k-1} &= A_{k-1}\hat{x}_{k-1} \\ P_{k/k-1} &= A_{k-1}P_{k-1}A_{k-1}^T + Q_{k-1}. \end{aligned}$$

- When γ_k=1, the update step,

$$\begin{aligned}
P^{yy}_{k|k-1} &= CP_{k/k-1}C^T + R_{k-1}\\
P^{xy}_{k|k-1} &= P_{k/k-1}C^T\\
K_k &= P^{xy}_{k|k-1}(P^{yy}_{k|k-1})^{-1}\\
\hat{x}_{k/k} &= \hat{x}_{k/k-1} + K_k(y_k - C\hat{x}_{k/k-1}),\\
P_{k/k} &= P_{k/k-1} - K_k P^{yy}_{k|k-1} K_k^T.
\end{aligned} \tag{2.27}$$

- When γ_k=0, the update step is given as,
  - Update the filter based on the SOD using (2.21) and (2.22).
  - Update the filter based on the IBT using (2.24) and (2.25).

2.4.1 Linearization based design

As stated in section 2.3, the optimal recursive solution for the Bayesian filtering problem can be reduced to ETKF under the assumption of all densities are Gaussian and the DSSM is linear. However, in the real time system the most of all system are typically nonlinear. Therefore, to approximate the posterior distribution for nonlinear DSSM we linearized the nonlinear system model for the prediction step, where the measurement update step is similar to ETKF. The ETEKF uses the first-order Taylor series expansion to approximate the nonlinear function where the Taylor expansion is carried out over the system function $f(x)$ and observation function $h(x)$ at $\hat{x}_k$ and $\hat{x}_{k|k-1}$, respectively, to compute the state estimate. The nonlinear function can be expressed as,

$$f(x_k) = f(\hat{x}_k) + H_k(x_k - \hat{x}_k) + \varphi(x_t, \hat{x}_k). \tag{2.28}$$

$$g(x_k) = g(\hat{x}_{k|k-1}) + G_k(x_k - \hat{x}_{k|k-1}) + \chi(x_k, \hat{x}_{k|k-1}). \tag{2.29}$$

where the matrices A_k and C_k represent the computed Jacobean of the nonlinear model given as,

$$H_k = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_k}, \quad G_k = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_{k|k-1}}. \tag{2.30}$$

The ETEKF design consists of two parts: the fist pat we linearizes the nonlinear function and compute the approximation of the prior distribution as in ETKF where the nonlinear function replaced with expression in (2.28). Thus, the integrals in (2.17) approximated as,

$$\begin{aligned}
\hat{x}_{k/k-1} &= f(\hat{x}_{k-1})\\
P_{k/k-1} &= H_{k-1}P_{k-1}H^T_{k-1} + Q_{k-1}.
\end{aligned} \tag{2.31}$$

with predicted measurement,

$$\hat{y}_{k/k-1} = G_k(\hat{x}_{k-1}). \tag{2.32}$$

For the update process of ETEKF, corresponding to the event triggered sampling strategy designed above, be similar as the derivation of update process for event ETKF. When $\gamma_k = 1$, the observation y_k is received at the remote center so that the update process of ETEKF is the same as presented in (2.27). When $\gamma_k = 0$, y_k is not received at the remote center so that the update follow the approximated posterior $p(x_k|\gamma_k = 0, y_{1:k-1}) \approx N(x_k; \hat{x}_k, P_k)$ with parameter mean and covariance based on SOD update using (2.21) and (2.22), or IBT update using (2.24) and (2.25). As a result, the filtering steps can be summarized as in Algorithm 2.1.

Algorithme 2.1 : Event trigger Extended Kalman filter (ETEKF)

$[\hat{x}_k, P_k]$:=**ETKF**$(\hat{x}_{k-1}, P_{k-1}, z_k, R_k, Q_k, \xi)$

- **Prediction Step:**
  - Compute the predicted state mean covariance $\hat{x}_{k|k-1}$, $P_{k|k-1}$

$$A_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1}}$$
$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1})$$
$$P_{k|k-1} = A_{k-1} P_{k-1} A_{k-1}^T + Q_k$$

  - Compute the predicted state mean covariance $\hat{y}_{k|k-1}$, $P_{k|k-1}^{yy}$

$$C_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k|k-1}}$$
$$\hat{y}_{k|k-1} = g(\hat{x}_{k-1})$$
$$P_{k|k-1}^{yy} = C_k P_{k-1} C_k^T + R_k$$

- **Update Step:**
  - When γ_k=1,
    * Update the filter using (2.27)
  - When γ_k=0,
    * Update the filter based on the SOD using (2.21) and (2.22).
    * Update the filter based on the IBT using (2.24) and (2.25).

2.4.2 Unscented transform based design

The ETEKF based on linearization approach can approximate the event trigger posterior density and aims to communication burden by performing updates of the state estimate and covariance using the information contained in the trigger rule when non triggering case. This is particularly useful in networked systems where resources are limited. However, the ETEKF linearizes the system model around the current state estimate, which can introduce errors and , and requires the computation of Jacobean matrices for both the system and measurement models, which can be computationally intensive, especially if the system is highly nonlinear. In situations where the linearization is poor, the ETEKF may not provide accurate estimates. To overcome this issue, in this section we consider the ETUKF to approximate the event trigger posterior density [18]. In the ETUKF the nonlinear function approximated using the Unscented transform rule, where the Gaussian weighted integrals in (2.13) and (2.17) can be transformed into weighted sum of sigma point $\psi \in R^{n\times 2n}$ given as,

$$\mathbb{E}[\,g(x_k)] \approx \sum_{i=0}^{2n} W_m^i g(\psi^i). \tag{2.33}$$

2.4.2.1 Unscented transform

The unscented transform forms the Gaussian approximation with the following procedure,

- form a set of sigma point

$$\begin{aligned} \psi^0 &= \mu \\ \psi^i &= \mu + \sqrt{n+\lambda}[\sqrt{P}]_i \quad i = 1, ..., n, \\ \psi^{i+n} &= \mu - \sqrt{n+\lambda}[\sqrt{P}]_i \quad i = 1, ..., n, \end{aligned} \tag{2.34}$$

  where $[.]_i$ denotes the ith column of the matrix, and λ given as $\lambda = \alpha^2(n+\kappa) - n$

- After propagation with $\psi_i^{f(x)} = f(\psi_i)$, we compute the Estimates of the mean and covariance as follows:

$$\begin{aligned} \mathbb{E}[\,g(x_k)] = \hat{x} &\approx \sum_{i=0}^{2n} W_{(m)}^i \psi_i^{f(x)} \\ Cov[\,g(x_k)] = \hat{P} &\approx \sum_{i=0}^{2n} W_{(c)}^i (\psi_i^{f(x)} - \hat{x})(\psi_i^{f(x)} - \hat{x})^T, \end{aligned} \tag{2.35}$$

  where the constant weights W are given as,

$$\begin{aligned} W_{(m)}^0 &= \frac{\lambda}{n+\lambda}, \\ W_{(c)}^0 &= \frac{\lambda}{n+\lambda} + 1 - \aleph^2 + \beta, \\ W_{(m)}^i &= W_{(C)}^i = \frac{\lambda}{2(n+\lambda)} \quad i = 1, ..., n, \end{aligned} \tag{2.36}$$

  Where β denote parameter that can be used for incorporating prior information.

2.4.2.2 ETUKF filter

Then by implementing the Unscented transformation steps as in (2.14) to (2.16) for each Gaussian weighted integral in (2.13) and (2.17), all the integral weights are approximated. Thus, we can some up with an approximate filtering ETUKF which presented in the following step:

- Prediction
  Firstly, the following Sigma points are generated according to the state estimation at the last time instance and error covariance matrix.

$$\begin{aligned} \psi_{k-1}^0 &= \hat{x}_{k-1} \\ \psi_{k-1}^i &= \hat{x}_{k-1} + \sqrt{n+\lambda}[\sqrt{S_{k-1}}]_i \quad i = 1, ..., n, \\ \psi_{k-1}^{i+n} &= \hat{x}_{k-1} - \sqrt{n+\lambda}[\sqrt{S_{k-1}}]_i \quad i = 1, ..., n, \end{aligned} \tag{2.37}$$

where S_{k-1} computed by factorize $P_{k-1|k-1} = S_{k-1|k-1}S_{k-1|k-1}^T$. After that we propagate the sigma point through the system state function $\psi_{k|k-1}^{i,f} = f(\psi_{k-1}^i)$. Then, we compute the predicted state mean and covariance $\hat{x}_{k|k-1}$, $P_{k|k-1}$ as as follows,

$$\begin{aligned}\hat{x}_{k|k-1} &= \sum_{i=0}^{2n} W_{(m)}^0 \psi_{k|k-1}^{i,f} \\ \hat{P}_{k|k-1} &= \sum_{i=0}^{2n} W_{(C)}^0 (\psi_{k|k-1}^{i,f} - \hat{x}_{k|k-1})(\psi_{k|k-1}^{i,f} - \hat{x}_{k|k-1})^T + Q_k.\end{aligned} \tag{2.38}$$

- Time update

  we compute the new sigma point based on the predicted state,

$$\begin{aligned}\psi_{k|k-1}^0 &= \hat{x}_{k|k-1} \\ \psi_{k|k-1}^i &= \hat{x}_{k|k-1} + \sqrt{n+\lambda}[\sqrt{S_{k|k-1}}]_i \quad i = 1, ..., n, \\ \psi_{k-1}^{i+n} &= \hat{x}_{k|k-1} - \sqrt{n+\lambda}[\sqrt{S_{k|k-1}}]_i \quad i = 1, ..., n.\end{aligned} \tag{2.39}$$

where $S_{k|k-1}$ computed by factorize $P_{k|k-1} = S_{k|k-1}S_{k|k-1}^T$. After that we propagate the sigma point through the system measurement function $\psi_{k|k-1}^{i,g} = g(\psi_{k|k-1}^i)$. Then, we compute the predicted measurement mean and covariance,

$$\begin{aligned}\hat{y}_{k|k-1} &= \sum_{i=0}^{2n} W_{(m)}^0 \psi_{k|k-1}^{i,g} \\ \hat{P}_{k|k-1}^{yy} &= \sum_{i=0}^{2n} W_{(C)}^0 (\psi_{k|k-1}^{i,g} - \hat{y}_{k|k-1})(\psi_{k|k-1}^{i,g} - \hat{y}_{k|k-1})^T + R_k\end{aligned} \tag{2.40}$$

and the cross covariance,

$$\hat{P}_{k|k-1}^{xy} = \sum_{i=0}^{2n} W_{(C)}^0 (\psi_{k|k-1}^{i,f} - \hat{x}_{k|k-1})(\psi_{k|k-1}^{i,g} - \hat{y}_{k|k-1})^T \tag{2.41}$$

- When γ_k=1,
  - Update the filter using (2.27)
- When γ_k=0,
  - Update the filter based on the SOD using (2.21) and (2.22).
  - Update the filter based on the IBT using (2.24) and (2.25).

Algorithm 2.2, summarizes all the steps of the proposed event-triggered unscented Kalman filter.

Algorithme 2.2 : Event trigger Unscented Kalman filter (ETUKF)

[$\hat{x}_{k|k}, P_{k|k}$]:=ETUKF($\bar{y}_k, R_k, Q_k, \alpha, \xi$)

- **Prediction Step:**
  - define from $P_{k-1} = S_{k-1}S_{k-1}^T$

$$\begin{aligned} \psi_{k-1}^0 &= \hat{x}_{k-1} \\ \psi_{k-1}^i &= \hat{x}_{k-1} + \sqrt{n+\lambda}[\sqrt{S_{k-1}}]_i \quad i = 1, ..., n, \\ \psi_{k-1}^{i+n} &= \hat{x}_{k-1} - \sqrt{n+\lambda}[\sqrt{S_{k-1}}]_i \quad i = 1, ..., n, \end{aligned} \tag{2.42}$$

  - $\psi_{k|k-1}^{i,f} = f(\psi_{k-1}^i) \quad i = 0 : n$.
  - $\hat{x}_{k|k-1} = \sum_{i=0}^{2n} W_{(m)}^0 \psi_{k|k-1}^{i,f}$
  - $\hat{P}_{k|k-1} = \sum_{i=0}^{2n} W_{(C)}^0 (\psi_{k|k-1}^{i,f} - \hat{x}_{k|k-1})(\psi_{k|k-1}^{i,f} - \hat{x}_{k|k-1})^T + Q_k$
  - with $S_{k|k-1}$ for $P_{k|k-1} = S_{k|k-1}S_{k|k-1}^T$

$$\begin{aligned} \psi_{k|k-1}^0 &= \hat{x}_{k|k-1} \\ \psi_{k|k-1}^i &= \hat{x}_{k|k-1} + \sqrt{n+\lambda}[\sqrt{S_{k|k-1}}]_i \quad i = 1, ..., n, \\ \psi_{k-1}^{i+n} &= \hat{x}_{k|k-1} - \sqrt{n+\lambda}[\sqrt{S_{k|k-1}}]_i \quad i = 1, ..., n, \end{aligned} \tag{2.43}$$

  - $\psi_{k|k-1}^{i,g} = g(\psi_{k|k-1}^i)$
  - $\hat{y}_{k|k-1} = \sum_{i=0}^{2n} W_{(m)}^0 \psi_{k|k-1}^{i,g}$
  - $\hat{P}_{k|k-1}^{yy} = \sum_{i=0}^{2n} W_{(C)}^0 (\psi_{k|k-1}^{i,g} - \hat{y}_{k|k-1})(\psi_{k|k-1}^{i,g} - \hat{y}_{k|k-1})^T + R_k$
  - $\hat{P}_{k|k-1}^{xy} = \sum_{i=0}^{2n} W_{(C)}^0 (\psi_{k|k-1}^{i,f} - \hat{x}_{k|k-1})(\psi_{k|k-1}^{i,g} - \hat{y}_{k|k-1})^T$
- **Update Step:**
  - When γ_k=1,
    - Update the filter using (2.27)
  - When γ_k=0,
    - Update the filter based on the SOD using (2.21) and (2.22).
    - Update the filter based on the IBT using (2.24) and (2.25).

2.4.3 Spherical Cubature rule based design

The ETUKF based on Unscented transformation can approximate the event trigger posterior density and aims to communication burden by performing updates of the state estimate and covariance using a punch of sigma points. However, the ETUKF limited to such condition as the scaling parameter κ which defined as $\kappa = n-3$ that determines the spread of the sigma points around the mean of the state around the current state estimate.This scaling parameter can lead the covariance matrix to be not positive definite if the system model dimension increases ($n > 3$). In situations may not provide accurate estimates and lead to filtering divergence. To overcome this issue, in this section we consider the ETCKF to approximate the event trigger posterior density [43]. In the ETCKF the nonlinear function approximated using the spherical Cubature rule, where the Gaussian weighted integrals in (2.13) and (2.17) can be transformed into an expectation over the unit Gaussian distribution $N(\Psi; 0, I)$ where $I \in R^{n\times n}$ is the unit matrix and the sigma point $\psi \in R^{n\times 2n}$ given as,

$$\psi^i = \begin{cases} \sqrt{n}e_i, & i = 1, ..., n, \\ -\sqrt{n}e_{i-n}, & i = n+1, ..., 2n. \end{cases} \tag{2.44}$$

Where e_i denotes a unit vector. Using third-degree spherical Cubature integration the Gaussian weighted integrals $\int f(x)N(x; \bar{x}, \bar{P})dx$, can be approximated as following,

$$\int f(x)N(x; \bar{x}, \bar{P})dx \approx \frac{1}{2n}\sum_{i=1}^{2n} f(\bar{x} + \sqrt{\bar{P}}\psi^i). \tag{2.45}$$

Then by implementing the Spherical Cubature Integration in (2.44) and (2.45) for each Gaussian weighted integral in (2.13) and (2.17), all the integral weights are approximated and presented in the following step,

- Time update
  Initially, Sigma points are generated based on the state estimation from the previous time instance and the corresponding error covariance matrix,

$$\{\psi^x_{i,k-1|k-1} = S_{k-1|k-1}\psi^i + \hat{x}_{k-1|k-1}\}^n_{i=1} \tag{2.46}$$

  Where $S_{k-1|k-1}$ computed by factorize $P_{k-1|k-1} = S_{k-1|k-1}S^T_{k-1|k-1}$. After that we propagate the sigma point through the system state function,

$$\{\psi^{f(x)}_{i,k|k-1} = f(\psi^x_{i,k-1|k-1})\}^n_{i=1}. \tag{2.47}$$

  Then we compute the predicted state mean and covariance $\hat{x}_{k|k-1}$, $P_{k|k-1}$ as follows,

$$\begin{aligned} \hat{x}_{k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\psi^{f(x)}_{i,k|k-1} \\ \hat{P}_{k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}(\psi^{f(x)}_{i,k|k-1} - \hat{x}_{k|k-1})(\psi^{f(x)}_{i,k|k-1} - \hat{x}_{k|k-1})^T + Q_k. \end{aligned} \tag{2.48}$$

- Time update
  We compute the new sigma point based on the predicted state,

$$\{\psi^z_{i,k-1|k-1} = S_{k-1|k-1}\psi^i + \hat{x}_{k|k-1}\}^n_{i=1}. \tag{2.49}$$

  Where $S_{k|k-1}$ computed by factorize $P_{k|k-1} = S_{k|k-1}S^T_{k|k-1}$. After that we propagate the sigma point through the system measurement function,

$$\{\psi^{g(x)}_{i,k|k-1} = g(\psi^y_{i,k-1|k-1})\}^n_{i=1}. \tag{2.50}$$

Then, we compute the predicted measurement mean and covariance,

$$\begin{aligned} \hat{y}_{k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}\psi^{g(x)}_{i,k|k-1}, \\ \hat{P}^{yy}_{k|k-1} &= \frac{1}{2n}\sum_{i=1}^{2n}(\psi^{g(x)}_{i,k|k-1} - \hat{y}_{k|k-1})(\psi^{g(x)}_{i,k|k-1} - \hat{y}_{k|k-1})^T + R_k. \end{aligned} \tag{2.51}$$

And the cross covariance,

$$\hat{P}^{xy}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}(\psi^{f(x)}_{i,k|k-1} - \hat{x}_{k|k-1})(\psi^{g(x)}_{i,k|k-1} - \hat{y}_{k|k-1})^T. \tag{2.52}$$

- When γ_k=1,
  - Update the filter using (2.27),
- When γ_k=0,
  - Update the filter based on the SOD using (2.21) and (2.22),
  - Update the filter based on the IBT using (2.24) and (2.25).

