UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DESIGN OF EVOLVING FUZZY PREDICTION INTERVALS

TESIS PARA OPTAR AL GRADO DE DOCTOR EN INGENIERÍA ELÉCTRICA

OSCAR ANDRÉS CARTAGENA VILLALOBOS

PROFESORA GUÍA:
DORIS SÁEZ HUEICHAPAN

MIEMBROS DE LA COMISIÓN:
DANIEL SBARBARO HOFER
CARLOS BORDONS ALBA
PABLO ESTÉVEZ VALENCIA

Este trabajo ha sido parcialmente financiado por el Instituto Sistemas Complejos de Ingeniería (ISCI): ANID PIA AFB230002, el Centro de Investigación en Energía Solar (SERC-Chile): ANID/FONDAP/1522A0006 y la Agencia Nacional de Investigación y Desarrollo (ANID): mediante el proyecto ANID/FONDECYT 1220507 y la beca de doctorado ANID-PFCHA/Doctorado Nacional/2020-21200709

SANTIAGO DE CHILE
2024

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE
DOCTOR EN INGENIERÍA ELÉCTRICA
POR: OSCAR ANDRÉS CARTAGENA VILLALOBOS
FECHA: 2024
PROF. GUÍA: DORIS SÁEZ HUEICHAPAN

DISEÑO DE INTERVALOS DE PREDICCIÓN DIFUSOS EN EVOLUCIÓN

En esta tesis se propone nuevas metodologías para implementar intervalos de predicción en evolución basados en la incorporación de conceptos provenientes del área de aprendizaje en ambientes no-estacionarios y el diseño de sistemas inteligentes en evolución. Esta investigación está dividida en tres etapas: primero, una revisión del estado del arte relacionada al uso de intervalos de predicción para modelar sistemas no-lineales. En segundo lugar, el desarrollo de un diseño de intervalo de predicción difuso en evolución basado en el aprendizaje en ambientes no-estacionarios y una segunda propuesta de intervalos difusos en auto-evolución, la cual surge como alternativa de menor complejidad para la implementación de los intervalos deseados. Ambas propuestas fueron evaluadas en esta tesis a través de pruebas simuladas y experimentales, mostrando resultados prometedores para el modelamiento de sistemas no lineales que presentan cambios en su dinámica. Finalmente, la última etapa consiste en la propuesta de un nuevo algoritmo de detección de fallas basado en intervalos que utiliza la información del ancho del intervalo de predicción para activar las alarmas de detección de fallas. Los resultados experimentales llevados a cabo sobre una planta de intercambiador de calor confirman la utilidad de la última etapa de esta tesis.

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE
DOCTOR EN INGENIERÍA ELÉCTRICA
POR: OSCAR ANDRÉS CARTAGENA VILLALOBOS
FECHA: 2024
PROF. GUÍA: DORIS SÁEZ HUEICHAPAN

DESIGN OF EVOLVING FUZZY PREDICTION INTERVALS

In this thesis, we propose new methodologies to design and implement new evolving fuzzy prediction intervals based on incorporating concepts from the field of learning in nonstationary environments and the design of evolving intelligence systems. This research is divided into three main stages: first, a review of the state-of-the-art regarding the use of prediction intervals for modeling nonlinear systems and the subsequent comparative analysis done for choosing the base interval model for developing the proposals. Second, the development of the evolving fuzzy prediction interval design based on the field of nonstationary environments. In addition, the self-evolving fuzzy prediction interval, is proposed as a less complex alternative for implementing the desired interval. Both proposals were evaluated in this thesis over some simulated and experimental tests (which include modeling a real heat exchanger plant), showing promising results for modeling nonlinear systems that present changes in their dynamics. Finally, the last stage of the proposal consists of a new interval-based fault detection algorithm that uses the information of the prediction interval width for activating the alarms of fault detection. The experimental results carried out over a heat exchanger plant confirm the utility of the last stage of this thesis.

Dedicated to my family and friends.

Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. Doris Sáez Hueichapan for their encouragement and support in the development of this thesis, especially in a difficult context due to the effects of the COVID pandemic.

In addition, I want to express my sincere gratitude to professors Manuel Roveri, Francesco Trovò and Igor Škrjanc who collaborated with me during the development of this thesis, and provided a helpful aid to improve my research.

Moreover, I want to thank all the people that I met during my stay in the Universidad de Chile and during my visiting stays done at the University of Ljubljana, Slovenia, and the Politecnico di Milano, Italy.

Furthermore, I would like to express my gratitude to all research projects that supported the publication of my articles, including the Instituto Sistemas Complejos de Ingeniería (ISCI) ANID PIA AFB230002, the Solar Energy Research Center SERC-Chile ANID/FONDAP/1522A0006 and the ANID/FONDECYT 1220507.

Finally, I would like to express my gratitude to the Agencia Nacional de Investigación y Desarrollo (ANID) under grant ANID-PFCHA/Doctorado Nacional/2020-21200709 which funded my PhD studies the last four years.

Table of Content

List of Tables

List of Figures

List of Acronyms

FLS:	Fuzzy logic systems.
NNs:	Neural networks.
FDS:	Fault diagnosis system.
FL:	Fuzzy logic.
MF:	Membership function.
PSO:	Particle swarm optimization.
ITLBO:	Improved teaching learning based optimization.
TS:	Takagi-Sugeno.
IT2:	Interval type-2.
A2-C0 IT2:	Interval type-2 with antecedents type-2 and consequences type-0.
A2-C1 IT2:	Interval type-2 with antecedents type-2 and consequences type-1.
T2:	Type-2.
FNN:	Fuzzy neural network.
IT2FNN:	Interval type-2 fuzzy neural network.
A1-C1:	fuzzy model with antecedents and consequences type-1.
EIS:	Evolving intelligent systems.
EFS:	Evolving fuzzy systems.
ENFS:	Evolving neuro-fuzzy systems.
FLEXFIS:	Flexible fuzzy inference system.
eTS:	Evolving Takagi-Sugeno.
EFuMo:	Evolving fuzzy model.
IBeM:	Interval-based evolving model.
FDS:	Fauld diagnosis systems.
PINAW:	Prediction interval averaged width.
PICP:	Prediction interval coverage probability.
FPI:	Fuzzy prediction interval.
ARX:	Auto-regressive with an exogenous input.
RMS:	Root mean square evaluation method.
TSK:	Takagi-Sugeno-Kang.
U-FPI:	Proposed fuzzy prediction interval which only includes the updating existing clusters mechanism.
C-FPI:	Proposed fuzzy prediction interval which only includes the cluster creation mechanism.