Algorithm 2.3, summarizes all the steps of the proposed event-triggered Cubature Kalman filter.

Algorithme 2.3 : Event-triggered Cubature Kalman filter (ETCKF)

$[\hat{x}_{k|k}, P_{k|k}]$:=**ETCKF**$(y_k, R_k, Q_k, \alpha, \xi)$

- **initial Step:**
  - define $\psi^i = \begin{cases} \sqrt{n}e_i, & i = 1, ..., n, \\ -\sqrt{n}e_{i-n}, & i = n+1, ..., 2n, \end{cases}$
  - $P_{k-1|k-1} = S_{k-1|k-1}S^T_{k-1|k-1}$
  - Generate sigma points $\psi^x_{i,k-1|k-1} = S_{k-1|k-1}\psi^i + \hat{x}_{k-1|k-1} \quad i = 1 : 2n$.
- **Prediction Step:**
  - $\psi^{f(x)}_{i,k|k-1} = f(\psi^x_{i,k-1|k-1}) \quad i = 1 : 2n$.
  - $\hat{x}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\psi^{f(x)}_{i,k|k-1}$
  - $\hat{P}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}(\psi^{f(x)}_{i,k|k-1} - \hat{x}_{k|k-1})(\psi^{f(x)}_{i,k|k-1} - \hat{x}_{k|k-1})^T + Q_k$
  - $\psi^z_{i,k|k-1} = S_{k|k-1}\psi^i + \hat{x}_{k|k-1} \quad i = 1 : 2n$, with $S_{k|k-1}$ for $P_{k|k-1} = S_{k|k-1}S^T_{k|k-1}$
  - $\psi^{h(x)}_{i,k|k-1} = g(\psi^z_{i,k|k-1})$
  - $\hat{y}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}\psi^{g(x)}_{i,k|k-1}$
  - $\hat{P}^{yy}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}(\psi^{g(x)}_{i,k|k-1} - \hat{y}_{k|k-1})(\psi^{g(x)}_{i,k|k-1} - \hat{y}_{k|k-1})^T + R_k$
  - $\hat{P}^{xy}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n}(\psi^{f(x)}_{i,k|k-1} - \hat{x}_{k|k-1})(\psi^{g(x)}_{i,k|k-1} - \hat{y}_{k|k-1})^T$
- **Update Step:**
  - When γ_k=1,
    - Update the filter using (2.27)
  - When γ_k=0,
    - Update the filter based on the SOD using (2.21) and (2.22).
    - Update the filter based on the IBT using (2.24) and (2.25).

2.5 Comparison study of the proposed filter

The comparison among ETEKF, ETUKF, and ETCKF revolves around their utilization of event-triggering mechanisms to reduce computational cost while maintaining accurate state estimation in dynamic systems.

Event-triggering is a strategy where the filter updates are triggered only when specific conditions, such as measurement or prediction errors exceeding a threshold, are met. In the case of ETEKF, it employs the EKF for state estimation and updates the filter selectively based on predefined thresholds. While EKF is effective for moderately nonlinear systems, its reliance on linearization can lead to inaccuracies in highly nonlinear scenarios.

In contrast, ETUKF utilizes the unscented transform, which avoids linearization and provides more accurate estimates for highly nonlinear systems compared to ETEKF. The ETUKF also

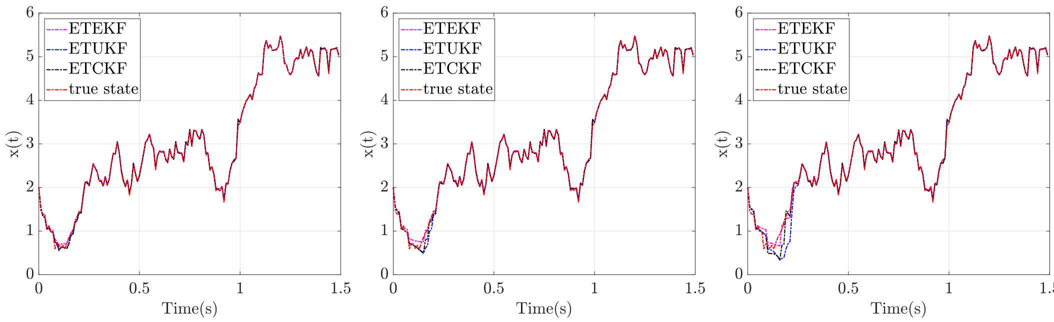


Figure 2.1: Tracking results for different communication rate (a) 90% (a) 80% (a) 60%.

incorporates event-triggering mechanisms to reduce computational burden, ensuring updates occur only when necessary. However, the UKF's use of the unscented transform for sigma point generation can be computationally intensive.

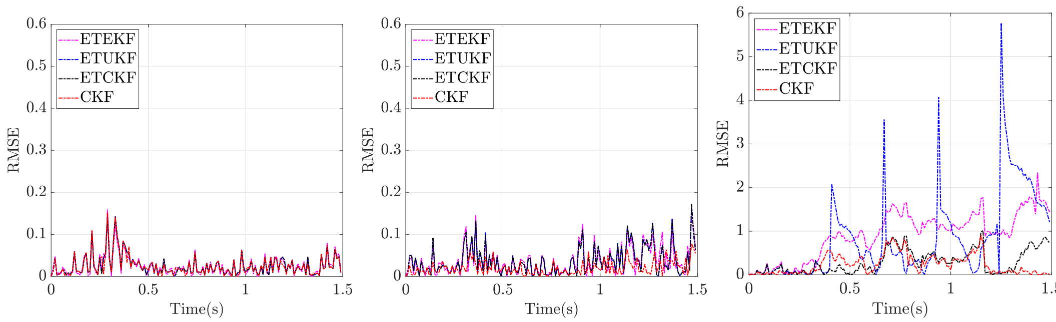
Similarly, The ETCKF employs the CKF, which, like UKF, avoids linearization by directly propagating a set of Cubature points through the nonlinear functions. This approach offers accuracy comparable to UKF while maintaining computational efficiency. The ETCKF, like the other filters, updates the filter based on event triggers to strike a balance between accuracy and computational cost.

In summary, the choice between ETEKF, ETUKF, and ETCKF depends on the specific characteristics of the dynamic system being modeled and the available computational resources. While ETUKF and ETCKF offer advantages over ETEKF in terms of accuracy for highly nonlinear systems, the ETCKF might be preferred when computational efficiency is critical. Each filter variant aims to strike a balance between accuracy and computational cost through the strategic use of event-triggering mechanisms.

2.6 Simulation results

In this simulation, the effectiveness of the discussed filters is tested by the nonlinear model as in (2.1) with $f(x_k) = 0.9995x_k - 0.0004x_k^2$ and measurement function $g(x_k) = x_k^2 + x_k^3$ the system noise is set as the Gaussian white noise, with system noise ω_k sampled from $\omega_k \sim N(x_k; 0, 0.01)$ and measurement noise sampled from $v_k \sim N(x_k; 0, 0.09)$. In order to study the estimation rsult for each discussed algorithm, The simulation of the motioned algorithms consists of two experiment , in the first part we run the simulation of three algorithms with SOD condition and we compare the state estimation for different communication rate. where the number of data transmission for the system without event-triggered mechanism, $\xi = 0$, is 150 which reduces significantly to 110, 84, and 68 for $\xi = 0.07$, $\xi = 0.7$ and $\xi > 1$, respectively. Figure 2.1 show the estimate state of the system for differ communication rate for the three filters, as a result we show that the three filter close to the true state.

In order to show the accuracy and efficiency of the proposed algorithms, the root mean square



(a) (b) (c)

Figure 2.2: RMSE for with different different communication rate (different thresholds) (a) 90% (a) 80% (a) 60%.

error (RMSE) is chosen as performance metric given as

$$RMSE_k = \sqrt{\frac{1}{runs}\sum_{i=1}^{runs}(x_k - \hat{x}^2_{k|k}}. \tag{2.53}$$

Where x_k and $\hat{x}_{k|k}$ are the true and state estimate computed by 2.1, 2.2 and 2.3 for 20 runs. The result of simulation are shown in figure 2.2 and table 2.1 .

Tableau 2.1: The average RMSE of different filters for various communication rate.

	RMSE			
Communication rate	CKF	ETEKF	ETUKF	ETCKF
90%	0.0130	0.0136	0.0134	0.0134
80%	0.0168	0.0236	0.0231	0.0231
60%	0.0267	0.0744	0.0738	0.0714

As illustrated in Table 2.1 and Fig.2.2, when the communication rate is reduced to 90% , all filters exhibit a consistent level of estimation accuracy, with RMSE values closely to the CKF at full communication rate. However, for reduced communication rate 80% the average of the proposed ETCKF is almost same as the ETUKF and perform better then ETEKF. As the communication rate decreases to 60%, the performance of the filtering techniques begins to deteriorate, leading to divergence in estimation results. This is evidenced by the increase in RMSE values across all filters, indicating a potential degradation in performance under lower communication efficiency. In general, according to the simulation analysis in Figs.2.1 , 2.2 and the result of table 2.1, it can be found that in the proposed the ETCKF has the robustness and more performance for reduced data and different change of the communication rate. It is inferred that the spherical–radial Cubature and unscented transformation rule in ETCKF and ETUKF can capture the posterior mean and covariance of the nonlinear state with a second order accuracy while the linearization in ETCKF can only achieve the first order.

2.7 Conclusion

This chapter introduces the approximation of the event trigger Bayesian problem under the assumption that the posterior distribution is Gaussian. First, we review the derivation of the Bayesian problem based on the event trigger condition SOD and IBT. Then, we observe that if the system is linear, the problem has a closed-form solution introduced by ETKF. For nonlinear systems, we approximate the integral weight using linearization and sum up with ETEKF. However, due to the limitation for high non-linearity, we review the Unscented transformation and conclude with ETUKF. Finally, we implement the ETCKF to tackle the problem of semi-definite using the spherical Cubature rule. According to the simulation experiment the proposed filters can guaranteed a good estimation accuracy for low communication and the ETCKF are good reference for comparison in our work.

Chapter 3

Event-Triggered Particle filter

3.1 Introduction

In Chapter 2, we saw that the posterior PDF was approximated based on a Gaussian assumption and reduced to event-triggered Kalman filters. However, in real time system and with an event-triggering approach, the Gaussian assumption collapsed and the Gaussian filter no longer be applicable. In this chapter [1], we present an event-triggered Particle Filter approach that designed for nonlinear and non-Gaussian systems, where the SOD mechanism is adopted to save the communication and power resources and the particle filter proposed to tackle the non Gaussian posterior.

The remainder of this chapter is as follows. In section 3.2, the problem formulation where the nonlinear system model is introduced with the SOD event-triggered. In section 3.3, the particle filter is discussed and the event trigger posterior approximated based particle filter. We illustrate our results using a simulated example in section 3.4.

3.2 Problem formulation

Recalling from chapter 1, the nonlinear system with intelligent sensor equipped with an event trigger strategy based on the SOD event trigger condition to save the power and communication resources. As shown in Fig.3.1 the non-linear system model is given as

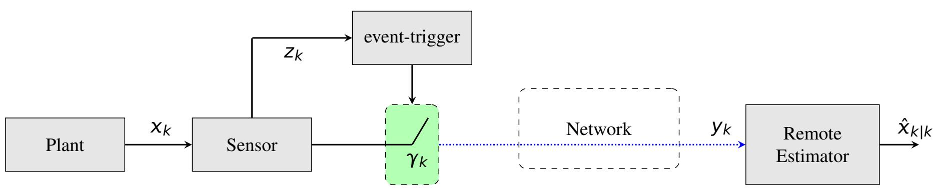


Figure 3.1: Black diagram of event-based state estimation.

$$x_{k+1} = f(x_k) + w_k, \tag{3.1}$$

$$z_k = g(x_k) + v_k. \tag{3.2}$$

where $f(.)$ and $g(.)$ are known nonlinear functions, x_k is the state vector, y_k is the sensor measurement, w_k and v_k are Gaussian noises sampled from the probability density functions p_μ and p_w, respectively. Based on the SOD trigger strategy decision variable γ_k, all the measurement at the remote estimator $y_{1:k} = y_{1:k-1} \cup \{y_k\}$, where y_k are given as,

$$y_k = \begin{cases} z_k, & \gamma_k = 1 \\ z_k \in \Xi_k, & \gamma_k = 0. \end{cases} \tag{3.3}$$

[1]The results of this chapter has been accepted for publication in the conference paper: E. Gasmi,M. A. Sid, O. Hachana, "Event-Triggered State Estimation Using Particle Filtering Approach," IEEE Conference on Advanced Electrical Engineering (ICAEE), 2022.

Due to the event trigger strategy, the posterior density no longer to be Gaussian. Thus, new approach based Particle filter are proposed to tackle the non-Gaussian problem.

3.3 Nonlinear event-trigger state estimation using particle filter

In this section, the nonlinear event trigger posterior are approximated based on the particle filter approach [32].

3.3.1 Particle filter

At time instant k, the particle filter computes the set of the particles,

$$\{x_k^{(i)}, \omega_k^{(i)} | i = 1, 2, 3, ..., M\} \tag{3.4}$$

Where the $x_k^{(i)}$ is the particle state and $\omega_k^{(i)}$ is the importance weights. The set of particles approximate the posterior distribution as,

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{M} \omega_k^{(i)} \delta(x_k - x_k^{(i)}) \tag{3.5}$$

The particle filter is designed based on an update rule for each particle by updating the particles states and weights from time $k-1$ to k,

$$\begin{aligned} x_{k-1}^{(i)} &\longrightarrow x_k^{(i)} \\ \omega_{k-1}^{(i)} &\longrightarrow \omega_k^{(i)}. \end{aligned} \tag{3.6}$$

Where the update rule used all the observed measurement $y_{1:k}$, based on the state and measurement model with their transition probability. After the computation of the posterior distribution in (3.5), the state mean and covariance can be approximated as follows,

$$\mathbb{E}[\, x_k|y_{1:k}] \approx \sum_{i=1}^{M} \omega_k^{(i)} x_k^{(i)} \tag{3.7}$$

To implement the particle filter we need to compute the probability density function of the all state $x_{1:k}$ given all the measurement and information $y_{1:k}$,

$$p(x_{1:k}|y_{1:k}) \tag{3.8}$$

Where the all state and measurement are given as,

$$\begin{aligned} x_{1:k} &= \{x_1, x_2, x_3, ..., x_k\} \\ y_{1:k} &= \{z_1, z_2, z_3, ..., z_k\} \end{aligned}$$

We note that computing the probability density function allow as to estimate all state $x_1, x_2, x_3, ..., x_k$, by knowing the mathematical form for all of them.

We consider the expectation of the all state involving the full state posterior density,

$$\mathbb{E}[x_{1:k}|y_{1:k}] = \int_{\mathbb{R}^n} x_{1:k} p(x_{1:k}|y_{1:k}) dx_{1:k} \tag{3.9}$$

The expectation above can be approximated based on the importance Sequential Importance Sampling Resampling (SIR). However, in the SIR we define new density as proposal density $q(x_{1:k}^{(i)}|z_{1:k})$ and the expectation in can be approximated as,

$$\mathbb{E}[x_{1:k}|y_{1:k}] \approx \sum_{i=1}^{M} \omega_k^{(i)} x_{1:k}^{(i)} \tag{3.10}$$

Where the particle state $x_{1:k}^{(i)}$ are sampled from the proposal density and the importance weights are given as,

$$\omega_k^{(i)} = \frac{p(x_{1:k}^{(i)}|y_{1:k})}{q(x_{1:k}^{(i)}|y_{1:k})} \tag{3.11}$$

The general form of the importance weights are fined as,

$$\omega_k = \frac{p(x_{1:k}|y_{1:k})}{q(x_{1:k}|y_{1:k})} \tag{3.12}$$

The goal of the particle filter rather then to directly derive the full posterior density we derive the recursive form of the general importance weights ω_k. In order to derive the importance weights first we write the full posterior density as follow,

$$p(x_{1:k}|z_k, y_{1:k-1}). \tag{3.13}$$

Then, by using bayes' rule, we have

$$p(x_{1:k}|z_k, y_{1:k-1}) = \frac{p(z_k|x_k)p(x_k|x_{k-1})p(x_{1:k-1}|y_{1:k-1})}{p(z_k|y_{1:k-1})}. \tag{3.14}$$

We derive the full proposal density $q(x_{1:k}|z_k, y_{1:k-1})$ as,

$$q(x_{1:k}|z_k, y_{1:k-1}) = q(x_k, x_{1:k-1}|y_{1:k}). \tag{3.15}$$

Then, by using bayes' rule, we have,

$$q(x_{1:k}|y_k, y_{1:k-1}) = q(x_k|x_{1:k-1}, y_{1:k})q(x_{1:k-1}|y_{1:k-1}). \tag{3.16}$$

Since the proposal density is not strictly related to the dynamical system, we assume,

$$q(x_k|x_{1:k-1}, y_{1:k}) = q(x_k|x_{1:k-1}). \tag{3.17}$$

As result we obtain

$$q(x_{1:k}|z_k, y_{1:k-1}) = q(x_k|x_{1:k-1})q(x_{1:k-1}|y_{1:k-1}). \tag{3.18}$$

Now we can substitute the equation (3.14) and (3.18) in the general imporatce weights equation in

(3.12) we obtain,

$$\begin{aligned}\omega_k =& C\frac{p(z_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{k-1})}\frac{p(x_{1:k-1}|y_{1:k-1})}{q(x_{1:k-1}|y_{1:k-1})},\\ =& C\frac{p(z_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{k-1})}w_{k-1},\\ \propto& \frac{p(z_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{k-1})}w_{k-1}.\end{aligned} \tag{3.19}$$

Where C is defined as $C = 1/p(z_k|y_{1:k-1})$ we note that the computation of the full importance weights depends only for the information in the state x_k, x_{k-1}, measurement z_k and the weights w_{k-1} at time $k-1$. Thus we can derive the weights for approximate the posterior density.

In the SIR filter we define the proposal density as $q(x_k|x_{k-1}) = p(x_k|x_{k-1})$, then the particle weights become as follows,

$$\omega_k \propto p(z_k|x_k)w_{k-1}. \tag{3.20}$$

This implies that for particle state $x_k^{(i)}$, the weight $\omega_k^{(i)}$,

$$\omega_k^{(i)} \propto p(z_k|x_k^{(i)})w_{k-1}^{(i)}. \tag{3.21}$$

We define $\tilde{w}_k^{(i)}$ as a normalized weights given as,

$$\tilde{w}_k^{(i)} = \frac{\omega_k^{(i)}}{\sum_{i=1}^{M}\omega_k^{(i)}}. \tag{3.22}$$

Consequently, the computed particle states and weights can approximate the posterior given in (3.5) with estimate state at iteration k and its associated covariance matrix $P_{k|k}$ as follows,

$$\hat{x}_{k|k} \approx \sum_{i=1}^{M}\tilde{\omega}_k^{(i)}x_k^{(i)}, \quad P_{k|k} \approx \sum_{i=1}^{M}\tilde{\omega}_k^{(i)}(x_k^{(i)} - \hat{x}_{k|k})(x_k^{(i)} - \hat{x}_{k|k})^T. \tag{3.23}$$

After several iteration of the particle filter algorithm we will have one or few particles with non negligible weights, this mean most of particle weights have very small value, effectively this means that the weights of this particles are equal to zero this so called degeneracy problem in particle filtering, since the time propagation o every particles consumes computational resources, we waste computational resources bu propagate the particles with very small weights. Thus , one of the effective approach to deal with this problem is re sample particles. In resampling we generate new set of particles from original set by selecting N set of particles with replacement and the probability of selecting the samples x_k^i with index i is proportional to its weights w_k^i.

The particle filter approach can be summarized in the fallowing steps,

- **Step 1:** according to the computed particle state $x_{k-1}^{(i)}$ or $i = 1, ..., M$ at the previous time $k-1$, we compute the particle state $x_k^{(i)}$ for $i = 1, ..., M$ based on the probabilty density function of the transition model $p(x_k|x_{k-1}^{(i)})$,

$$x_{k-1}^{(i)} \sim p(x_k|x_{k-1}^{(i)}). \tag{3.24}$$

- **Step 2:** based on the computed weights $w_{k-1}^{(i)}$ at time $k-1$ and the observed measurement y_k , we compute the update of the particles weights $\omega_k^{(i)}$ based on the probabilty density function of the measurement model,

$$\omega_k^{(i)} \propto p(y_k|x_k^{(i)})w_{k-1}^{(i)}. \tag{3.25}$$

- **Step 3:** we normalize the weights as,

$$\omega_k^{(i)} = \frac{\omega_k^{(i)}}{\sum_{i=1}^{M} \omega_k^{(i)}}. \tag{3.26}$$

- **Step 4:** we resample the article if the weights satisfied the following condition,

$$N_{eff} = \frac{1}{\sum_{i=1}^{M} (\omega_k^{(i)})^2}. \tag{3.27}$$

then if $N_{eff} < M/3$ we re sample new set of particles,

$$\{x_k^{(i)}, \omega_k^{(i)} = \frac{1}{M}\} = Resample(x_k^{(i)}, \omega_k^{(i)}) \quad i = 1, ..., M. \tag{3.28}$$

3.3.2 Event-trigger particle filter

In this section, we apply the particle filtering approach in section 3.3.1 to approximate the event trigger posterior $p(x_k|y_{1:k})$. However, the particle filter approximate the posterior distribution based on the computation of particles state and weights in the presence of an event-triggered mechanism.

The event trigger particle filtering approach implements the filtering recursions by propagating the particles to approximate the prior distribution then use the measurement and trigger information to update that weights and approximate the posterior pdf. Let the already computed particles state $x_{k-1}^{(i)}$ and weights , $w_{k-1}^{(i)}$ for $i = 1 : M$. Therefore, the posterior $p(x_{k-1}|y_{1:k-1})$ at time $k-1$ can be approximated as,

$$p(x_{k-1}|\mathbb{Y}_{k-1}) \approx \sum_{i=1}^{M} w_{k-1}^{(i)} \delta(x_{k-1} - x_{k-1}^{(i)}). \tag{3.29}$$

From (3.19), the importance weights satisfy,

$$\begin{aligned} \omega_k^{(i)} &= \frac{p(x_{1:k}^{(i)}|y_{1:k})}{q(x_{1:k}^{(i)}|y_{1:k})}, \\ &= \frac{p(x_k|x_{1:k-1}, y_{1:k-1})}{q(x_k|x_{1:k-1}, y_{1:k-1})} \frac{p(x_{1:k-1}^{(i)}|y_{1:k-1})}{q(x_{1:k-1}^{(i)}|y_{1:k-})}, \\ &= \frac{p(x_k|x_{1:k-1}, y_{1:k-1})}{q(x_k|x_{1:k-1}, y_{1:k-1})} w_{k-1}^{(i)}. \end{aligned} \tag{3.30}$$

Then, we generate the state particles $x_k^{(i)}$from the proposal density $q(x_k|x_{k-1}, y_{1:k-1})$, the prior distribution can be approximated as,

$$p(x_k|\mathbb{Y}_{k-1}) \approx \sum_{i=1}^{M} w_{k-1}^{(i)} \delta(x_k - x_k^{(i)}). \tag{3.31}$$

Where the predicted particle $x_k^{(i)}$ is computed by propagate the particles $x_{k-1}^{(i)}$ through the equation model in (3.1) with sampled particle noise $\{\mu_{k-1}^{(i)}\}_{i=1}^{M}$ from p_μ. The predicted particle is computed as,

$$x_k^{(i)} = f(x_{k-1}^{(i)}) + w_k^{(i)}. \tag{3.32}$$

The posterior distribution $P(x_k|y_{1:k})$ can be approximated as $p(x_k|y_{1:k}) \approx \sum_{i=1}^{M} \omega_k^{(i)} \delta(x_k - x_k^{(i)})$ with particle state $x_k^{(i)}$ computed by (3.32) and weights $\omega_k^{(i)}$ presented in the following theorem.

Theorem 1 *Consider the system* (3.1) *along with the triggered measurement in* (3.3)*. The particle filter weights are given by:*

$$\omega_k^{(i)} = \omega_{k-1}^{(i)}[\gamma_k p(z_k|x_k^{(i)}) + (1-\gamma_k) m_*(x_k^{(i)}, w_k^{(i)}, v_k^{*(i)})]. \tag{3.33}$$

In this cases the $m_*(.)$ *are the judgment variable given as,*

$$m_*(x_k^{(i)}, w_k^{(i)}, v_k^i) = \begin{cases} 1, & y_k^{(i)} \in \Xi_k \\ 0, & y_k^{(i)} \notin \Xi_k \end{cases}. \tag{3.34}$$

Proof 1 *To compute the particle weights, we follow the same derivation in particle filter steps as in* (3.12) *with posterior pdf is similar to the derivation in* (1.15) *and* (1.16) *where two cases are considered: the measurement* y_k *is sent to the estimator (*$\gamma_k = 1$*) and otherwise (*$\gamma_k = 0$*).*

We consider the following two cases (see section 1.5.2.2).

- ***Case 1:*** *If* $\gamma_k = 1$ *the measurement is transmitted to the remote estimator. Thus, the particle filter uses the particle state generated from the proposal density* $q(x_k|x_{k-1})$ *to compute particle state* $x_k^{(i)}$ *and the particle weights is computed based on the likelihood pdf which reduced to sensor likelihood function* $p(z_k|x_k^{(i)})$*. Consequently, using* (3.25) *the importance weights can be derived as,*

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} p(z_k|x_k^{(i)}). \tag{3.35}$$

- ***Case 2:*** *if* $\gamma_k = 0$ *the measurement* z_k *is non triggered. Therefore, using the trigger information(*Ξ_k*), the likelihood function can be specified as* $p(z_k \in \Xi|x_k^{(i)})$*. Thus, the importance weights can be updated as follows,*

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} p(z_k \in \Xi|x_k^{(i)}). \tag{3.36}$$

The computation of $p(z_k \in \Xi|x_k^{(i)})$ *involves solving the following integral,*

$$p(z_k \in \Xi|x_k^{(i)}) = \int_{\Xi} p(z_k|x_k^{(i)}) dz_k. \tag{3.37}$$

We note that in (3.37)*, the specified likelihood function is computed based on the integration over the set* Ξ *for each particle state* $x_k^{(i)}$*. Which is hard for computation in real time system and lead to computation burden. For numerical method the computation for each particle is too heavy to sustain and one feasible method is to introduce the Bayesian constrained method in [37], where the posterior distribution is assumed as constraint posterior Problem.*

Lemma 2 *If the system satisfies the constraint,*

$$\phi(x_k) \in \Omega_k. \tag{3.38}$$

The posterior of constraint system at time $k-1$ is approximated as ,

$$p(x_{k-1}|y_{1:k-1}) \approx \sum_{i=1}^{M} w_{k-1}^{(i)} \delta(x_{k-1} - x_{k-1}^{(i)}). \tag{3.39}$$

The importance weights can be updated as,

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} L(x_k^{(i)}) p(z_k \mid x_k^{(i)}). \tag{3.40}$$

where $x_k^{(i)}$ generated from the prior distribution $p(x_k|x_{k-1}^{(i)})$ and $L(x_k^{(i)})$ is judgment variable satisfies,

$$L(x_k^{(i)}) = \begin{cases} 1, & \phi^*(x_k^{(i)}) \in \Omega, \\ 0, & \phi^*(x_k^{(i)}) \notin \Omega. \end{cases} \tag{3.41}$$

From 2, we define $\phi^*(x_{k-1}, w_k, v_k)$ be the system constraint where,

$$\phi(x_k) = \phi(f(x_{k-1}) + w_k) = \phi^*(x_{k-1}, w_k, v_k). \tag{3.42}$$

Let the particles noises $w_k^{(i)}$, $v_k^{(i)}$, and the predicted state $x_k^{(i)} = f(x_{k-1}^{(i)}) + w_k^{(i)}$. From 2 and following [46], $L(x_k^{(i)}, w_k^{(i)}, v_k^{(i)})$ is the decision variable,

$$L(x_k^{(i)}, u_k^{(i)}, v_k^{(i)}) = \begin{cases} 1, & \phi^*(x_{k-1}, w_k, v_k) \in \Omega \\ 0, & \phi^*(x_{k-1}, w_k, v_k) \notin \Omega \end{cases}. \tag{3.43}$$

Thus, the importance weight obtained as,

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} L(x_k^{(i)}, w_k^{(i)}, v_k^{(i)}) p(y_k|x_k^{(i)}). \tag{3.44}$$

Let z_k is taken as the measurement of the system state constraint given as

$$\phi_1^*(x_{k-1}, w_k, v_k) = h(f(x_{k-1}) + w_k) + v_k = z_k. \tag{3.45}$$

Due to the event trigger measurement the posterior distribution $p(x_k|y_{1:k})$ can be assumed as constrained Bayesian state estimation problem. Thus, following lemma 2, we define non triggered measurement z_k as the state constraint $\phi_1^*(x_k, w_k, v_k)$ where,

$$\phi_1^*(x_k, w_k, v_k) = h(f(x_{k-1}) + w_k) + v_k = z_k. \tag{3.46}$$

For each particle,

$$\phi_1^*(x_{k-1}^{(i)}, w_k^{(i)}, v_k^{(i)}) = h(f(x_{k-1}^{(i)}) + w_k^{(i)}) + v_k^{(i)} = z_k^{(i)}. \tag{3.47}$$

With system state particles $x_k^{(i)}$ and the observation particle $z_k^{(i)}$ can be obtained as,

$$x_k^{(i)} = f(x_{k-1}^{(i)}) + w_k^{(i)}, \tag{3.48}$$
$$z_k^{(i)} = h(x_k^{(i)}) + v_k^{(i)}. \tag{3.49}$$

Then, we define the trigger condition Ξ_k as the constraint set Ω_k. Consequently, the corresponding judgment variable in (3.43) can be rewritten as,

$$L(z_k^{(i)}) = m_*(x_k^{(i)}, w_k^{(i)}, v_k^{(i)}) = \begin{cases} 1, & z_k^{(i)} \in \Xi_k \\ 0, & z_k^{(i)} \notin \Xi_k \end{cases}. \tag{3.50}$$

Finally, the constraint importance weights become,

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} m_*(x_k^{(i)}, w_k^{(i)}, v_k^{(i)}) = \begin{cases} \omega_{k-1}^{(i)}, & z_k^{(i)} \in \Xi_k \\ 0, & z_k^{(i)} \notin \Xi_k \end{cases} \tag{3.51}$$

The Bayesian constraint step are clearly discussed in figure 3.2.

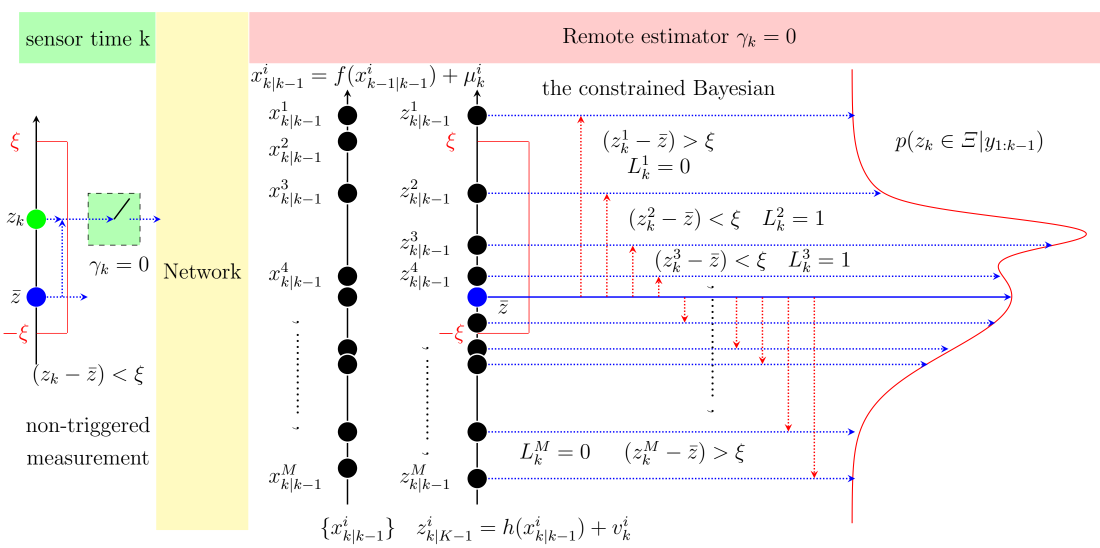


Figure 3.2: Black diagram of event-trigger Bayesian constraint approximation.

After computing (3.35) *and* (3.51), *we obtain ,*

$$\omega_k^{(i)} = \omega_{k-1}^{(i)}[\gamma_k p(z_k|x_k^{(i)}) + (1-\gamma_k) m_*(x_k^{(i)}, w_k^{(i)}, v_k^{*(i)})]. \tag{3.52}$$

The proof is completed.

We note that to approximate the posterior distribution under the non-Gaussian assumption we follow the same steps for particle derivation with modified weights which depend on the computation of the likelihood cases based in the measurement received and the information of the trigger mechanism. The pseudo-code of event trigger particle filtering approach is provided in the algorithm 3.1.