E-FPI: Evolving fuzzy prediction interval.
SE-FPI: Self-evolving fuzzy prediction interval.
RMSE: Root mean square error.
APRBS: Amplitude-modulated pseudo random binary signal.
FD: Fault detection.

1 Introduction

1.1 Motivation

The use of fuzzy logic systems (FLS) and neural networks (NNs) has proliferated in the literature for modeling systems. Since both types of models are universal approximators that can identify relationships between input and target variables (outputs), they are generally used when the system to be modeled follows nonlinear dynamics [1]. Although some FLS, like the Takagi-Sugeno fuzzy model, and NNs like the recurrent neural networks, exhibit adequate performances when dealing with the modeling of dynamical systems, uncertainty is not typically quantified by these approaches. However, having information on the dispersion of possible future model outputs may be more useful from a decision-making point of view than having models that only provide expected values [2, 3]. With that purpose, prediction interval models have been proposed to address the problem of quantifying prediction uncertainty. A prediction interval establishes a range around the output of the model, representing the uncertainty present in the system. The main motivation for the construction of prediction intervals is to provide information on the future values of a system along with information on its expected uncertainty, allowing multiple scenarios to be considered for the best and worst conditions of that system [4].

Additionally, the modeling of a nonlinear process grows in complexity when the system has a time-variant dynamic. For example, the system may present changes in its parameters, which can occur due to internal factors (replacement of elements/actuators or faults produced) and external influences (changes in the environment of the process). To address the challenge of achieving online data processing for this kind of dynamic process, the concept of evolving intelligent systems has been proposed [5]. Considering the functionality of the prediction interval models based on fuzzy and neuro-fuzzy approaches for modeling nonlinear time-invariant systems with uncertainties [6], the incorporation of concepts from evolving intelligent systems arises as a useful method for obtaining new prediction intervals, which would be able to model time-variant dynamics with their uncertainties.

Finally, for fully monitoring the status of the modeled nonlinear process, the concept of Fault Diagnosis Systems (FDS) can be applied. FDS has been applied in the literature to verify the consistency of real-time information measured from the system [7]. Due to this verification process, early detection and proper response against possible dangerous faults can be achieved.

Based on these points, an explanation of the problem statement considered in this thesis

is given in what follows.

1.2 Problem statement

For a safe, secure, and reliable operation of industrial processes, it is necessary to monitor the nonlinear system's behavior with its uncertainty and have knowledge about changes presented by its dynamic. The importance of monitoring the behavior of the process lies in the fact that the hypothetical presence of faults and abrupt changes in the system dynamics can cause critical situations where the integrity of the process could be compromised [8]. Under this premise, the design and implementation of evolving fuzzy prediction strategies are considered to give the necessary information about the expected system behavior and its uncertainty to the user or controller that handles the process; so it can react on time when unexpected changes appear on the system.

To handle this problem, this thesis proposal is focused on the following topics:

- First, carry out an evaluation of the fuzzy prediction intervals that have been developed in the literature to model nonlinear time-invariant systems with its uncertain behavior, to determine the suitable methods to be included in the algorithms to be proposed. To address this topic, fuzzy prediction interval methods will be tested using some case studies, such as the modeling cases of solar power generation (real data) and generic nonlinear dynamical systems (synthetic data).

- Secondly, to design a new evolving prediction interval to model nonlinear time-variant systems, using the fuzzy prediction interval methods previously evaluated as the basis. This will be approached by incorporating elements previously designed for evolving fuzzy models into the structure of fuzzy prediction interval methods. Additionally, new methods for interval width updating will be studied in this work, in order to achieve an improvement in the online characterization of the uncertainty in a nonlinear time-variant system.

- Third, to evaluate the new evolving fuzzy interval designs. This evaluation will be handled by making a comparison between some classical methods of fuzzy prediction intervals and the proposed evolving intervals when applied to different cases. This is done with the purpose of highlighting the benefits associated with the solutions proposed in this work.

- Lastly, to design an interval-based fault detection algorithm that relies on the estimation of the interval coverage level inspired by the concepts of residual evaluation functions used by fault diagnosis systems. This proposed fault detection algorithm will be tested on a heat exchanger plant affected by external changes introduced in its behavior.

Based on the activities mentioned above, the hypotheses, objectives, and main contributions of this work are presented in what follows.

1.3 Hypotheses

The hypotheses associated with this work are:

- The fuzzy prediction interval methods developed in the literature can successfully handle the modeling of a vast range of nonlinear time-invariant systems with their uncertainties.

- An adaptation of previous fuzzy prediction interval methods can be carried out, incorporating evolving mechanisms that will allow the new interval to model nonlinear time-variant dynamics with uncertainty.

- The new evolving techniques to be proposed can obtain in nonstationary environments performance metrics similar to those achieved by classical prediction intervals during their training process. Thus, the methods to be proposed have to minimaze the interval width while maintaining a certain coverage level, despite the changing uncertainty behavior that a nonlinear time-variant dynamics may present across time.

- With the use of the proposed interval design, it is possible to implement a new interval-based fault detection algorithm that can rely on a decreasing behavior of the interval coverage level during the system's online operation to detect a fault.

1.4 General objective

The general objective of this work is to design, analyze, and validate novel techniques for evolving fuzzy prediction intervals. These evolving fuzzy prediction intervals will be designed considering and integrating concepts from fuzzy prediction interval methods and evolving fuzzy models.