Algorithme 3.1 : Event trigger Particle Filter (ETPF)

$[\hat{x}_k, \hat{P}_k, \{x_k^{(i)}, \omega_k^{(i)}\}_{i=1}^M]$**:=ETPF(**$y_k, \{x_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^M, R_k, Q_k, \xi$)

- **initial Step:**
  - Draw $x_0^{(i)} \sim p_{x_0}(x_0|\bar{y}_0)$, for $i = 1 : M$,
- **Prediction Step:**
  - Compute the $w_k^{(i)} \sim p_\mu(w_k)$, for $i = 1 : M$
  - Compute , $x_k^{(i)} = f(x_{k-1}^{(i)}) + w_k^{(i)}$, for $i = 1 : M$,
  - Compute the predicted state estimate
    - $\hat{x}_{k|k-1} \approx \sum_{i=1}^M \tilde{\omega}_{k-1}^{(i)} x_k^{(i)}$,
- **Update Step:**
  - Compute the importance weight $\tilde{\omega}_k^{(i)}$ in (3.33) and (3.34) , for $i = 1 : M$,
  - Normalize the importance weight: $\omega_k^{(i)}$ in (3.26) , for $i = 1 : M$,
- **Estimate Step:** Estimate the posterior (MMSE) state and covariance
  - $\hat{x}_k \approx \sum_{i=1}^M \omega_k^{(i)} x_k^{(i)}$,
  - $P_k \approx \sum_{i=1}^M \omega_k^{(i)} (x_k^{(i)} - \hat{x}_{k|k})(.)^T$,
- **Resampling Step:** if $N_{eff} < M/3$ we re sample new set of particles
  - $[\{x_k^{(i)}, \omega_k^{(i)}\}_{i=1}^M]$**:=RESAMPLE(**$\{x_k^{(i)}, \omega_k^{(i)}\}_{i=1}^M$),

---

3.4 Simulation results

In this section, simulation experiments are developed to evaluate the performance of the proposed Algorithm 3.1. The two link robot arm example are proposed as nonlinear system as presented in figure 3.3 , where the joint angles $\theta_k = [\theta_{k,1}\theta_{k,2}]^T$ is the state vector and the measured vector is the end effector of the robot arm. The state space model of the nonlinear system given as,

$$\begin{aligned} \theta_{k+1} &= \theta_k + \mu_k \\ Y_k &= (\begin{bmatrix} l_1 cos(\theta_{1,k}) + l_2 cos(\theta_{1,k} + \theta_{2,k}) \\ l_1 sin(\theta_{1,k}) + l_2 sin(\theta_{1,k} + \theta_{2,k}) \end{bmatrix} + v_k \end{aligned} \tag{3.53}$$

The parameter $l_1 = 1$, $l_2 = 2$ are the length of the links, the state and measurement noises are given as Gaussian with $\mu_k \sim N(0, [0.01, 0.1])$ and $v_k \sim N(0, 0.005I)$, respectively. We define the number of particles as $M = 1000$ and we simulate the system for different event-triggered threshold $\xi = 0.01, 0.05, 0.1$ for different communication rate 63%, 18% and less than 18%. The simulation results in figure 3.4 and 3.5 shows the estimated result of the position of the end effector (x,y), we can see that the both estimate states are clearly close and follow the actual state for

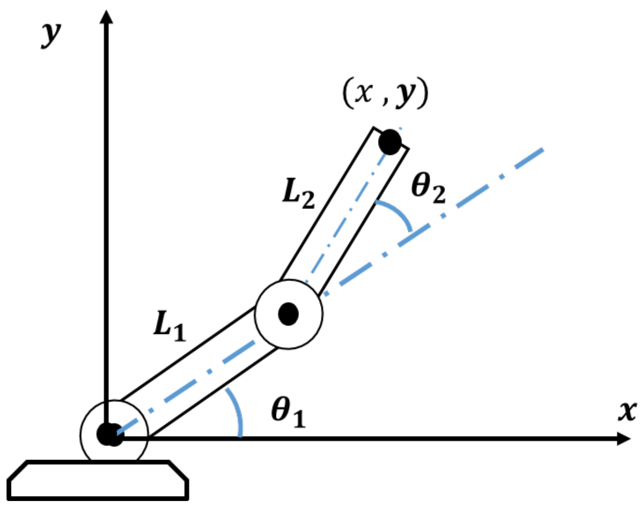


Figure 3.3: The two link robot arm system.

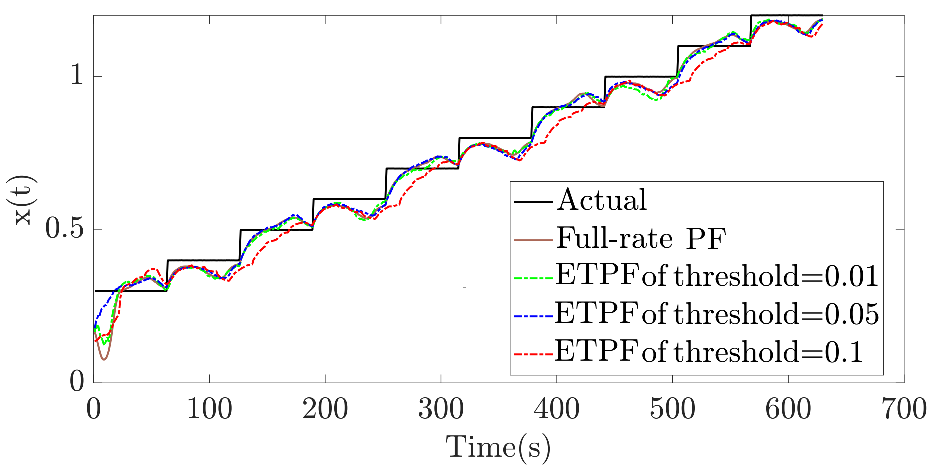


Figure 3.4: Tracking results of the position x of a system with the event-triggered.

reduced communication rate with threshold $\xi = 0.01, 0.05$. however when event-triggered threshold increased more then $\xi = 0.1$ the quality of the tracking start to deteriorates due to the lack of data transmission.

To compare the filter performance we compute the RMSE of the state for 20 runs. The simulation result are given in figure 3.6 under different communication rate, it can be shown that the estimation quality are very close to the full rate particle filter which improve the performance of the event trigger particle filter algorithm. However, the increase the event trigger threshold the estimation quality would be degraded. By the further comparison from Fig. 3.7 the performance of the ETPF is more accurate than that of the ETCKF due to the benefits of using the particle filter approximation.

3.5 Conclusion

This chapter introduces the approximation of the event trigger Bayesian problem under the assumption that the posterior distribution is Non-Gaussian. First, we review the derivation of the particle

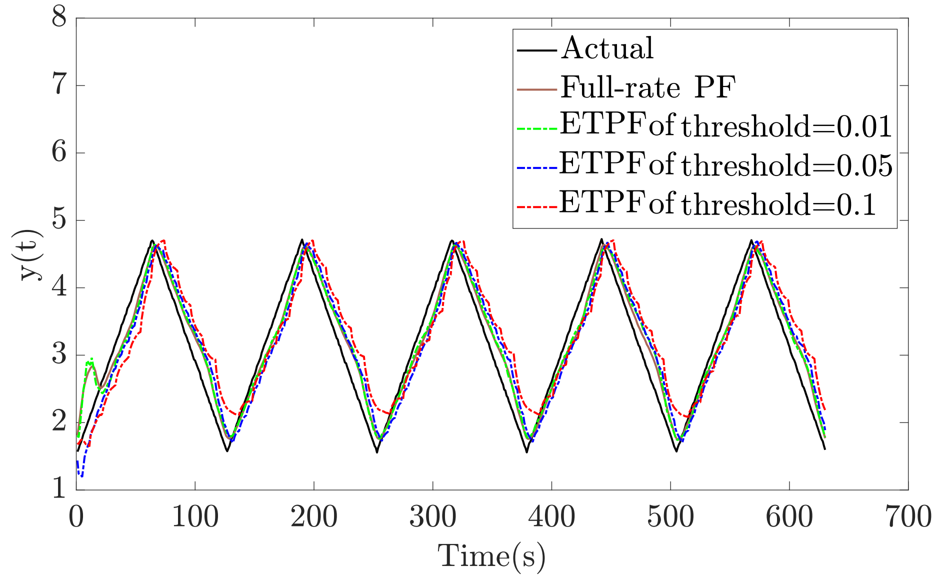


Figure 3.5: Tracking results of the position y a system with the event-triggered.

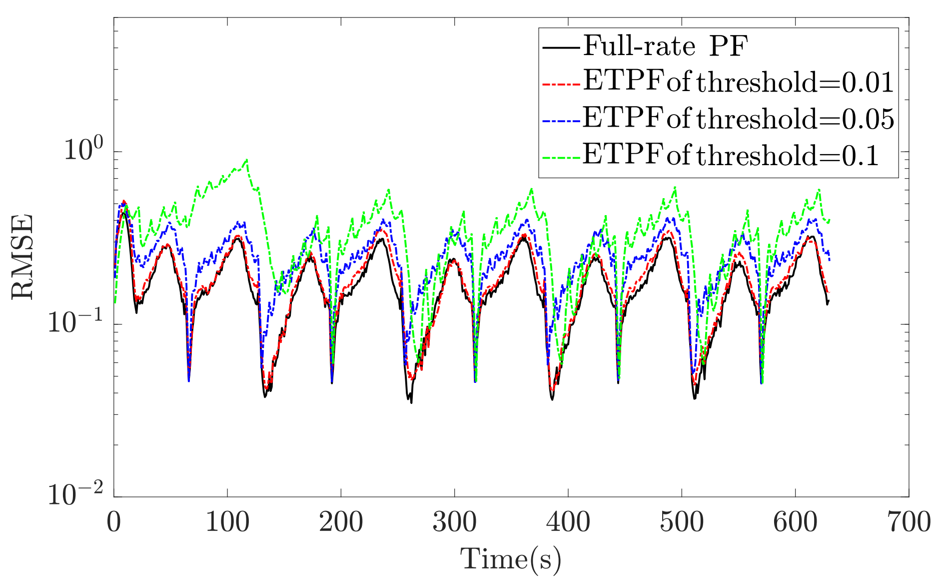


Figure 3.6: RMSE results of ETPF for the different event-triggered.

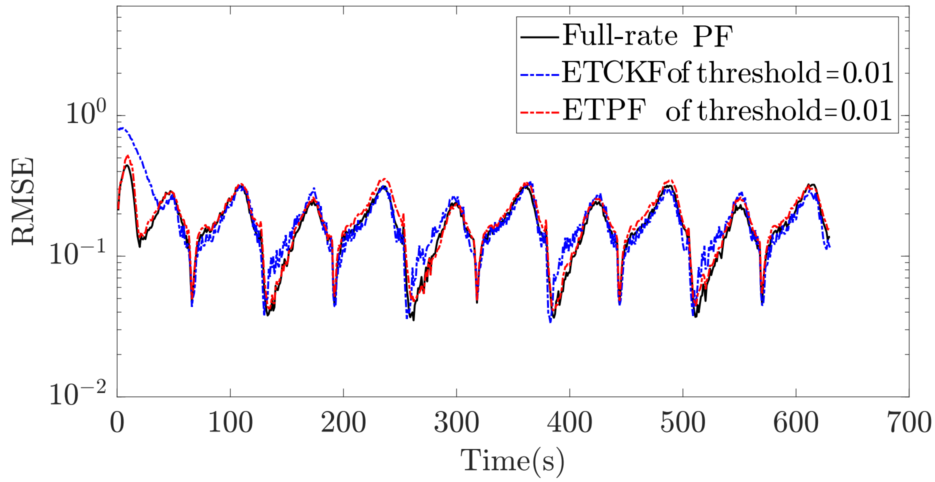


Figure 3.7: RMSE results of ETPF compared with ETCKF the event-triggered.

filter approach to approximate the Bayesian problem based on particles and weights. Then, we reformulate the likelihood unction based on the event trigger condition SOD and we derived the event trigger particle filter algorithm where the weights is approximated based on Bayesian constraint to reduce the computation burden. Finally, the implemented ETPF Algorithm validated through simulation experiment and compared with the Gaussian ETCKF where ETPF provide more accuracy result to the Gaussian filter.

Chapter 4

Event-Trigger Particle filter under packet loss

4.1 Introduction

In this chapter[1], as extension to the developed algorithm in chapter 3, we present a novel event-triggered Particle Filter algorithm designed for nonlinear dynamic systems where packet loss occurs during data transmission in communication channels. The packet loss is modeled as Bernoulli distribution random variables. The Particle Filter, utilizing the Monte Carlo method, approximates nonlinear and non-Gaussian probability density functions (pdfs) and SOD condition mechanism to save the communication resources.

Demonstrating the effectiveness of the event-triggered strategy, we reveal a reduction in data transmission between sensors and the state estimator. By carefully selecting an event-triggered threshold and tuning both the event-triggered mechanism and packet loss rate, our approach minimizes data transmission while ensuring filtering accuracy. Furthermore, we utilize a specialized MCS algorithm to derive the approximated posterior with corresponding RMSE. The performance of the estimator is assessed by comparing the error covariance of Posterior Cramér-Rao Lower Bound (PCRLB).

The remainder of this chapter is as follows. In Section 4.2, the problem formulation where the nonlinear system model is introduced with the SOD event-triggered and the packet loss. In Section 4.3, the designed event-trigger particle filter under packet loss algorithm. In Section 4.4, Performance analysis of the ETPF under packet loss based on the PCRLB. We illustrate our results using a simulated example in in Section 4.5.

4.2 Problem formulation

Recalling from Chapter 1, the nonlinear system model,

$$x_{k+1} = f(x_k) + w_k \tag{4.1}$$

$$z_k = g(x_k) + v_k \tag{4.2}$$

As shown in Fig.4.1, to save the communication resources the sensor equipped with an event trigger scheduler γ_k, where the triggered measurement y_k given as

$$y_k = \begin{cases} z_k, & \gamma_k = 1 \\ z_k \in \Xi_k, & \gamma_k = 0 \end{cases} \tag{4.3}$$

with Ξ_k obey the SOD condition. The triggered measurement transmitted via a network which is unreliable where the packet loss can be occurs.

4.2.1 Packet loss

The unreliable communication channel is modeled by the random variable of packet loss λ_k. When $\lambda_k = 1$, the measurement is received by the remote estimator; otherwise ($\lambda_k = 0$), the measurement is lost. Following conventions in [42, 43], we define $P(v_k|\lambda_k)$, the noise distribution which depends

[1]The results of this chapter have been accepted for publication in the article: E. Gasmi, M. A. Sid, O. Hachana, "Nonlinear event-based state estimation using particle filter under packet loss," ISA Transactions October 2023.

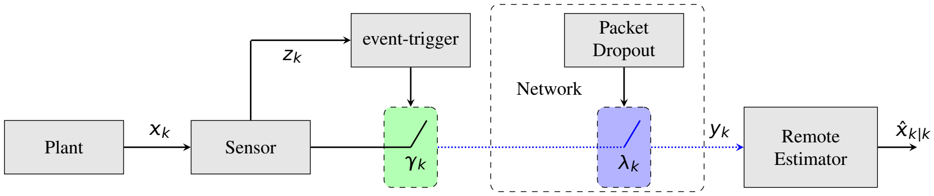


Figure 4.1: Black diagram of event-based state estimation with packet loss.

on λ_k, as follows,

$$P(v_k|\lambda_k) = \begin{cases} N(0, R_k), & \lambda_k = 1 \\ N(0, \sigma^2 \mathbb{I}_m), & \lambda_k = 0 \end{cases}. \tag{4.4}$$

where $\sigma \to \infty$ and R_k is the covariance following the system dimensions.

Remark 1 *The remote estimator is aware of γ_k and λ_k. Thus, the following cases occur at the remote estimator: the measurement is received ($\gamma_k = 1$, $\lambda_k = 1$); the measurement is not received in two cases: when it is non-triggered ($\gamma_k = 0$) or when triggered but there is packet loss ($\gamma_k = 1$, $\lambda_k = 0$).*

Based on the event trigger scheduler and packet loss, we define all the received measurement at the remote estimator as $y_{1:k} \triangleq \{\{\gamma_0\ ,..,\gamma_k\}, \{\lambda_0, .., \lambda_k\}, z_0, .., z_k\}$. The objective in this chapter to approximate the posterior distribution x_k given all the measurement $y_{1:k}$ under the event trigger and packet loss ($p(x_k|y_{1:k})$), the we compute the expected state $\hat{x}_{k|k}$ as,

$$\hat{x}_{k|k} \triangleq \mathbb{E}\left[x_k|y_{1:k}\right] \tag{4.5}$$

and covariance $P_{k|k}$ as,

$$P_{k|k} \triangleq \mathbb{E}\left[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T|y_{1:k}\right]. \tag{4.6}$$

4.3 Event-trigger particle filter under packet loss

In this section, we derive the nonlinear filtering problem with the event-triggered mechanism under packet loss. The approximation based particle filtering method is implemented to form the posterior distribution based on particle state and importance weights in the following discussion.

We assume in our study that the remote estimator have the well knowledge λ_k and γ_k, while the packet dropout occurs only if the measurement z_k is transmitted from sensor to the remote estimator. According to this assumption, if the measurement is transmitted and no packet drops the derivation of the importance weights follow the same derivation of event trigger particle filter, otherwise the posterior approximated based the information depends on the information of the trigger and packet loss. Recalling from Chapter 3, the particle filter derivation in section 3.3.1 and the importance weights in theorem 1. We define the set $\bar{\Xi}$ as all the trigger measurement z_k where $z_k \notin \Xi_k$. Thus, similar to the derivation of the posterior distribution in Chapter 3, three cases need to be considered, the measurement is triggered, a packet loss occurs ($\lambda_k = 0$) and measurement is triggered and received at the remote estimator ($\gamma_k = 1$, $\lambda_k = 1$) or not triggered ($\gamma_k = 0$).

(a) if $\gamma_k = 1$ and $\lambda_k = 1$, measurement z_k is transmitted and received. The observation in the remote estimator $y_{1:k}$ is updated to $y_{1:k} = y_{1:k-1} \cup \{z_k\}$. We apply Bayesian the theorem, the posterior pdf become

$$\begin{aligned} p(x_k|y_{1:k}) &= p(x_k|y_{1:k-1}, y_k), \\ &= \frac{p(z_k|x_k)p(x_k|y_{1:k-1})}{p(z_k|y_{1:k-1})}. \end{aligned} \tag{4.7}$$

(b) if $\gamma_k = 1$ and $\lambda_k = 0$ the measurement z_k is triggered and packet loss occurs, we update the posterior using the new set z_k satisfy the condition $\bar{\Xi}_k$ where the measurement triggered and lost during the transmission $\{z_k \in \bar{\Xi}_k\}$. Consequently,

$$\begin{aligned} p(x_k|y_{1:k}) &= p(x_k|y_{1:k-1}, \{z_k \in \bar{\Xi}_k\}, \lambda_k = 0), \\ &= \frac{p(\{z_k \in \bar{\Xi}_k\}|\lambda_k = 0, x_k)p(x_k|y_{1:k-1})}{p(\{z_k \in \Xi_k\}|y_{1:k-1})}. \end{aligned} \tag{4.8}$$

where

$$p(z_k \in \bar{\Xi}_k|\lambda_k = 0, x_k) = \int_{\bar{\Xi}_k} p(z_k|\lambda_k = 0, x_k)dy_k. \tag{4.9}$$

(c) if $\gamma_k = 0$ the posterior updated based on the event trigger set $\{z_k \in \Xi_k\}$. Consequently,

$$\begin{aligned} p(x_k|y_{1:k}) &= p(x_k|y_{1:k-1}, \{z_k \in \Xi_k\}), \\ &= \frac{p(\{z_k \in \Xi_k\}|x_k)p(x_k|y_{1:k-1})}{p(\{z_k \in \Xi_k\}|y_{1:k-1})}. \end{aligned} \tag{4.10}$$

where,

$$p(z_k \in \Xi_k(\bar{z})|x_k) = \int_{\Xi_k} p(y_k|x_k)dy_k. \tag{4.11}$$

The posterior distribution $p(x_k|y_{1:k})$ with event trigger and packet loss can be approximated as,

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{M} w_k^{(i)} \delta(x_k - x_k^{(i)}). \tag{4.12}$$

Where the particle state $x_k^{(i)}$ are generated from prior distribution $p(x_k|y_{1:k-1})$ and the importance weights are given in the following theorem,

Theorem 2 *Consider the system* (4.1) *along with the current transmitted measurement (measurement transmitted under the event-triggered mechanism)* -(4.3) *and the packet dropout parameter* λ_k. *Under this conditions, the particle filter weights are given by:*

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} \begin{cases} \lambda_k p(y_k|x_k^{(i)}) + (1 - \lambda_k)\bar{L}_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{*(i)}) & \gamma_k = 1 \\ m_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{(i)}) & \gamma_k = 0 \end{cases} \tag{4.13}$$

In this cases the $m_*(.)$, $\bar{L}_*(.)$ are the judgment variable given as,

$$m_*(x_k^{(i)}, \mu_k^{(i)}, v_k^i) = \begin{cases} 1, & y_k^{(i)} \in \Xi_k \\ 0, & y_k^{(i)} \notin \Xi_k \end{cases}, \quad \bar{L}_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{*i}) = \begin{cases} 1, & y_k^{*(i)} \in \bar{\Xi}_k \\ 0, & y_k^{*(i)} \notin \bar{\Xi}_k \end{cases}. \tag{4.14}$$

Proof 2 *The derivation of the importance weight depends on the event trigger cases and packet loss variable λ_k.*

- ***Case 1:*** *If $\lambda_k = 1, \gamma_k$ no packet loss occurs, the observation $y_{1:k}$ is $y_{1:k-1}$ with the triggered or non-trigger measurement z_k. Thus, the posterior derivation obey the event trigger particle filter. Consequently, using* (3.25) *the importance weights can be derived as,*

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} \begin{cases} p(y_k|x_k^{(i)}) & \gamma_k = 1 \\ m_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{(i)}) & \gamma_k = 0 \end{cases}. \tag{4.15}$$

where $L_*(.)$ is the judgment variable given as

$$m_*(x_k^{(i)}, \mu_k^{(i)}, v_k^i) = \begin{cases} 1, & y_k^{(i)} \in \Xi_k \\ 0, & y_k^{(i)} \notin \Xi_k \end{cases}. \tag{4.16}$$

- ***Case 2:*** *if $\lambda_k = 0$ and $\gamma_k = 1$: the measurement z_k is triggered and packet loss occurs. The importance weights computed on the non trigger set with packet loss $\bar{\Xi}_k$, where the distribution of noise follows $v_k \sim N(0, \sigma^2 \mathbb{I}_m)$, by substituting (4.10) in (3.30) the importance weight is given by*

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} p(y_k \in \bar{\Xi}|\lambda_k = 0, x_k^{(i)}). \tag{4.17}$$

The computation of the probability integral in equation (4.17) presents a significant challenge, as discussed in chapter 3. Similar to section 3.3.2 we define $\bar{\phi}_1^*(x_{k-1}, u_k, v_k)$ is a the state constraint for triggered measurement with packet loss. However, the new constraint computed based on the system noise v_k under packet loss is given as follows,

$$v_k^* \sim p(v_k|\lambda_k = 0) = N(0, \sigma^2 \mathbb{I}_m). \tag{4.18}$$

For each particle in equation (4.18), the particle noise $v_k^{*(i)}$ are generating corresponding to packet loss $\lambda_k = 0$. Consequently,

$$\bar{\phi}_1^*(x_{k-1}^{(i)}, \mu_k^{(i)}, v_k^{*(i)}) = y_k^{*(i)} = h(x_k^{(i)}) + v_k^{*(i)}. \tag{4.19}$$

Remark 2 *Note that the state constrained $\bar{\phi}_1^*(x_{k-1}^{(i)}, \mu_k^{(i)}, v_k^{*(i)})$ is computed based on the particle measurement y_k^i and the particle noise $v_k^{(i)}$ given $\lambda_k = 0$.*

Using lemma 2, y_k^i is the state constraint $\bar{\phi}_1^*(x_{k-1}^{(i)}, \mu_k^{(i)}, v_k^{*(i)})$ and ω_k is trigger set $\bar{\Xi}$, Thus, *the judgment variable can be defined as,*

$$\bar{L}_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{*i}) = \begin{cases} 1, & y_k^{*(i)} \in \bar{\Xi}_k \\ 0, & y_k^{*(i)} \notin \bar{\Xi}_k \end{cases}. \tag{4.20}$$

Finally, the importance weight obtained as,

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} \bar{L}_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{*(i)}) = \begin{cases} \omega_{k-1}^{(i)}, & y_k^{*(i)} \in \bar{\Xi}_k \\ 0, & y_k^{*(i)} \notin \bar{\Xi}_k \end{cases}. \tag{4.21}$$

After computing the equations (3.35) *and* (4.21), *we obtain* (4.13) *the proof completed.*

Using the predicted particle $x_k^{(i)}$ and the importance weights $\omega_k^{(i)}$ in Eq (4.21), The posterior distribution $p(x_k|y_{1:k})$ can be approximated as

$$p(x_k|y_{1:k}) \approx \hat{p}(x_k|\mathbb{Y}_k) = \sum_{i=1}^{M} \omega_k^{(i)} \delta(x_k - x_k^{(i)}). \tag{4.22}$$

4.3.1 Convergence study

In this section, we study the conference of the ETPF under packet loss in algorithm 4.1, according to the posterior approximation we have,

$$p(x_k|y_{1:k}) \approx \hat{p}(x_k|y_{1:k}) = \sum_{i=1}^{M} \frac{1}{M} \delta(x_k - x_k^{(i)}). \tag{4.23}$$

Where the importance weights computed in theorem 2. However, the study of the convergence can be done sing to the expectation of the posterior according the cases depends for trigger variable γ_k and packet loss λ_k as in section 4.3. Thus, we have,

(a) if $\gamma_k = 1$, $\lambda_k = 1$, then,

$$0 < \mathbb{E}(p(y_k|x_k^{(i)})) < \infty, \tag{4.24}$$

(b) if $\gamma_k = 0$, then,

$$0 < \mathbb{E}(m_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{(i)})) \leq 1 < \infty, \tag{4.25}$$

(c) if $\gamma_k = 1$, $\lambda_k = 0$, then,

$$0 < \mathbb{E}(\bar{L}_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{*(i)})) \leq 1 < \infty, \tag{4.26}$$

Hence, $0 < p(z_k|x_k^{(i)}) < \infty$.

The approximated $p(x_k|y_{1:k})$ posterior $p(x_k|y_{1:k})$ in (4.23) approaches to the true posterior, with $p(x_k|x_{k-1}) < \infty$ and the above three step are satisfied for $M \to \infty$ as in Theorem 1 in [47].

$$\lim_{x \to \infty} \hat{p}(x_k|\mathbb{Y}_k) = p(x_k|\mathbb{Y}_k). \tag{4.27}$$

Algorithme 4.1 : Event Trigger Particle Filter (ETPF) under packet loss

- *Initialization*:
  - For $i = 1 : M$, Draw $x_0^{(i)} \sim \pi_0$, Set $\omega_0^{(i)} = \frac{1}{M}$;
- **while** $k \leq T$
- *Time update*
  - **For** $i = 1 : M$, Draw $x_k^{(i)} \sim p_x(x_k|x_{k-1}^{(i)})$
- *Measurment update:*
  - **For** $i = 1 : M$, Draw $\mu_k^{(i)} \sim p_\mu(.)$, $v_k^{(i)} \sim p_v(.)$, $v_k^{*(i)} \sim p_{v_k}(v_k|\lambda_k = 0)$

    if $\gamma_k = 1$ and $\lambda_k = 1$ **then**

    $\omega_k^{(i)} = \omega_{k-1}^{(i)} p(y_k|x_k^{(i)})$,

    else if $\gamma_k = 1$ and $\lambda_k = 0$ **then**

    $\omega_k^{(i)} = \omega_{k-1}^{(i)} \bar{L}_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{*(i)})$,

    else

    $\omega_k^{(i)} = \omega_{k-1}^{(i)} m_*(x_k^{(i)}, \mu_k^{(i)}, v_k^{(i)})$,

    end if
  - **For** $i = 1 : M$, Calculate the normalized weight $\tilde{\omega}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^M w_k^{(i)}}$
- *Resampling*:
  - **For** $i = 1 : M$

    samples $x_k^{(j)}$ according to $Pr(x_k^j = x_k^{(i)}) = \tilde{\omega}_k^{(i)}$,

    set $\{x_k^{(i)}, \tilde{\omega}_k^{(i)}\} = \{x_k^{(j)}, \frac{1}{M}\}$,
- *Estimation*:

  $\hat{x}_{k|k} \approx \sum_{i=1}^M \tilde{\omega}_k^{(i)} x_k^{(i)}$,

  $P_{k|k} \approx \sum_{i=1}^M \tilde{\omega}_k^{(i)} (x_k^{(i)} - \hat{x}_{k|k})(x_k^{(i)} - \hat{x}_{k|k})^T$,
- set $k = k + 1$
- **end while**

4.4 Performance analysis

In this section, the performance of the proposed filter in algorithm 4.1 is evaluated by the PCRLB. However, due to the system's non-linearity and non-Gaussian, the upper bound for error covariance for PF sill a open challenge . Deriving an analytical expression for the pdf by eliminating intermediate variables is difficult, necessitating some form of approximation for nonlinear filtering. Although a closed-form solution for nonlinear PF is not available, The PCRLB provides the best achievable limit on the mean square error matrix [32]. However the PCRLB has gained significant popularity due to its efficient implementation where adopted and further refined through a recursive technique inspired by the same reference. To proceed the PCRLB, let $X_k = \{x_0, x_0, x_0, ..., x_k\}$ define all the state to time k, and observation set $y_{1:k}$ and $y_{1:k} \triangleq \{\{\gamma_0\ , .., \gamma_k\}, \{\lambda_0, .., \lambda_k\}, z_0, .., z_k\}$, define hybrid observation set. Let the distribution $p_k \triangleq p(X_k, \mathbb{Y}_k)$ as the joint pdf of the all state X_k and observation $y_{1:k}$. The covariance $P_{k|k}$ has a PCRLB $J_{x_k}^{-1}$ where,

$$\mathbb{E}\{(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^T\} \geq J_{x_k}^{-1}. \quad (4.28)$$

where J_{x_k} (Fisher information matrix) is defined as,

$$J_{x_k} = -\mathbb{E}[\Delta_{x_k}^{x_k} \log p_k].$$

where Δ_a^b is the second partial derivative operator defined as $\Delta_a^b = \Delta_b * \Delta_a^T$ with $\Delta_a = [\frac{\partial}{\partial a_1} \frac{\partial}{\partial a_2} \cdots \frac{\partial}{\partial a_m}]$ for $a = [a_1 a_2 ... a_m]^T$. Let the decompose of the state $X_k = \{X_k^T, x_k^T\}^T$ to compute J_{x_k}, then J_{x_k} can be accordingly decomposed as,

$$J_{x_k} = \begin{bmatrix} A_k & B_k \\ B_k^T & C_k \end{bmatrix} \triangleq \begin{bmatrix} -\mathbb{E}[\Delta_{x_{k-1}}^{x_{k-1}} \log p_k] & -\mathbb{E}[\Delta_{x_k}^{x_{k-1}} \log p_k] \\ -\mathbb{E}[\Delta_{x_k}^{x_{k-1}} \log p_k]^T & -\mathbb{E}[\Delta_{x_k}^{x_k} \log p_k] \end{bmatrix}. \quad (4.29)$$

As a result,

$$J_{x_k} = C_k - B_k^T (A_k)^{-1} - B_k.$$

Note that the computation of J_{x_k} requires the approximation of all matrices A_k, B_k and C_k. Where the anthers in [32, p. 68], employed such method to be calculate this matrices recursively. following the proposed solution's for general J_{x_k} that computed in [32, p. 68], we calculate the event-trigger PCRLB for the state x_k.

Assuming that the PCRLB J_{x_k} has already been computed at time k. Following the same approach as in the previous sections and the work in [32, p. 68], we consider three cases: (I) When the measurement triggered an received at the estimator without loss ($\gamma_{k+1} = 1$) and ($\lambda_{k+1} = 1$), (II) measurement is triggered with packet loss ($\gamma_{k+1} = 1$) and ($\lambda_{k+1} = 0$). (III) When the non-trigger case ($\gamma_{k+1} = 0$).

The following theorem summarize the recursive form of the PCRLB.

Theorem 3 *given the assumption in appendix A.2 and considering the joint pdf* $p_{k+1} \triangleq p(X_{k+1}, y_{1:k+1})$ *then, the RMSE of the state* x_{k+1} *is bounded with* $J_{x_{k+1}}^{-1}$ *where,*

$$P_{k|k} \approx \sum_{i=1}^{M} \tilde{\omega}_k^{(i)} (x_k^{(i)} - \hat{x}_{k|k})(x_k^{(i)} - \hat{x}_{k|k})^T \geq J_{x_{k+1}}^{-1}. \quad (4.30)$$

where $J_{x_{k+1}}$ *(Fisher information matrix) is defined as,*

$$J_{x_{k+1}} = D_k^{22} - D_k^{21} (J_{x_k} + D_k^{11})^{-1} - D_k^{12}. \quad (4.31)$$

where,

$$\begin{aligned} D_k^{11} &= -\mathbb{E}[\Delta_{x_k}^{x_k} \log p_x(x_{k+1}|x_k)], \\ D_k^{12} &= -\mathbb{E}[\Delta_{x_k}^{x_{k+1}} \log p_x(x_{k+1}|x_k)], \\ D_k^{21} &= [D_k^{12}]^T. \end{aligned} \quad (4.32)$$

and D_k^{22} *is defined as the following cases :*

(a) *if* $\gamma_k = 1,\ \lambda_k = 1$*:*

$$D_k^{22} = -\mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log p_x(x_{k+1}|x_k)] - \mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log p_x(y_{k+1}|x_k)].$$

(b) *if* $\gamma_k = 1,\ \lambda_k = 0$ *and* $\gamma_k = 0$ *:*

$$D_k^{22} = -\mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log p_x(x_{k+1}|x_k)].$$

Proof 3 *Consider, the PCRLB* J_{x_k} *is computed at time k, we compute the PCRLB of* $P_{k+1|k}$ *according to the definition all of pdf given in section 1.5.2.2. For each case we have*

(a) $\gamma_{k+1} = 1$, $\lambda_{k+1} = 1$*: All the observation at k+1 is given as* $y_{1:k+1} = \{y_k, \gamma_{k+1} = 1, \lambda_{k+1}, z_{k+1}\}$. *Thus, we have,*

$$\begin{aligned} p_{k+1} &= p(X_{k+1}, \mathbf{y}_{1:k+1}) = p(X_{k+1}|y_{1:k}, z_{k+1}, \lambda_{K+1}), \\ &= p(X_k, y_{1:k}) p_x(x_{k+1}|x_k) p_z(z_{k+1}|x_k), \\ &= p_k p_x(x_{k+1}|x_k) p_y(y_{k+1}|x_k). \end{aligned} \tag{4.33}$$

Then, the following recursion can be obtained as in [34],

$$J_{x_{k+1}} = D_k^{22} - D_k^{21}(J_{x_k} + D_K^{11})^{-1} - D_k^{12}. \tag{4.34}$$

where,

$$\begin{aligned} D_k^{11} &= -\mathbb{E}[\Delta_{x_k}^{x_k} \log p_x(x_{k+1}|x_k)], \\ D_k^{12} &= -\mathbb{E}[\Delta_{x_k}^{x_{k+1}} \log p_x(x_{k+1}|x_k)], \\ D_k^{21} &= [D_k^{12}]^T, \\ D_k^{22} &= -\mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log p_x(x_{k+1}|x_k)] - \mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log p_x(z_{k+1}|x_k)]. \end{aligned} \tag{4.35}$$

Using the particle filter approximation in section 3.3.1, all the expectation in (4.35) can be computed approximated and the recursive PCRLB $J_{x_{k+1}}$ *in (4.34) is obtained.*

(b) $\gamma_{k+1} = 1$, $\lambda_{k+1} = 0$*: the measurement si triggered with packet loss* $\mathbb{Y}_{k+1} = \{y_{1:k}, \lambda_{K+1} = 0, z_{k+1}\}$, *using equation in section 3,*

$$\begin{aligned} p_{k+1} &= p(X_{k+1}, \mathbf{y}_{1:k+1}) = p(X_{k+1}|y_{1:k}, z_{k+1}, \lambda_{K+1}) \\ &= p(X_k, y_{1:k}) p_x(x_{k+1}|x_k) \int_{z_{k+1} \in \bar{\Xi}_{k+1}} p(z_{k+1}|\lambda_{k+1} = 0, x_k, y_{1:k}) dz_{k+1} \\ &= p_k p_x(x_{k+1}|x_k) \int_{z_{k+1} \in \bar{\Xi}_{k+1}} p(z_{k+1}|\lambda_{k+1} = 0, x_k, y_{1:k}) dz_{k+1} \end{aligned} \tag{4.36}$$

Then, similar to the case A above, $J_{x_{k+1}}$ *can be obtained as*

$$J_{x_{k+1}} = \bar{D}_k^{22} - D_k^{21}(J_{x_k} + D_K^{11})^{-1} - D_k^{12} \tag{4.37}$$

where D_k^{11}, D_k^{12} and D_k^{21} similar as in equation (4.35), while $\bar{D}_k^{22}$ is obtained as follows,

$$\begin{aligned} D_k^{22} = &- \mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log p_x(x_{k+1}|x_k)] \\ &- \mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log \int_{z_{k+1} \in \bar{\Xi}_{k+1}} p(z_{k+1}|\lambda_{k+1} = 0, x_k, \mathbb{Y}_k) dz_{k+1}] \end{aligned} \tag{4.38}$$

Due to the non trigger set of measurement $z_k \in \bar{\Xi}_{k+1}$. the integral $\int_{z_{k+1} \in \bar{\Xi}_{k+1}} p(y_{k+1}|\lambda_{k+1} = 0, x_k, y_{1:k}) dz_{k+1}$ approximated to a constant number. Therefore, its gradient is 0. As a consequence, the equation (4.38) become,

$$D_k^{22} = -\mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log p_x(x_{k+1}|x_k)]. \tag{4.39}$$

Similar to the case A, the recursive PCRLB matrix $J_{x_{k+1}}$ still have the same form, but with a modified D_k^{22}.