1.5 Specific objectives

To achieve the goal of this thesis, the following specific objectives can be identified:

- To analize and evaluate the performance of fuzzy prediction intervals when modeling nonlinear time-invariant dynamical systems with uncertainties, in terms of the interval width and coverage level.

- To design new methods of evolving fuzzy prediction intervals that are capable of modeling nonlinear time-variant systems with uncertainties.

- To analize and evaluate the performance of the proposed methods in terms of interval width, coverage level, and computational cost of the online algorithm when modeling synthetic data generated from a generic nonlinear time-variant system and real data extracted from a heat exchanger plant.

- To design a new interval-based fault detection algorithm based on the information provided by the proposed interval methods. Then, to analize and evaluate its performance

in terms of precision and accuracy, when real data measured from a real heat exchanger plant is used.

1.6 Main contributions

The contributions of this thesis are:

- Development of a literature review regarding the use of fuzzy prediction intervals for modeling nonlinear dynamics systems with their uncertanties. As part of this literature review, theoretical analysis and empirical evaluation of fuzzy prediction interval methods were carried out using the modeling case of synthetic data and real measurements of solar power generation.

- Proposition of two novel evolving prediction interval methods, which can react and adapt the model structure to the changes occurring in the system. One proposal is based on the offline identification of previous fuzzy prediction intervals, while the second approach relies on a less complex model structure with self-recursive adaptation of its parameters.

- Validation of the proposed solution through a simulated benchmark case and modeling a real dynamical process. This validation was carried out by evaluating the performance metrics previously used to compare previous implementations of fuzzy prediction intervals.

- Proposition of a novel interval-based fault detection algorithm that can detect changes in the modeled system based on the behavior of the interval metrics. In particular, the proposed algorithm relies on the decreasing behavior of the interval coverage level to determine the occurrence of a system fault.

1.7 Publications

The articles published during the development of this work are the following:

Journals

- **O. Cartagena**, F. Trovò, M. Roveri and D. Sáez, "Evolving Fuzzy Prediction Intervals in Nonstationary Environments," in IEEE Transactions on Emerging Topics in Computational Intelligence, (Early Access), doi: 10.1109/TETCI.2023.3296486 (Journal Impact Factor 2022: 5.3 - Q2 Computer Science).

- **O. Cartagena**, M. Ožbot, D. Sáez, I. Škrjanc, "Evolving fuzzy prediction interval for fault detection in a heat exchanger", Applied Soft Computing, vol. 145, 110625, 2023, doi: 10.1016/j.asoc.2023.110625 (Journal Impact Factor 2022: 8.7 - Q1 Computer Science).

- A. Endo, S Parra, **O. Cartagena**, D. Sáez, C. Muñoz, and J. I. Huircan. "Energy–Water Management System Based on MPC for a Greenhouse in a Mapuche Indige-

nous Community" Applied Sciences 13, no. 8: 4734, 2023, doi: 10.3390/app13084734 (Journal Impact Factor 2022: 2.7 - Q2 Engineering, Multidisciplinary).

- **O. Cartagena**, S. Parra, D. Muñoz-Carpintero, L. G. Marín and D. Sáez, "Review on Fuzzy and Neural Prediction Interval Modelling for Nonlinear Dynamical Systems," in IEEE Access, vol. 9, pp. 23357-23384, 2021, doi: 10.1109/ACCESS.2021.3056003. (Journal Impact Factor 2022: 3.9 - Q2 Computer Science).

Conferences

- L. Rojas, J. Ocaranza, **O. Cartagena**, D. Sáez, L. Daniele and C. Ahumada, "Robust Energy-Water Management System with Prediction Interval Based on Deep Learning," 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 2023, pp. 1-9.

- A. Endo, **O. Cartagena**, J. Ocaranza, D. Sáez and C. Muñoz, "Fuzzy and Neural Prediction Intervals for Robust Control of a Greenhouse," 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Padua, Italy, 2022, pp. 1-8.

- A. Endo, **O. Cartagena**, D. Sáez and D. Muñoz-Carpintero, "Predictive Control based on Fuzzy Optimization for Multi-Room HVAC Systems," 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2020, pp. 1-8.

- D. Muñoz-Carpintero, S. Parra, **O. Cartagena**, D. Sáez, L. G. Marín and I. Škrjanc, "Fuzzy Interval Modelling based on Joint Supervision," 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2020, pp. 1-8.

- T. Lara, **O. Cartagena**, S. Céspedes and D. Sáez, "Robust Model-based Predictive Control for a Cooperative Cycling Cyber-physical System," 2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), 2019, pp. 1-6.

1.8 Thesis outline

This document is organized as follows: Chapter 2, Literature Review, presents a review of the state-of-the-art for the topics of fuzzy prediction intervals and evolving intelligent systems.

Chapter 3, Proposed Methodologies presents the previous fuzzy prediction interval models considered as the basis of this work. Additionally, this chapter continues with the presentation of the details about the proposals related with the novel evolving fuzzy prediction interval design and the interval-based fault detection algorithm.

Simulated and experimental results of the evaluation of fuzzy prediction intervals, the implementation of the proposed evolving intervals, and the fault detection algorithm are shown in Chapter 4, Cases of study.

Finally, Chapter 5 presents the main conclusions of this thesis and proposes some directions to conduct future work.

2 Literature Review

This chapter shows a review of the state-of-the-art of fuzzy prediction intervals, evolving systems, and model-based fault detection systems, which are the topics to be covered in this research. First, an explanation of the fuzzy prediction intervals is presented, showing how these methods are used to handle the uncertainty introduced during the modeling of a nonlinear dynamical system. Then, this chapter continues with the description of recent works related to the implementation of evolving fuzzy and neuro-fuzzy models, highlighting those methods which have been applied for the interval modeling of time-variant dynamics of nonlinear systems. Finally, this chapter describes some related approaches previously developed in the literature for implementing fault detection algorithms based on models identified from the monitored system. This chapter concludes with a brief discussion to link the methods reported in this literature review, with the proposal to be developed in this research work.