(c) $\gamma_{k+1} = 0$, $\forall \lambda_{k+1}$*: the measurement is non-triggered, the observation become,* $\mathbf{y}_{1:k+1} = \mathbf{y}_{1:k-1} \cup \{\gamma_k = 0\}$*. Similar to the derivation equation in the case B,*

$$p_{k+1} = p_k p_x(x_{k+1}|x_k) \int_{z_k \in \Xi_k} p(z_{k+1}|x_k, y_{1:k}) dz_{k+1}. \tag{4.40}$$

Due to the trigger set of measurement $z_k in \Xi_k$. the integral in (4.40) reduced constant number. Therefore, its gradient is 0. As a consequence, the equation (4.38) become,

$$\begin{aligned} \bar{D}_k^{22} &= -\mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log p_x(x_{k+1}|x_k), \\ &\quad - \mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log \int_{z_{k+1} \in \Xi_{k+1}} p(y_{k+1}|x_k, \mathbb{Y}_k) dz_{k+1}], \\ &= -\mathbb{E}[\Delta_{x_{k+1}}^{x_{k+1}} \log p_x(x_{k+1}|x_k)]. \end{aligned} \tag{4.41}$$

The computation of $\bar{D}_k^{22}$ gives the same form as in the Case B,

Finally, from equation (4.35) and $\bar{D}_k^{22}$ from (4.39) and (4.41) the recursion form in equation (4.34) can be computed and the proof is completed.

Remark 3 *The PCRLB form the best achievable of ETPF for nonlinear and non-Gaussian system by approximate the recursive form (4.31) in theorem 3.*

Remark 4 *The presence of an event trigger scheduler and packet loss, leads to a non-Gaussian posterior density. Fully characterizing non-Gaussian distributions requires high-order moments. While the PCRLB provides the best achievable lower bound, it does not fully capture the performance of nonlinear filtering algorithms such as Algorithm 4.1. The boundedness of the RMSE depends on the boundedness of the expected PCRLB, which is challenging to determine directly. To obtain all bounds for the expected PCRLB, researchers have considered and researchers have examined specific restricted nonlinear systems [48, 49].*

4.5 Simulation results

In this section, simulation experiments are conducted to evaluate the performance of the proposed ETPF, as outlined in Algorithm 1. Drawing from recent literature on event-based state estimation, we use the example of an air traffic control system [43]. The state of the target is represented by $X_k = [x_k, \dot{x}_k, y_k, \dot{y}_k, w_k]$, where x and y denote the position coordinates along the x-axis and y-axis, respectively, and $\dot{x}$ and $\dot{y}$ denote their corresponding velocities. The term w_k represents the turn rate. Observations from the sensor are denoted by $z_k = [r_k, \theta_k]$, where r_k represents the range and θ_k represents the bearing. The nonlinear model for the kinematics of the turning motion is expressed as follows:

$$X_{k+1} = \begin{pmatrix} 1 & \frac{\sin(w_k T_{in})}{w_k} & 0 & \frac{1-\cos(w_k T_{in})}{w_k} & 0 \\ 0 & \cos(w_k T_{in}) & 0 & -\sin(w_k T_{in}) & 0 \\ 0 & \frac{1-\cos(w_k T_{in})}{w_k} & 1 & \frac{\sin(w_k T_{in})}{w_k} & 0 \\ 0 & \sin(w_k T_{in}) & 0 & \cos(w_k T_{in}) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} X_k + \mu_k, \tag{4.42}$$

$$z_k = \begin{pmatrix} r_k \\ \theta_k \end{pmatrix} = \begin{pmatrix} \sqrt{x_k^2 + y_k^2} \\ \tan^{-1} \frac{y_k}{x_k} \end{pmatrix} + v_k. \tag{4.43}$$

The time-interval is $T_{in} = 1s$, the covariance of process noises are given as $Q = diag(q_1 M, q_1 M, q_2 T)$, and for the measurement $R = diag(\sigma^2, \sigma)$ where,

$$M = \begin{pmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{pmatrix}.$$

Where $\sigma = 10$ and the parameters $q_1 = 0.1ms^{-1}$, $q_2 = 1.75 * 10^{-4}s^{-3}$. The radar fixed in $[20000, 20000]$ in meters to measure the the bearing θ_k and range r_k. The time sample in this simulation is given $N = 500$, for full communication rate, and simulation scenarios are discussed,

Case 1 : In this scenario, the initial state $x_0 = [10^4 m, 150ms^{-1}, 3.5 * 10^4 m 0ms^{-1}, -3s^{-1}]$ with initial covariance $P_0 = diag[10^2 m^2, 10m^2 s^{-2} 10^2 m^2, 10ms^{-2} 0.1rad^2 s^{-2}]$, To test the performance of the ETPF algorithm. with /Without packet loss, the communication rate reduction by $(70\%, 30\%,$ and $15\%)$, for different thresholds ξ (100, 400, and 800), respectively. The number of particles $M = 1000$ and in the simulation result we refer to the ETPF with loss as ETPF and the ETCKF with loss as ETCKf as in Fig. 4.4a.

As result of simulation in the 4.4a, the estimate state of proposed ETPF Algorithm follow the trajectory even if the communication rate reduced. Let the simulation runs with packet loss rate (α=0.2 and 0.5) as in Figure 4.4b and 4.4c. its shown that estimation performance remains guaranteed for the same communication rate and in the presence of the packet loss rate

Case 2: we set the initial state with $x_0 = [10^3 m, 300ms^{-1}, 10^3 m, 0ms^{-1}, -3s^{-1}]$ with initial covariance $P_0 = diag[10^2 m^2,$
$10m^2 s^{-2} 10^2 m^2, 10ms^{-2} 0.1rad^2 s^{-2}]$. First, we set without packet loss ($\alpha = 0$) and event-triggered thresholds (100, 400) . Second, we compare ETPF and ETCKF from [43] for different packet loss rates (α = 0.2, 0.5), fixed threshold for (ξ = 400). As result of simulation in figure 4.3, both the ETEKF and [43] provide good estimate for low communication rate and packet loss. However,

we shown that from 4.3, the ETPF performer with good state estimate compared to ETCKF for communication rate reduced to 30% and 20% of packet loss, if the communication rate reduced more and packet loss increased both filter start to degraded.

The tracking results of the ETEKF and ETUKF are not mentioned in this simulation experiment. Due to the ETEKF may introduce the large error induced by the linearization of the nonlinear system model. In addition, in chapter 3 the [43] illustrates a comparison of estimation errors between ETCKF and ETUKF, revealing superior results for ETCKF over ETUKF.

To study the error performance of the proposed ETPF with packet loss, the RMSE is calculated as

$$RMSE_k = \sqrt{\frac{1}{RN}\sum_{i=1}^{RN}(x_k^{(i)} - \hat{x}_{k|k}^{(i)})^2 + (y_k^{(i)} - \hat{y}_{k|k}^{(i)})^2} \tag{4.44}$$

with true and state estimate $x_k^{(i)}, y_k^{(i)}$ and $\hat{x}_{k|k}^{(i)}, \hat{y}_{k|k}^{(i)}$ computed via ETPF Algorithm. The RMSE is computed for the simulation of the state as in figure 4.3b for independent Monte Carlo runs $RN = 20$ and the result are presented in in Table 4.1 and Fig. 4.4.

Figure 4.4 and Table 4.1 demonstrates that performance declines as the packet dropout rate increases. From time $t = 0$ to $t = 40$, both filters provide good performance. However, at $t > 40$, the state transition becomes highly nonlinear, making it difficult for the ETCKF with packet loss [43] to track changes in the system state accurately. This degradation in filtering accuracy is due to the lack of transmitted information. In contrast, the ETPF continues to deliver acceptable filtering performance under reduced communication and increased packet loss rates, ensuring the robustness of the proposed algorithm.

In Figure 4.5, the RMSE error performance of the ETPF is illustrated for different particle numbers: 500, 1000, and 2000, with an event-triggered threshold of 100 and a packet loss rate of 0.2. For comparison, the error curves for the PCRLB and the ETCKF with packet loss are also shown. It is observed that the ETPF performs better and approaches the PCRLB limit as the number of particles increases, thereby improving the accuracy of the ETPF algorithm.

Because the ETPF inherits the high accuracy advantage of the PF, it offers greater precision and accuracy compared to the ETCKF with packet loss. Therefore, the proposed ETPF algorithm presents a viable alternative for state estimation in event-triggered nonlinear systems experiencing packet loss.

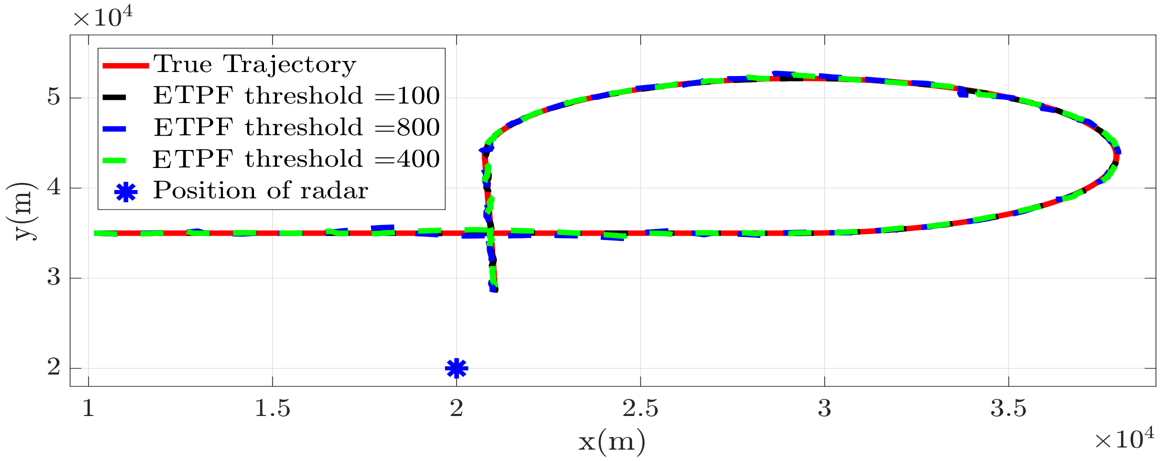
Tableau 4.1: Estimation performance with packet dropout

packet loss rate(α)	ETCKF (RMSE)	ETPF (RMSE)
0%(α=0)	141	92.73
20%(α=0.2)	189.71	126.53
50%(α=0.5)	362.74	183.35

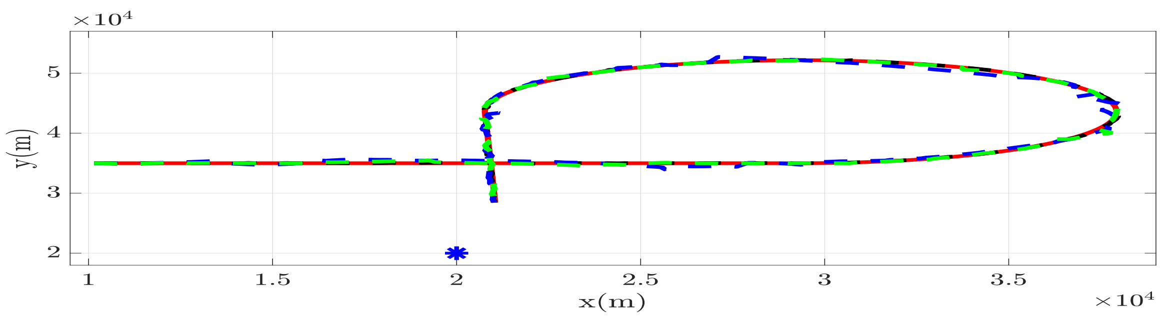
4.6 Conclusion

In this chapter, a discrete-time event-trigger Particle Filter for a nonlinear dynamic system with packet dropout is investigated. Initially, an event-trigger strategy is developed as an SOD event mechanism, which reduces the number of data transmissions between the sensors and the state estimator in the presence of packet dropout. The proposed ETPF algorithm provides a good approximation of the posterior distribution for the nonlinear, non-Gaussian system with packet dropout.

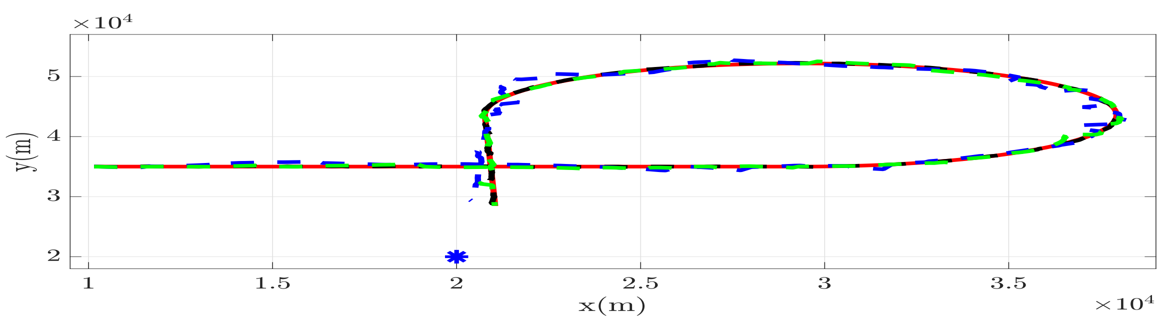
Secondly, the corresponding filtering algorithm is enhanced to specifically include a recursive RMSE estimator, and a lower bound for the PCRLB with packet loss is derived. The findings confirm that the ETPF has superior performance compared to the ETCKF under the same communication constraints and can still perform effectively even when the communication rate is low. Performance evaluation shows that the root mean square estimation error of the ETPF under different communication rates is closer to the PCRLB than that of the ETCKF.



(a)

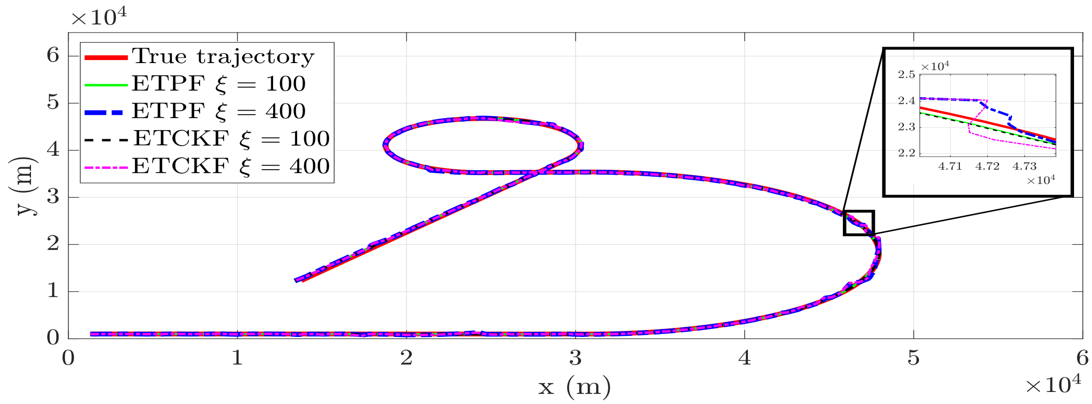


(b)

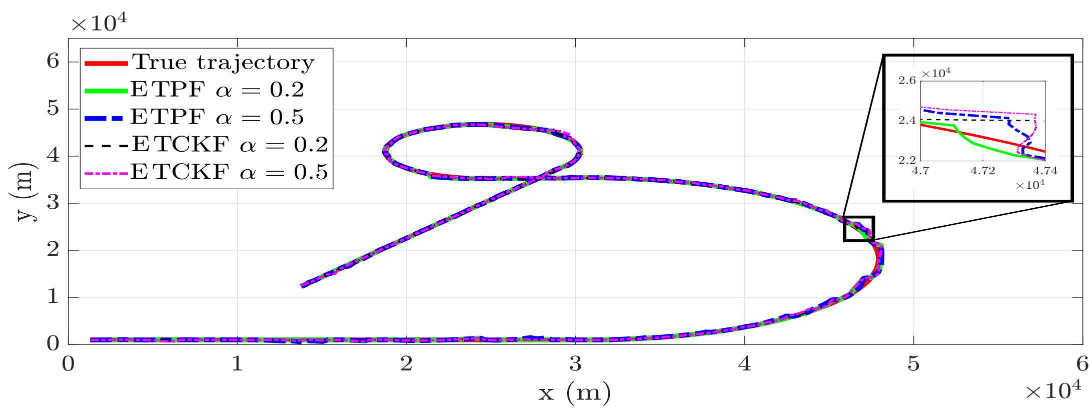


(c)

Figure 4.2: State estimation ETPF case 1 (a) Without packet loss ($\alpha = 0$) (b) With packet loss ($\alpha = 0.2$) (c) With packet loss ($\alpha = 0.5$).