2.1 Fuzzy prediction intervals

Fuzzy logic systems (FLS) are generally known in the literature as universal approximators that can identify relationships between input and target (output) variables and are generally used when the system to be modeled follows nonlinear dynamics [1]. The importance of FLS lies in the fact that can be used to solve a broad range of problems, from forecasting to classification, to control, etc. (see as examples the works presented in [9, 10, 11]). These models are based on fuzzy logic (FL) which is a method that resembles human reasoning, imitating the way of decision-making in humans.

In the literature, rule-based fuzzy models are mainly considered in the problem of time-series forecasting. According to [12], these rule-based models correspond to fuzzy systems that are composed of four main stages: fuzzification, inference, rule base, and defuzzification.

In this type of model, the Fuzzification stage is the process of converting the crisp input to a fuzzy value, performed by the use of the information of fuzzy sets received from a clustering algorithm. Here, the membership functions (MF) of the input data are calculated for the different fuzzy sets, using one of the several types of curves reported in the literature (Gaussian, triangular, and trapezoidal MFs are the most commonly used) [13]. Then, in the rule-based stage, the expert information available from the operation of the system is formulated as a finite number of local models (also denoted as rules of the FLS). Subsequently, the inference is where the fuzzy decisions of the model are produced, from the several rules

available. Finally, in the defuzzification stage, the set of fuzzy decisions corresponding to the rules of the FLS is translated into a single crisp value. Several heuristic defuzzification methods can be applied here, such as the weighted integral (or sum) of the outputs of the different rules with their respective activation degrees [12], or the center-of-area method, which takes the center of gravity of the fuzzy set to obtain the crisp values [13].

Among the rule-based fuzzy models reported in the literature, the formulation presented by Takagi and Sugeno in [14] arises as one of the most used structures for rule base fuzzy modeling. Under this formulation, the fuzzy model is represented by the weighted sum of several linear models.

In order to handle the effect of uncertainties at the moment of performing the model identification, prediction interval modeling is considered a useful tool [15]. The advantage of prediction intervals is their usefulness in representing the uncertainty behavior, using a structure similar to the model previously used to approximate the dynamics of a system. Additionally, it is possible to establish ranges for the future measurements of the system with the use of fuzzy prediction intervals, thus providing helpful information for the implementation of robust controllers and decision-making environments.

Due to the utility of the fuzzy models based on the formulation presented by Takagi-Sugeno for time series forecasting and nonlinear system modeling, the current section will focus on the prediction intervals based on this kind of model.

In the specialized literature, several methods based on Takagi-Sugeno fuzzy models have been proposed in order to obtain a prediction interval. In this work, those methods are classified by the kind of procedure followed during their construction.

The first category corresponds to the Direct Methods which construct the fuzzy prediction intervals at the same time when the model identification is performed. The second category called Sequential Methods is made by the intervals which are constructed based on a model previously identified. The intervals of the first category are briefly explained below.

2.1.1 Direct methods

The direct methods correspond to the fuzzy prediction intervals which are constructed in parallel with the model identification.

Min-max method

The main idea of this kind of method was introduced in [16], where a min-max method is proposed for identifying the bounds of the fuzzy interval. Then, the interval is obtained by the identification of two different fuzzy functions, called the upper and lower functions ($\overline{f}(z)$ and $\underline{f}(z)$, respectively), which are found by solving the following optimization problems:

$$\min_{\overline{f}} \max_{z_i \in Z} |y_i - \overline{f}(z_i)| \quad \text{subject to} \quad y_i - \overline{f}(z_i) \leq 0, \quad \forall i, \tag{2.1}$$

$$\min_{\underline{f}} \max_{z_i \in Z} |y_i - \underline{f}(z_i)| \quad \text{subject to} \quad y_i - \underline{f}(z_i) \geq 0, \quad \forall i, \tag{2.2}$$

where z_i are the samples of the training dataset Z.

The min-max method has been widely applied to the fault detection problem over various types of systems. In [17], this method was used to detect the fault in a Motor-Generator Plant, in [18, 19, 20] it was used for formulating a fault-detection system for a nonlinear system with uncertain parameters, and in [21] this method was used for the construction of a belief rule-based model for the identification problem of uncertain nonlinear systems. Also, the fuzzy intervals based on this min-max method were used for the implementation of a robust control strategy [22, 23, 24] and were applied to the estimation of time-series related to renewable energy systems (photovoltaic, wind and battery power measurements) [23, 24, 25].

Method based on interval-valued data

In [26] another variant of fuzzy interval models was proposed, based on the use of interval arithmetic for the modeling of an interval-valued output. Under this formulation of fuzzy interval modeling, the modeled output and the parameters of the rules are defined by a center and a radius.

The model resulting from the interval arithmetic was applied in [26] to the identification of a nonlinear system. A similar strategy was carried out in [27, 28], where a fuzzy model is identified for interval-valued data characterized by confidence intervals obtained from an electro-mechanical throttle valve using the Chebyshev's inequality.

Method based on fuzzy numbers (direct version)

In [29], a third type of fuzzy interval is proposed, based on the idea of interval fuzzy numbers [30]. In [29], the parameters of the local linear models considered for each rule are defined by interval fuzzy numbers, i.e. they are defined by mean and spread values. Thus, each rule results in two different outputs, which are associated with the bounds of the local interval. Then, the bounds of the prediction interval resulting from this method are computed by performing the weighted sum of the local intervals available from the different rules, similar to what is done in the defuzzification stage of the Takagi-Sugeno models.

For obtaining the interval fuzzy model, the mean and spread values for the parameters must be identified by solving an optimization problem, which minimizes the interval width and the difference between the measured coverage index with its target value. Because the optimization is usually nonlinear, the problem is solved in [29] by using Particle Swarm Optimization (PSO) and Improved Teaching Learning Based Optimization (ITLBO). The interval based on fuzzy numbers was applied only to load forecasting in a microgrid [29].

Method based on type-2 Takagi & Sugeno models

In the task of considering the effect of uncertainties in the model identification, the type-2 fuzzy models presented in [31] arise as an appropriate extension of the fuzzy rule-based models. In this case, type-2 fuzzy sets, which were originally introduced in [32], are used for the antecedents of the rules. Thus, a type-2 fuzzy set A has a membership function defined by two different values, corresponding to its upper and lower limits ($\overline{\mu}_A(x)$ and $\underline{\mu}_A(x)$

respectively).

According to [33], four kinds of interval type-2 (IT2) fuzzy systems have been developed in the literature, following the TS formulation: the unnormalized A2-C0 IT2, the normalized A2-C0 IT2, the unnormalized A2-C1 IT2, and the normalized A2-C1 IT2. In this nomenclature, the first acronym A2 indicates that the Takagi-Sugeno fuzzy system is defined by type-2 antecedents, i.e. type-2 fuzzy sets are used in the fuzzification stage. Meanwhile, the second acronym indicates whether the consequences (also denoted as the parameters of the local models) are type-0 or type-1. If that acronym is C0, then type-0 consequences are used, i.e. the parameters of the local models associated with each rule are defined by crisp values. On the other hand, if that acronym is C1, then type-1 consequences are used, i.e. the parameters of the local models associated with each rule are defined by type-1 fuzzy numbers (as mentioned before, that means the parameters are defined by mean, and spread values).

The unnormalized variants of these methods compute the interval bounds directly as the weighted sum of the local intervals, following a similar approach as done in the defuzzification stage of the classic Takagi-Sugeno formulation. On the other hand, the normalized variants include a defuzzification of higher complexity for the computation of the interval bounds.

Despite the fact that the use of type-2 fuzzy models was focused in the modeling of a crisp output (see as example [34, 35, 36, 37, 38]), due to the advantage of its model structure that incorporates information about the signals uncertainties, this kind of models also has been used to implement prediction intervals. These prediction intervals based on T2 TS fuzzy models have been mainly applied as prediction intervals in several cases related to the modeling of renewable energy systems; such as modeling of solar power [39, 40], wind power forecasting [41, 42, 43, 44], and characterizing the loads in a microgrid [43].

Method based on fuzzy neural networks

Despite the use of type-2 fuzzy sets in the formulation of IT2 TS fuzzy models, in reality its use in the literature is mainly focused on the construction of prediction intervals based on fuzzy neural networks (FNN).

Following the main ideas of IT2 TS models, the fuzzy neural networks can be implemented by the use of type-2 fuzzy sets in the membership layer (this is analog to the use of type-2 fuzzy sets as antecedents in the IT2 TS model). Also, the rule layer of the FNN can be defined by type-1 fuzzy numbers (similar to use of type-1 consequences in the IT2 TS models). Finally, the bounds of this prediction interval are then given by the values of the neurons in the output layer.

This structure is denoted in the specialized literature as the Interval Type-2 Fuzzy Neural Network (IT2FNN), and has been applied in several cases of modeling systems. For example, the IT2FNN was used for the modeling of chaotic time series and nonlinear systems in [45, 46, 47, 48, 49, 50]. Other specific applications include the uses for the modeling of an antilock braking systems [51] and for wind power forecasting [52]. Additionally, in [53] an overview of this kind of prediction intervals used for the prediction of chaotic time series is presented.

Joint supervision method

The Joint Supervision Method is based on two conflicting goals: predictions must be as close as possible to the expected (crisp) value, and the prediction interval must be wide enough to comply with imposed coverage probability restrictions. The version of this method that uses fuzzy models is proposed in [54], where an adaptation of the joint supervision method, previously developed for neural networks in [55], is performed.

In [54] a fuzzy model with 3 outputs is used, where two of them are associated with the upper and lower interval bounds, while the third output corresponds to the expected target value. The training procedure is performed by solving an optimization problem with a different cost function for each output. First, all model outputs share a cost component to minimize prediction error. On the other hand, the outputs associated with the interval bounds introduce an additional term in the cost function that is applied only when data points fall outside the predicted interval. This version of the method is applied in [54] for temperature forecasting only.

With the direct methods based on fuzzy models already covered in this review, now the intervals of the second category aforementioned (the sequential methods) will be explained below.

2.1.2 Sequential methods

The sequential methods correspond to designing intervals in a subsequent step to the model identification. The whole process of prediction interval construction is considered as a sequence formed by the model identification stage and the subsequent spread identification.

Covariance method

In [56], the upper and lower bounds that define a prediction interval are constructed based on the error covariance of each rule of the Takagi-Sugeno fuzzy model.

This method of fuzzy interval has been applied for the forecasting of renewable generation, and demand data from an installed microgrid [57, 58, 59] and for the robust control of a solar collector field [60]. Also, this method has been used for the identification of intervals for traffic measurements [61], for a robust predictive control design applied to a climatization system [62], a fault detection system for an aircraft [63], and for the implementation of an indoor localization algorithm [64].

Method based on interval fuzzy number for the output uncertainty

In [39], an interval is proposed from a Takagi-Sugeno fuzzy model. In that case, after identifying the model, a single interval fuzzy number is added in order to approximate the model uncertainties present in the output of the system. That interval was later denoted in [40] as A1-C1 Takagi-Sugeno-Kang, because it is defined by type-1 antecedents and consequences. The intervals obtained with this method have been applied for the forecasting of solar power generation in [39, 40].

Method based on fuzzy numbers (sequential version)

In [2], another variant of fuzzy interval is proposed, following a similar idea of the method presented in [29]. In this version of the method, the bounds of the fuzzy interval are still given for each rule by the use of mean and spread values for defining the parameters of the consequences. The main difference with the direct version is that at the moment of performing the identification, the mean values of the parameter are known in advance. Then, only the spread values are obtained from the solution of an optimization problem which minimizes the interval width and the difference between the measured coverage index and its target value.

This interval method based on fuzzy numbers was applied to the forecasting of the load in a microgrid and the energy consumption in some residential dwellings [2].