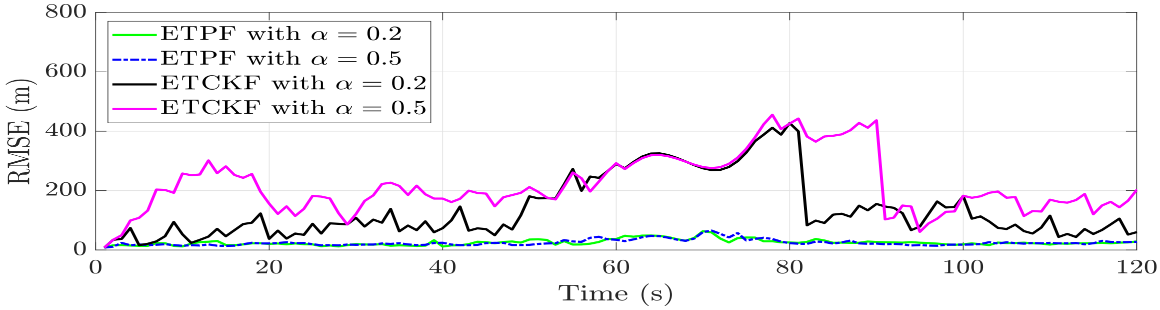


(a)

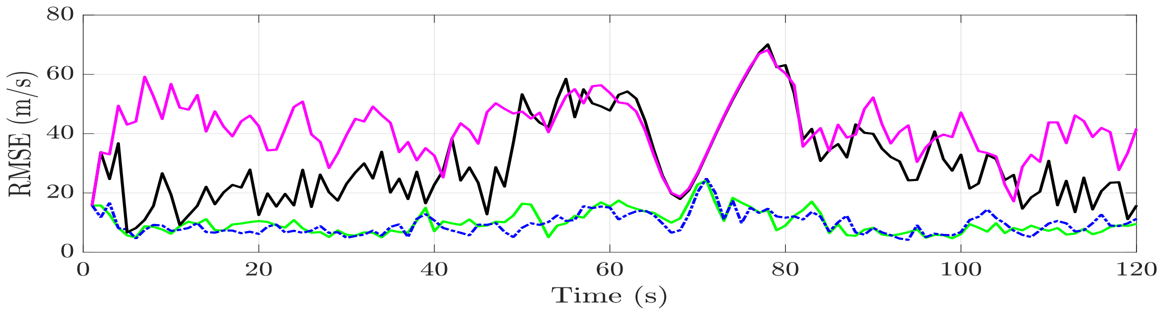


(b)

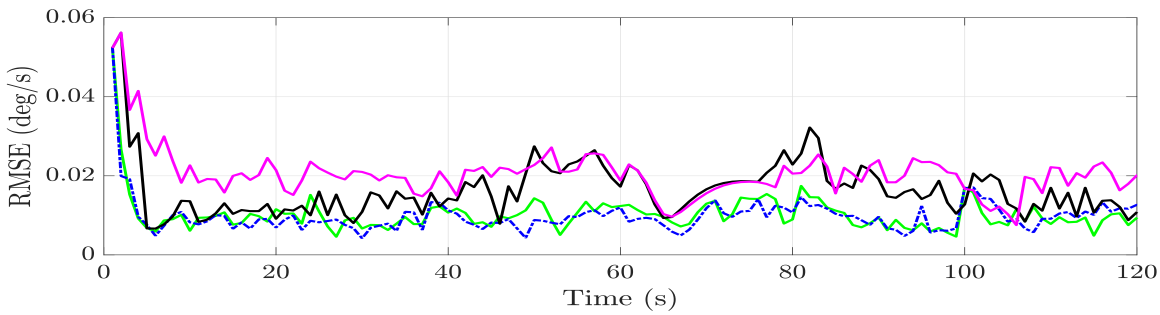
Figure 4.3: Comparison between ETPF and ETCKF state estimation. (a) Different triggering threshold ($\alpha = 0.2$) (b) Different packet loss ($\xi = 400$).



(a)



(b)



(c)

Figure 4.4: Comparison between ETPF and ETCKF RMSE with a fixed threshold ($\xi = 100$) (a) Position (b) Velocity (c) Turn rate.

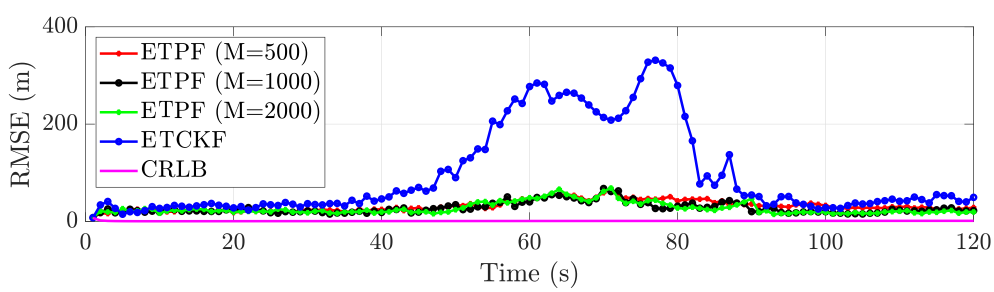


Figure 4.5: Illustration of RMSE Position ETPF and ETCKF with packet loss rate 0.2 and threshold ($\xi = 400$)and corresponding CRLB.

Chapter 5

Conclusions and Future Work

State estimation is a highly active research area with a substantial body of existing work. For linear time-invariant systems, Kalman filters are the gold standard, serving as an essential tool for estimating the system state while minimizing the variance of the estimation error. However, many applications involve nonlinear systems, necessitating the development of appropriate nonlinear filters for accurate estimation.

Traditionally, sensor information is directly available for processing in estimation formulations. However, in recent years, systems and control have increasingly relied on wireless communication networks to interconnect various system components, requiring careful management of communication resources.

In this thesis, we address the development of nonlinear event-based state estimators to tackle the challenges mentioned. The primary goal is to reduce data transmission between different components of the networked system while maintaining comparable performance under varying conditions. We consider a discrete-time nonlinear system with additive Gaussian noise and focus on the problem of sequentially estimating the state using Bayes's rule. In this context, the posterior density cannot be represented by a finite number of statistics, necessitating an approximation.

To solve this, we develop event trigger non-Gaussian filter based particle filters, which provide a flexible and powerful solution to the nonlinear state estimation problem. Unlike Gaussian Kalman filters , particle filters do not assume that the predictive density of the joint state-measurement random variable is Gaussian. Instead, particle filters approximate the posterior distribution using a set of weighted particles, representing possible states of the system. This approach allows for the accurate representation of a wide range of distributions, making particle filters particularly well-suited for nonlinear and non-Gaussian estimation problems. By using a resampling mechanism, particle filters effectively address the issue of particle degeneracy and maintain a robust and accurate state estimation.

The outcomes of our research attempts are further summarized as follows:

- First, assuming a Gaussian conditional distribution of the state within an event-triggered mechanism, we study an event-based state Gaussian estimator, specifically the discrete-time event-triggered extended, unscented and cubature Kalman filter, for nonlinear systems. This filter utilizes the "send on delta" event-triggered mechanism, which minimizes the number of feedback communications between sensors and the estimator. Additionally, it reduces measurement transmissions between the sensors and the remote state estimator while ensuring the maintenance of estimation performance.

- Second, Assuming a non-Gaussian conditional distribution of the state within an event-triggered mechanism, we develop an event-triggered particle filter, where the probability density function is approximated using a Monte Carlo approach. Bayesian constraints are proposed to address the computational burden associated with calculating the likelihood in of the non-trigger case.

- Finally, we investigate the impact of packet dropouts during data transmission in communication channels on state estimation and we devolved an event trigger particle filter for nonlinear system with packet dropout. and we study the performance of developed filter using Cramér-Rao lower bound.

The research results, provided in this thesis can be extended and pursued in the following areas:

- **Event trigger sum of Gaussian filter** in the second chapter we discussed a different event trigger filter under the Gaussian assumption, and we proposed the non-Gaussian particle filter to tackle the problem on non-Gaussianity. The particle filter performance based on the increased number of particle ,which no longer to be applicable in such limited real time system . Thus we propose as a future work, to approximate the non Gaussian pdf based on sum of gaussian.
- **Event trigger particle filter with one step delay measurement** the one step or randomly delayed measurement is inevitable in networked control system, we propose to extend the work of event trigger particle filter to tackle the problem of system with one step delay.
- **distributed Event trigger particle filter** in our research we discussed the event trigger problem for single sensor. However, in real CPS application most of the system based on multi sensor or distributed system . Thus we propose to extend this work for multi agent and multi sensor.

Appendix A

Appendix A

A.1 Bayes rule

$$p(A|B) = p(B|A)p(A)/p(B)$$

A.2 Markov assumption

Assumption 1 *The state vector x_k is assumed to be Markovian, i.e: $p_x(x_{k+1}|x_{0:k}) = p_x(x_{k+1}|x_k)$, the measured output of the sensor z_k and the state x_k are given by $p_z(z_k|x_{0:k}, y_{1:k-1}) = p_z(z_k|x_k)$.*

Appendix B

Appendix B

B.1 Event trigger Gaussian approximation

B.1.1 Event trigger EKF

```
function [x_est,P_est,Kgain,gamt] = etekf(y,X0,P0,Q,R,del)
n_iter=size(y,1);
dt = 0.01; %seconds
%Filter Arrays
x_est   = zeros(n_iter,1); %state estimate
x_pred  = zeros(n_iter,1); %state estimate prediction
y_pred  = zeros(n_iter,1); %measurement prediction
P_est   = zeros(n_iter,1); %state covariance estimate
P_pred  = zeros(n_iter,1); %state covariance prediction
innov   = zeros(n_iter,1); %innovation
R_innov = zeros(n_iter,1); %innovation covariance
Kgain   = zeros(n_iter,1); %gain

%Filter initial conditions
x_est(1) = X0; %original 2.0
P_est(1) = P0; %original 0.01
zbar=0;
%Noise
Rww_fil = Q;
Rvv_fil = R ;
for k = 2:n_iter

    %Prediction  x = (1-0.05*dt)*x + 0.04*dt*(x^2);
    x_pred(k) = f(x_est(k-1),dt);
    y_pred(k) = x_pred(k)^2 + x_pred(k)^3;
    A = (1-0.05*dt) + 0.08*dt*x_pred(k);
    C = 2*x_pred(k) + 3*(x_pred(k)^2);
    P_pred(k) = A*P_est(k-1)*A + Rww_fil;
```

```
    %Innovation
    innov(k) = y(k) - y_pred(k);
    R_innov(k) = C*P_pred(k)*C + Rvv_fil;
    pxy          =P_pred(k)*C;



    %triger schedular
     %trigger
     if (y(k) - zbar)'*(y(k)- zbar) > del
     gamt(k) = 1;
     zbar = y(k);
     else
     gamt(k) = 0;
     end
     alp1 = 0.3;  % 0.02 In Li Li paper
         alp2 = 0.35; % 0.02;

     if gamt(k) == 1
    %Update
     %Kalman Gain
    Kgain1(k) = P_pred(k)*C/R_innov(k);
    x_est(k) = x_pred(k) + Kgain1(k)*innov(k);
    P_est(k) = ( 1 - Kgain1(k)*C )*P_pred(k);
     else
     %Kalman Gain
    Kgain(k) =(1 + alp1)*pxy*inv((1 + alp1)*pxy*inv(P_pred(k))*pxy+(1 + alp2)*Rvv_fil+(1 + 1/alp1+ 1/alp2)*del);
    x_est(k) = x_pred(k) + Kgain(k)*(zbar- y_pred(k));
    P_est(k) = (1 + alp1)* (1-Kgain(k)*pxy*inv(P_pred(k)))*P_pred(k)*(1-Kgain(k)*pxy*inv(P_pred(k)))+(1 + alp2)*Kgain(k)*Rvv_fil*Kgain(k)+ (1 + 1/alp1+ 1/alp2)*Kgain(k)*del*Kgain(k);
     end
end
```

B.1.2 Event trigger UKF

```
function [x_est,P_est,Kgain,gamt] = etukfg(y,X0,P0,Q,R,del)
n_iter=size(y,1);
dt = 0.01; %seconds
%Filter Arrays
x_est    = zeros(n_iter,1); %state estimate
x_pred   = zeros(n_iter,1); %state estimate prediction
```

```
y_pred   = zeros(n_iter,1); %measurement prediction
P_est    = zeros(n_iter,1); %state covariance estimate
P_pred   = zeros(n_iter,1); %state covariance prediction
innov    = zeros(n_iter,1); %innovation
R_innov = zeros(n_iter,1); %innovation covariance
Kgain    = zeros(n_iter,1); %gain

%Filter initial conditions
x_est(1) = X0; %original 2.0
P_est(1) = P0; %original 0.01
n = 1;
zbar=0;

%Noise
Rww_fil = Q;
Rvv_fil = R;
for k = 2:n_iter
    %%Prediction state
%         s1 = chol(P_est, 'lower');
        [xhat3,ppri1] = UTx(x_est(k-1),P_est(k-1),dt);
        x_pred(k) = xhat3;
        P_pred(k)=ppri1+Rww_fil;
        %%Prediction measurement
        [ypri1,pypri1,pxy] = UTy(xhat3,P_pred(k),dt);

    %Innovation
    innov(k) = y(k) - ypri1;
    R_innov(k) = pypri1 + Rvv_fil;



    %triger schedular
     %trigger
     if (y(k) - zbar)'*(y(k)- zbar) > del
     gamt(k) = 1;
     zbar = y(k);
     else
     gamt(k) = 0;
     end
     alp1 = 0.3;  % 0.02 In Li Li paper
        alp2 = 0.35; % 0.02;

     if gamt(k) == 1
```

```
    %Update
     %Kalman Gain
    Kgain1(k) = pxy/R_innov(k);
    x_est(k) = x_pred(k) + Kgain1(k)*innov(k);
    P_est(k) = P_pred(k) - Kgain1(k)*R_innov(k)*Kgain1(k)';
     else
     %Kalman Gain
    Kgain(k) =(1 + alp1)*pxy*inv((1 + alp1)*pxy*inv(P_pred(k))*pxy+(1 +
        alp2)*Rvv_fil+(1 + 1/alp1+ 1/alp2)*del);
    x_est(k) = x_pred(k) + Kgain(k)*(zbar- ypri1);
    P_est(k) = (1 + alp1)* (1-Kgain(k)*pxy*inv(P_pred(k)))*P_pred(k)*(1-
        Kgain(k)*pxy*inv(P_pred(k)))+(1 + alp2)*Kgain(k)*Rvv_fil*Kgain(k
        )+ (1 + 1/alp1+ 1/alp2)*Kgain(k)*del*Kgain(k);
     end

end
```

the unscented transformation:

```
function [xhat3,SigX3] = UTx(x_est,P_est,dt)
nxa = 1;
col=2*nxa+1;
alp = 0.2;
kepa = 3-nxa;
beta = 2;
lambda = alp^2*(nxa+kepa)-nxa;
alpham1 = lambda/(nxa+lambda);
alphamk = 1/(2*(nxa+lambda));
alphac1 = lambda/(nxa+lambda)+(1-alp^2+beta);
alphack = 1/(2*(nxa+lambda));
alpha_mean = [alpham1 alphamk(:,ones(1,2*nxa))]';
alpha_cov = [alphac1 alphack(:,ones(1,2*nxa))]';
        s1=sqrt(P_est);
        xhat_aug = x_est;
        X = xhat_aug(:,ones([1 col]))+sqrt(nxa+lambda)*[zeros(nxa,1), s1
            ,-s1];
        X_x =f(X,dt);
        xhat3 = X_x*alpha_mean;
        square_X = (X_x(:,2:end) - xhat3(:,ones(1,2*nxa)))*sqrt(alphack)
            ;
    square_X1 =X_x(:,1)-xhat3;
    SigX3 = square_X*square_X'+alphac1*square_X1*square_X1;
end
%
% k = 3-n;
```

```
% lamda=(a^2)*(n+k)-n;
% col=2*n+1;
% Dm= zeros(1, col);
% W= zeros(2, col);
% W(1,1)= lamda\(lamda+n);
% W(2,1)=lamda\(lamda+n)+(1-a^2+b);
% for i=2:n+1
%       Dm(i)=sqrt(lamda+n);
%       Dm(i+n)=-sqrt(lamda+n);
%       W(:,i)=lamda\(2*(lamda+n));
%       W(:,i+n)=lamda\(2*(lamda+n));
%
% end
```

```
function [xhat3,SigX3,SigXy] = UTy(x_est,P_est,dt)
nxa = 1;
col=2*nxa+1;
alp = 0.2;
kepa = 3-nxa;
beta = 2;
lambda = alp^2*(nxa+kepa)-nxa;
alpham1 = lambda/(nxa+lambda);
alphamk = 1/(2*(nxa+lambda));
alphac1 = lambda/(nxa+lambda)+(1-alp^2+beta);
alphack = 1/(2*(nxa+lambda));
alpha_mean = [alpham1 alphamk(:,ones(1,2*nxa))]';
alpha_cov = [alphac1 alphack(:,ones(1,2*nxa))]';
          s1=sqrt(P_est);
          xhat_aug = x_est;
          X = xhat_aug(:,ones([1 col]))+sqrt(nxa+lambda)*[zeros(nxa,1), s1
              ,-s1];
          X_x =h(X,dt);
          xhat3 = X_x*alpha_mean;
          square_X = (X_x(:,2:end) - xhat3(:,ones(1,2*nxa)))*sqrt(alphack)
              ;
          square_y = (X(:,2:end) - x_est(:,ones(1,2*nxa)))*sqrt(alphack);
     square_Xy =X(:,1)-x_est;
     square_X1 =X_x(:,1)-xhat3;
     SigX3 = square_X*square_X'+alphac1*square_X1*square_X1;
     SigXy = square_y*square_X'+alphac1*square_Xy*square_X1;

end
```