2.1.3 Comparative analysis of the methods

In order to compare the performance and applicability of each of the above mentioned methods, it is necessary to observe the main characteristics that distinguish each model, such as the methodology used to generate interval outputs, the number of parameters, the identification procedure, and the assumptions on the behavior of uncertainty. To facilitate this analysis, a summary of the main characteristics of each interval is shown in Table 2.1 for direct methods and in Table 2.2 for sequential methods, based on the comparative analysis developed in [6]. In these tables, when specifying the number of parameters of each method, n_y and n_u specify the number of inputs of the model associated with past outputs of the system (number of regressors for the output signal) and past realizations of an exogenous signal (number of regressors for the exogenous input signal), respectively. Also, r stands for the number of rules considered in the fuzzy model.

In the fuzzy prediction interval methods, the number of parameters is affected by the structure of the base model. To perform a uniform quantification of this variable in Tables 2.1 and 2.2, an assumption is taken first. For the antecedents of the fuzzy models, fuzzy sets with Gaussian membership degrees are assumed to be used, thus for each input variable, the mean and variance of the Gaussian distributions are considered as parameters. In summary, each local model has its own set of parameters for antecedents and consequences; thus, the total quantity of parameters to be considered for both antecedents and consequences of the fuzzy model increases proportionally with the number of rules.

It is important to note that, due to the different methodologies used by each model to quantify its computational cost, this variable was indirectly handled in Tables 2.1 and 2.2 according to three criteria: the number of additional parameters introduced to the model structure by the interval method (the computational cost increases with the number of parameters), the type of optimization method that has to be used for parameter identification and the number of times a new model has to be trained to converge to an optimal solution (the computational cost increases with the number of models and training repetitions).

Based on these tables, the direct methods tend to have a higher computational cost and number of parameters than sequential methods. This tendency can be explained since sequential methods rely mostly on a previously trained base model, so the training procedure is only applied for obtaining those parameters which are directly involved in the calculation

Table 2.1: Characteristics of the fuzzy prediction intervals based on direct methods.

Type of Interval	**Characteristics**
Min-Max Method [16]	- Method based on two TS models - Antecedents defined by $2r(n_y + n_u)$ parameters - Consequences defined by $2r(n_y + n_u)$ parameters - Model training performed by a nonlinear optimization method - No differences between internal and external sources of uncertainty
Interval-Valued Data [26]	- Method based on one TS model and interval arithmetic - Antecedents defined by $2r(2n_y + n_u)$ parameters - Consequences defined by $r(2n_y + n_u)$ parameters - Model training performed by a nonlinear optimization - Uncertainty of the output signal used as input of the model
Fuzzy Numbers (direct version) [29]	- Method based on one TS model - Antecedents defined by $2r(n_y + n_u)$ parameters - Consequences defined by $2r(n_y + n_u + 1)$ parameters - Model training performed by nonlinear optimization - No differences between internal and external sources of uncertainty
Type-2 TS models [33]	- Method based on one type-2 TS model - Antecedents defined by $3r(n_y + n_u)$ (assumption of $2r(n_y + n_u)$ membership functions with shared centers) - Consequences defined by $r(n_y + n_u)$ parameters in unnormalized A2-C0 model (+2 in normalized version) - Consequences defined by $2r(n_y + n_u)$ parameters in unnormalized A2-C1 model (+2 in normalized version) Model training performed by a nonlinear optimization - Effects of internal and external sources of uncertainty can be handled separately
IT2 Fuzzy Neural Networks [45]	- Method based on one fuzzy neural network model - Antecedents defined by $3r(n_y + n_u)$ parameters (similar assumptions of type-2 TS models) - Consequences defined by $2r(n_y + n_u)$ parameters - Model training performed by nonlinear optimization - Effects of internal and external sources of uncertainty can be handled separately by the different layers of the network
Joint Supervision Method [54]	- Method based on multiple identification of three TS models (varying one hyper-parameter) - Consequences defined by $3r(n_y + n_u)$ parameters - Model training performed by nonlinear optimization method - No differences between internal and external sources of uncertainty

* n_y is the number of regressors for the output signal and n_u is the number of regressors for the exogenous input signal

Table 2.2: Characteristics of the fuzzy prediction intervals based on sequential methods.

Type of Interval	Characteristics
Covariance Method [56]	- Method based on one previously identified TS model - Interval defined by 2 tuning parameters - Model training performed by a linear optimization method - No differences between internal and external sources of uncertainty (effects can be separated under some optional assumptions)
Fuzzy Number for Output Uncertainty [39]	- Method based on one previously identified TS model - Interval defined by 2 additional parameters for antecedents and 2 additional parameters for consequences - Interval training performed by a fuzzy identification - Only external uncertainty is considered
Fuzzy Numbers (sequential version) [2]	- Method based on one previously identified TS model - Interval defined by $2r(n_y + n_u + 1)$ additional parameters for consequences - Model training performed by a nonlinear optimization - No differences between internal and external sources of uncertainty

* n_y is the number of regressors for the output signal
and n_u is the number of regressors for the exogenous input signal

of the prediction interval. Instead, direct methods incorporate both the predictive model training and the interval model training into a single procedure, which can introduce some complexities (for example, fuzzy methods that consider larger quantities of parameters by including the concepts of type-2 fuzzy sets).

Another aspect that can be noted is that the selection of the most suited interval model for any experiment depends on the characteristics of the application on which it will be used. Variables such as computational time available, uncertainty behavior (how many assumptions can be made without losing too much performance), and interval informativeness (how much useful information the interval provides) may be the most relevant when making this decision.