B.1.3 Event trigger CKFF

```
function [x_est,P_est,Kgain,gamt] = etckfg(y,X0,P0,Q,R,del)
n_iter=size(y,1);
dt = 0.01; %seconds
%Filter Arrays
x_est   = zeros(n_iter,1); %state estimate
x_pred  = zeros(n_iter,1); %state estimate prediction
y_pred  = zeros(n_iter,1); %measurement prediction
P_est   = zeros(n_iter,1); %state covariance estimate
P_pred  = zeros(n_iter,1); %state covariance prediction
innov   = zeros(n_iter,1); %innovation
R_innov = zeros(n_iter,1); %innovation covariance
Kgain   = zeros(n_iter,1); %gain

%Filter initial conditions
x_est(1) = X0; %original 2.0
P_est(1) = P0; %original 0.01
n = 1;
zbar=0;

%Noise
Rww_fil = Q;
Rvv_fil = R;
for k = 2:n_iter
    %%Prediction state
%         s1 = chol(P_est, 'lower');
        [xhat3,ppri1] = CUTx(x_est(k-1),P_est(k-1),dt);
        x_pred(k) = xhat3;
        P_pred(k)=ppri1+Rww_fil;
        %%Prediction measurement
        [ypri1,pypri1,pxy] = CUTy(xhat3,P_pred(k),dt);

    %Innovation
    innov(k) = y(k) - ypri1;
    R_innov(k) = pypri1 + Rvv_fil;



    %triger schedular
     %trigger
     if (y(k) - zbar)'*(y(k)- zbar) > del
     gamt(k) = 1;
```

```
      zbar = y(k);
      else
      gamt(k) = 0;
      end
      alp1 = 0.3;  % 0.02 In Li Li paper
          alp2 = 0.35; % 0.02;

      if gamt(k) == 1
     %Update
      %Kalman Gain
     Kgain(k) = pxy/R_innov(k);
     x_est(k) = x_pred(k) + Kgain(k)*innov(k);
     P_est(k) = P_pred(k) - Kgain(k)*R_innov(k)*Kgain(k)';
      else
      %Kalman Gain
     Kgain(k) =(1 + alp1)*pxy*inv((1 + alp1)*pxy*inv(P_pred(k))*pxy+(1 +
         alp2)*Rvv_fil+(1 + 1/alp1+ 1/alp2)*del);
     x_est(k) = x_pred(k) + Kgain(k)*(zbar- ypri1);
     P_est(k) = (1 + alp1)* (1-Kgain(k)*pxy*inv(P_pred(k)))*P_pred(k)*(1-
         Kgain(k)*pxy*inv(P_pred(k)))+(1 + alp2)*Kgain(k)*Rvv_fil*Kgain(k
         )+ (1 + 1/alp1+ 1/alp2)*Kgain(k)*del*Kgain(k);
      end

end
```

the Cubature transformation:

```
function [xhat3,SigX3] = CUTx(x_est,P_est,dt)
nxa = 1;
col=2*nxa;
alphack = 1/(2*(nxa));
alpha_mean = [alphack(:,ones(1,col))]';
          s1=sqrt(P_est);
          xhat_aug = x_est;
          X = xhat_aug(:,ones([1 col]))+sqrt(nxa)*[ s1,-s1];
          X_x =f(X,dt);
          xhat3 = X_x*alpha_mean;
          square_X = (X_x - xhat3(:,ones(1,2*nxa)));
     SigX3 = (alphack)*(square_X*square_X');
end
%
% k = 3-n;
% lamda=(a^2)*(n+k)-n;
% col=2*n+1;
% Dm= zeros(1, col);
```

```
% W= zeros(2, col);
% W(1,1)= lamda\(lamda+n);
% W(2,1)=lamda\(lamda+n)+(1-a^2+b);
% for i=2:n+1
%       Dm(i)=sqrt(lamda+n);
%       Dm(i+n)=-sqrt(lamda+n);
%       W(:,i)=lamda\(2*(lamda+n));
%       W(:,i+n)=lamda\(2*(lamda+n));
%
% end
```

```
function [xhat3,SigX3,SigXy] = UTy(x_est,P_est,dt)
nxa = 1;
col=2*nxa;
alphack = 1/(2*(nxa));
alpha_mean = [alphack(:,ones(1,col))]';
          s1=sqrt(P_est);
          xhat_aug = x_est;
          X = xhat_aug(:,ones([1 col]))+sqrt(nxa)*[ s1,-s1];
          X_x =h(X,dt);
          xhat3 = X_x*alpha_mean;
          square_X = (X_x - xhat3(:,ones(1,2*nxa)));
          square_y = (X - x_est(:,ones(1,2*nxa)));
     SigX3 =(alphack)*( square_X*square_X');
     SigXy = (alphack)*(square_y*square_X');

end
```

B.2 Event trigger non-Gaussian approximation

```
function [X,W,xhat,Zold,LL] = etpfUpdate(X,W,z,Zold,delta)

global R;
global Q;
%
    %%=============================================================================

%%% Genrate a set of Cubature Points
%
    %%=============================================================================

%%%trigger sesor
    ==============================================================
 [LL, Zold]=triger(Zold,z,delta);
```

```
%
    %%=================================================================
M =size(X,2);    % No. of particles
 if LL==1
 e = repmat(z,1,M) - MstEq(X);     % Here y_k is available we compute y_k-
    E[y_k|I_{k-1}] by making use of y_k
W=W.*((2 * pi) ^ (- 5 / 2) * det(R) ^ (-0.5) *(exp(-0.5*sum(e.*(inv(R)*e
    ),1))));

 else
         yt =  repmat(Zold,1,M) - MstEq(X); % Here y_k is not available ,
             so we compute y_k-E[y_k|I_{k-1}] using predicted particles
          old=zeros(1,M);
          for i=1:20
          y_diff = yt+(chol(R)*randn(2,M)); % y_k-E[y_k|I_{k-1}] for each
              of the particle x_k^i stacked columnwise
          y_diff =(sum((y_diff).^2,1).^(1/2));
          y_diff(find(y_diff==0)) = 1;
          y_diff(find(abs(y_diff)>delta)) = 0;
          y_diff(find(y_diff~=0)) = 1;
          old=y_diff+old;
%            sum(old)
          end
         W=W.*old;
end
W=W./sum(W);
 Neff = 1 ./ sum(W.^2);
          if Neff <= M/3
              index = sysresample(W);                                % 4.
                  Resample
               X= X(:,index);
               W = (1/M).* ones(1, M);
          end
 xhat= sum(repmat(W,2,1).*X,2);
X= X+chol(Q)*randn(2,M);
```

The trigger function :

```
function [LL, Zold]=triger(Zold,z,delta)

                e=z-Zold ;
                f=(e'*e);
                d=norm((e));

```

```
            if d <=delta
             LL = 0;
            else
             LL= 1;
             Zold = z;
            end


end
```

The resampling function:

```
function i=sysresample(q)
qc=cumsum(q);    M=length(q);
u=([0:M-1]+rand(1))/M;
i=zeros(1,M);    k=1;
for j=1:M
  while (qc(k)<u(j))
    k=k+1;
  end
  i(j)=k;
end
end
```

Bibliography

[1] J. Baek, E. Hableel, Y. J. Byon, D. S. Wong, K. Jang and H. Yeo, "How to Protect ADS-B: Confidentiality Framework and Efficient Realization Based on Staged Identity-Based Encryption," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 3, pp. 690-700,(2017).

[2] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards Fully Autonomous Driving: Systems and Algorithms," 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, pp. 163-168, (2011).

[3] D. F. Opila, X. Wang, R. McGee, R.B.Gillespie, J. A. Cook, J. W. Grizzle, "An Energy Management Controller to Optimally Trade off Fuel Economy and Drivability for Hybrid Vehicles," IEEE Transactions on Control Systems Technology , vol. 20, No. 6, pp. 1490, (2012).

[4] "Science of Design for Societal-Scale Cyber-Physical Systems (CPS)" IEEE Transactions on Control Systems Technology, `https://archive.cps-vo.org/group/sdss-cps`, (2008).

[5] Y. Huang, W. Liang, H.B Yu, and Y. Xiao, "Target Tracking based on a Distributed Particle Filter in Underwater Sensor Networks," Wireless Communications and Mobile Computing, vol. 8, No. 8, pp. 1023–1033, (2008).

[6] A. Simonetto, T. Keviczky, R. Babuska, "Distributed Non-linear Estimation for Robot Localization using Weighted Consensus", In IEEE Inter. Con. on Robotics and Automation(ICRA), pp. 3026-3031, (2010).

[7] H. Aghajan and A. Cavallaro, "Multi-Camera Networks," Principles and Applications, New York: Academic, (2009).

[8] G.G. Rigatos, "Distributed Particle Filtering over Sensor Networks for Autonomous Navigation of UAVs," Advanced Strategies for Robot Manipulators, SciYo Publications, Croatia, (2010).

[9] M. Miskowicz, Send-on-delta concept: An event-based data reporting strategy, Sensors, 6 49–63 (2006).

[10] J. Wu, Q. S. Jia, K. H. Johansson, L. Shi, Event-based sensor data scheduling: Trade-off between communication rate and estimation quality, IEEE Transactions on Automatic Control 58 (4) 1041–1046 (2013).

[11] A. Mohammadi, K. N. Plataniotis, Event-Based Estimation with Information-Based Triggering and Adaptive Update, IEEE Transactions on Signal Processing 65 (18) 4924–4939 (2017).

[12] D. Han, Y. Mo, J. Wu, S. Weerakkody, B. Sinopoli, L. Shi, Stochastic event-triggered sensor schedule for remote state estimation, IEEE Transactions on Automatic Control 60 (10) 2661–2675 (2015).

[13] Z. Q. Luo, An isotropic universal decentralized estimation scheme for a bandwidth constrained ad hoc sensor network, IEEE Journal on Selected Areas in Communications 23 (4) 735–744 (2005).

[14] A. Ribeiro, G. B. Giannakis, Bandwidth-constrained distributed estimation for wireless sensor networks - Part I: Gaussian case, IEEE Transactions on Signal Processing 54 (3) 1131–1143 (2006).

[15] A. Ribeiro, G. B. Giannakis, S. I. Roumeliotis, Soi-kf: Distributed kalman filtering with low-cost communications using the sign of innovations, IEEE Transactions on Signal Processing 54 4782–4795 (2006).

[16] J. Sijs, M. Lazar, Event-based state estimation with time-synchronous updates, IEEE Transactions on Automatic Control 57 2650–2655 (2012).

[17] R. Hossein, M. Ghorbani, Event-triggered resilient distributed extended Kalman filter with consensus on estimation, International Journal of Robust and Nonlinear Control 32 1303-1315 (2022).

[18] W. Song, J. Wang, S. Zhao , J.Shan, Event-triggered cooperative unscented Kalman filtering and its application in multi-UAV systems, Automatica 105 264-273 (2019).

[19] Li. Li, D. Yu, Y. Xia, H. Yang, Event-triggered UKF for nonlinear dynamic systems with packet dropout, Automatica 27 4208-4226 (2017).

[20] S. Li, Z. Li, J. Li, T. Fernando, H. H. C. Iu, Q. Wang, X. Liu, Application of event-triggered cubature Kalman filter for remote nonlinear state estimation in wireless sensor network, IEEE Transactions on Industrial Electronics 68 5133–5145 (2021).

[21] L. S. D. Shi, T. Chen, Event-triggered maximum likelihood state estimation, Automatica 50 247–254 (2014).

[22] Y. Niu, L. Sheng, M. Gao, D. Zhou, Dynamic Event-Triggered State Estimation for Continuous-Time Polynomial Nonlinear Systems with External Disturbances, IEEE Transactions on Industrial Informatics 17 (6) 3962–3970 (2021).

[23] S. Li, K. Liang, W. He, Fully distributed event-triggered secure consensus of general linear multi-agent systems under sequential scaling attacks, ISA Transactions 127 146–155 (2022).

[24] L. Liang, S. Liu, Y. Li, M. Zhong, Multi-agent based event-triggered distributed cooperative fault detection, ISA Transactions 129 69–78 (2022).

[25] J. Huang, J. Fan, T. N. Dinh, X. Zhao, Y. Zhang, Event-triggered interval estimation method for cyber-physical systems with unknown inputs, ISA Transactions 135 1–12 (2023).

[26] K. Yin, D. Yang, Asynchronous fault detection filter of positive Markov jump systems by dynamic event-triggered mechanism, ISA Transactions 138 197–211 (2023).

[27] A. Basit, M. Tufail, M. Rehan, I. Ahmed, A new event-triggered distributed state estimation approach for one-sided Lipschitz nonlinear discrete-time systems and its application to wireless sensor networks, ISA Transactions 137 74–86 (2023).

[28] Z. Echreshavi, M. Farbood, M. Shasadeghi, Disturbance observer-based fuzzy event-triggered ismc design: Tracking performance, ISA Transactions 138 243–253 (2023).

[29] D. D. Zhou, Dynamic event-triggered distributed observer for linear systems, ISA Transactions 137 87–97 (2023).

[30] A. Basit, M. Tufail, M. Rehan, An adaptive gain-based approach for event-triggered state estimation with unknown parameters and sensor nonlinearities over wireless sensor networks, ISA Transactions 129 41–54 (2022).

[31] D. Shi, L. Shi, T. Chen, Event-Based State Estimation: A Stochastic Perspective, Springer, (2016).

[32] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman filter: Particle filters for tracking applications, Artech house, (2003).

[33] M. Cea and G. Goodwin, Event-based sampling in non-linear filtering, Control Engineering Practice, 20 963–971 (2012).

[34] M. A. Sid, S. Chitraganti, Nonlinear event-based state estimation using particle filtering approach, International Conference on Modelling, Identification and Control ICMIC , 874–879 (2016).

[35] E. Gasmi, M. A. Sid, O. Hachana, Event-Triggered State Estimation Using Particle Filtering Approach. In 2022 2nd International Conference on Advanced Electrical Engineering (ICAEE) . IEEE, 1–5 (2022).

[36] S. Davar and A. Mohammadi, "Event-based particle filtering with point and set-valued measurements," in 25th European Signal Processing Conference, Kos, Greece, Aug., pp. 211–215 (2017).

[37] X. Liu, L. Li, Z. Li, X. Chen, T. Fernando, H. H. C. Iu, G. He, Event-trigger particle filter for smart grids with limited communication bandwidth infrastructure, IEEE Transactions on Smart Grid 9 (6) 6918–6928 (2018).

[38] J. Ruuskanen, A. Cervin, Event-based state estimation using the auxiliary particle filter, 2019 18th European Control Conference, ECC 2019 1854–1860 (2019).

[39] S. Yan, H. Qian, P. Ding, S. Chu, H. Wang, Finite-time tolerant containment control for it2 t–s fuzzy network multi-agent systems with actuator faults, packet dropouts and dos attacks, ISA Transactions 137 199–209 (2023).

[40] C. Li, X. Zhao, C. Wu, L. Liu, N. Zhao, Periodic event-triggered dynamic output feedback control for networked control systems subject to packet dropouts, ISA Transactions (6) (2023).

[41] H. Rezaei, R. M. Esfanjani, A. Akbari, M. H. Sedaaghi, Scalable event-triggered distributed extended Kalman filter for nonlinear systems subject to randomly delayed and lost measurements, Digital Signal Processing, 111 102957 (2021).

[42] L. Li, D. Yu, Y. Xia, H. Yang, Event-triggered UKF for nonlinear dynamic systems with packet dropout, International Journal of Robust and Nonlinear Control 27 (18) 4208–4226 (2017).

[43] M. Kooshkbaghi, H. J. Marquez, Event-Triggered Discrete-Time Cubature Kalman Filter for Nonlinear Dynamical Systems with Packet Dropout, IEEE Transactions on Automatic Control 65 (5) 2278–2285, (2020).

[44] M. Kooshkbaghi, Event Triggered Cubature Kalman Filter, University of Alberta, (2020).

[45] L. Lang, W. shiang Chen, B. R. Bakshi, P. K. Goel, S. Ungarala, Bayesian estimation via sequential Monte Carlo sampling—Constrained dynamic systems, Automatica 43 (9) 1615–1622 (2007).

[46] X. Shao, B. Huang, J. M. Lee, Constrained Bayesian state estimation – A comparative study and a new particle filter based approach, Journal of Process Control 20 (2) 143–157 (2010).

[47] D. Crisan, A. Doucet, A survey of convergence results on particle filtering methods for practitioners, IEEE Transactions on signal processing 50 (3) 736–746 (2002).

[48] C. Yang, H. Fang, Modified particle filter and Gaussian filter with packet dropouts. International Journal of Robust and Nonlinear Control, 28 (8) 2961-2975 (2018).

[49] C. Yang, H. Fang, B. She, Particle filter with Markovian packet dropout and time delay. Journal of the Franklin Institute, 365 (1) 675-696 (2019).