From a general point of view, the type-2 fuzzy models and IT2 fuzzy neural networks stand out as the best-suited methods for the identification and differentiation of the effects of internal and external sources of uncertainty. This is due to the structure of these models, where part of the external uncertainty (mainly those associated with external inputs) can be represented by the use of type-2 fuzzy sets in the antecedents of the rules, while the internal uncertainty of the system can be handled by both the rest of the type-2 antecedents and the type-1 consequences. However, in the specific case of this work, the higher complexity of these models could be a disadvantage for the later development of the proposed research. Due to that, a further analysis focused on the applicability of the methods is performed, identifying in which cases it is convenient to use each method. Below are included some general comments regarding the applicability of each method:

1. **Min-Max method [16]:** The more recent interval models outperform the capacities of this method, such as the fuzzy numbers and the joint supervision methods. However, this method is still useful for cases where the uncertainty has a strong and persistent effect over time.

2. **Interval-Valued Data [26]:** Its feasibility is restricted to applications where the dynamical system to be modeled presents interval-valued data as training inputs.

3. **Fuzzy Numbers Method (direct version) [29]:** Even though this method shares most of the advantages of its sequential counterpart, the training process of this version results more complex and computationally demanding since base model and interval identification are performed simultaneously.

4. **Type-2 TS models and IT2 Fuzzy Neural Networks [33, 45]:** Both methods are suitable for applications with complex system dynamics and uncertainty behaviors. The structure of these models allows an exhaustive characterization of uncertainty in most system dynamics. However, this comes at the disadvantage of an elevated computational cost for its training procedure.

5. **Joint Supervision Method [54]:** Suitable for applications showcasing complex uncertainty behavior where the number of parameters and computational time is not a concern. This method works well in most dynamic systems, but in order to obtain good results, the three models which define the interval structure must be identified multiple times due to the search process of its hyperparameter.

6. **Covariance Method [56]:** This method is suitable for applications that require solutions with a low computational cost, where there are no external sources of uncertainty and system uncertainty can be reasonably approximated as homogeneous and Gaussian.

7. **Fuzzy Numbers for Output Uncertainty [39]:** The simplicity of the interval structure only allows its applicability to systems with additive external uncertainty.

8. **Fuzzy Numbers Method (sequential version) [2]:** The interval structure of this method is versatile and can be applied to different types of systems and several uncertainty behaviors. Also, due to the use of a previously identified model, its training process does not increase in complexity.

Based on these comments, the type-2 TS models and the joint supervision stand out as the most appropriate methods to characterize any type of nonlinearities. On the other hand, the sequential method based on fuzzy numbers arises as an important candidate to be used in this research work due to its simpler structure and versatility to be applied over a wide range of nonlinear systems with uncertainty behaviors. Considering this, the sequential version of the fuzzy numbers method, the joint supervision, and the type-2 fuzzy models are the selected methods that will be evaluated in the first stage of this thesis to determine which method will be used as the basis of the new prediction interval designs.

This part of the literature review was presented in the journal paper:

- **O. Cartagena**, S. Parra, D. Muñoz-Carpintero, L. G. Marín and D. Sáez, "Review on Fuzzy and Neural Prediction Interval Modelling for Nonlinear Dynamical Systems," in IEEE Access, vol. 9, pp. 23357-23384, 2021, doi: 10.1109/ACCESS.2021.3056003. **[Journal Impact Factor 2022: 3.9 - Q2 Computer Science - published]**.

With the prediction intervals based on the fuzzy model already covered, the next section continues with the presentation of recent works related to the implementation of evolving models based on fuzzy approaches and state-of-the-art of evolving intervals.

2.2 Evolving systems

The problem of processing and extracting knowledge from data streams grows in complexity when the system to be modeled presents time-variant dynamics. An example of a time-variance phenomenon is the change in system parameters. This situation can occur due to different factors, e.g., the replacement of elements/actuators, the occurrence of faults during the online operation of the system, and changes in the environment where the process takes place. To address the challenge of achieving online data processing for dynamic processes, the concept of evolving intelligent systems (EIS) has been proposed and developed in the literature [65].

EIS is based on previous fuzzy and neuro-fuzzy techniques with a model structure that allows better interpretability of the extracted information and can include the functionality of evolving based on the changes presented by the incoming data. The EIS usually has a higher level of adaptation when compared with conventional adaptive systems from control theory. Additionally, the EIS presents some differences with the classical identification theory, machine learning, and statistical learning methodologies, where system processes are usually assumed to follow a Gaussian distribution. Thanks to the universal approximation characteristic of fuzzy models, combined with the additional characteristic of evolving in time, these tools can represent highly non-linear and non-Gaussian processes. Due to the scope of this thesis, below is presented a review of the state-of-art of fuzzy and neuro-fuzzy approaches of EIS, applied to the specific cases of system identification, regression, and classification.

2.2.1 Evolving fuzzy and neuro-fuzzy models

In recent years, several evolving methods have been developed based on fuzzy models, neural networks, and hybrid neuro-fuzzy approaches. Furthermore, according to [66, 67], fuzzy systems have been widely used in the EIS framework to implement the denoted evolving fuzzy systems (EFS), thanks to the interpretability that ruled-based fuzzy models usually have. In this field, [66, 67] proposed the combination of EFS with the field of neural networks to design evolving neuro-fuzzy systems (ENFS).

These kinds of methods have been recently reviewed in a survey paper [66] that covers several methods implemented for system modeling (including clustering, regression, and model identification) and data classification. Due to the importance of system modeling for the objectives of this thesis (including monitoring and controlling systems), the following brief review focuses on the evolving methods developed for system modeling.

According to [66], there are several manners to classify the evolving fuzzy and neuro-fuzzy models, depending for example, on the evolving clustering algorithm used, the type of its base model, or its ability of adaptation. In this work, the classification based on the adaptability of the models is used below to describe the types of evolving models.

Based on the ability of adaptation, three categories are identified for the models which adapt their behavior to the system changes. The first category corresponds to the models which have a specific structure that is obtained by performing offline training, and only the model parameters are adapted. Because the model structure does not present any changes

over time, these methods are called adaptive methods, and according to [66], this category is not considered evolving. One example of this kind of method is the generalized adaptive neuro-fuzzy inference system [68].

The second category corresponds to the incremental methods, which consist in models that consider the addition of new rules (fuzzy case) and/or new neurons (neural and neuro-fuzzy cases) as part of the updating mechanisms. However, this type of model does not have a mechanism to remove old and useless rules or neurons, so the structure of the model only increases with time. The most relevant models which form part of this category are the resource-allocating network [69], the self-constructing neural fuzzy inference network [70], the dynamic evolving neural-fuzzy inference system [71], the flexible fuzzy inference system (FLEXFIS) [72], and the evolving Takagi-Sugeno (eTS) [73].

Finally, the third category corresponds to the proper evolving methods, which consider similar updating mechanisms to those contemplated by the incremental methods. The main difference with respect to the incremental methods is the inclusion of an additional mechanism of elimination and merging of rules or neurons. Some of the most important methods of this category are the sequential adaptive fuzzy inference system [74], the self-organizing fuzzy neural network [75], evolving fuzzy neural networks [76], evolving granular modeling (using neural networks, fuzzy models and fuzzy neural networks) [77, 78, 79], dynamic fuzzy neural networks [80, 81], evolving neuro-fuzzy model [82], some variants of the evolving Takagi-Sugeno (simpl_eTS and eTS+) [83, 84], a variant of the flexible fuzzy inference system (FLEXFIS+) [85], the generalized smart evolving fuzzy systems [86], and the evolving fuzzy model (eFuMo) [87, 88].

Considering that the classification of these evolving methods depends on the updating mechanisms used in their designs, it becomes necessary to review the updating mechanisms which usually define the design of the different evolving methods. According to [66], the updating mechanisms considered for the evolving methods are:

Cluster updating

A cluster (or neuron) of a model is made up of several parameters associated with the mean, variance, and coefficients of the local model. This first mechanism of the evolving model consists of the online adjustment of these parameter values, to reduce the model error. Among the evolving methods mentioned here, there are various ways to perform the parameters of the model, which are influenced by the structure of the model.

Cluster addition

The act of adding a cluster or a neuron is one of the most important mechanisms of adaptability of the evolving model. This process corresponds to the initialization of a new cluster, which includes a new local model with its respective parameters. This mechanism is similar to all evolving methods developed in the literature, varying only in the criteria used to decide when this mechanism should be activated and how the model parameter updating is performed. Usually, these decisions can be made based on the model error, distance metrics or other specific conditions.

Cluster merging

This mechanism consists of merging two clusters that begin to present an overlap, because of the previous adaptations applied to the clusters. This step is important for reducing the redundancy of the evolving model. Usually, the decision to merge clusters is taken after verifying the gap between the clusters with a distance metric.

Cluster splitting

When some points that belong to a single cluster show an increase in the error of the model, these points can be separated into a new cluster. By splitting a cluster that starts to present an increase in the model error, the evolving model can be tuned, achieving a finer structure. In this mechanism, a procedure similar to that performed with the cluster addition process must be followed to initialize the two new clusters resulting from the split.

Cluster removal

During the online operation of the model, some clusters will become obsolete, due to the evolution that both the system and the model can present. This mechanism removes from the model old clusters that have not been activated or used for a long period of time. Among the evolving methods, this mechanism is usually implemented by using an index that reflects the obsolescence of the cluster.

These mechanisms define the general explanation of how this type of evolving model usually works. However, the specific details associated with the updating mechanism are not shown here, due to the large number of variations existing for the updating criteria, dependent on each evolving method mentioned in this section. The mechanisms to be considered in this research will be reviewed and explained in more detail in Chapter 3, focusing on the specific methodologies to be used in this work.

2.2.2 Evolving intervals

Considering the vast number of evolving methods that have been developed in the literature for modeling signals and systems, various works have tried to incorporate the previous evolving mechanisms into an interval model. These new models are called evolving intervals and provide both the inherent functionalities of the intervals (establishing ranges for the modeled signal) and the utilities of EIS (adaptability to changes in time due to the dynamics of the system).

An example of these attempts to incorporate the evolving mechanisms into interval models is the interval-based evolving model (IBeM) proposed in [89, 90], which is based on the granular modeling framework [77, 78, 79], and it was implemented to model meteorological and economic time series. In the IBeM, the interval identification is performed online and the parameters of the rule-based model are updated recursively.

In the literature, if the field of fuzzy logic is considered to design an evolving interval modeling method, the type-2 fuzzy systems arise as a suitable option for handling the uncertainties associated with system modeling. The use of type-2 sets and functions in fuzzy models allows the implementation of uncertainty bands around the predicted output based on

the range of values constructed for the antecedent and consequence parameters of the model. The scientific literature provides a large number of works that follow this approach, where evolving type-2 fuzzy models were used for data classification [91] and learning from data streams. Moreover, in the past years, several works proposed the implementation of type-2 fuzzy neural networks for the problem of learning from large data streams. Among the most significant work, we cite [92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106], where the updating of model parameters is handled by adaptive filtering strategies. This methodology for implementing evolving models has shown good results for modeling data streams. However, this advantage is associated with a large requirement of parameters for constructing the model using type-2 fuzzy sets and functions. This doctoral thesis takes inspiration from their designs of evolving type-2 fuzzy neural networks to propose a simpler model structure capable of obtaining a prediction interval as an output.

Finally, other approaches for implementing evolving intervals have been designed in the literature using different base models. For instance, the evolving probabilistic fuzzy modeling [107] bases its algorithm on an interval Takagi-Sugeno model trained over an interval-valued time series, and [108] uses confidence intervals based on statistical quantile theory in combination with evolving generalized Takagi and Sugeno model [86]. In [109], an interval is constructed around a previously evolving fuzzy model. The updating process of the interval width proceeds based on the value of the error's variance. This evolving interval method was applied for modeling a simplified Wiener-Hammerstein-type nonlinear dynamic process and an industrial tank reactor [109].

It is important to note that the works based on EIS are not the only ones that consider updating mechanisms for performing a model adaptation. A complete research area in the literature is aligned to adapt models to changes presented by processes. Below is a brief description of the leading solutions for the learning problem in nonstationary environments.

2.2.3 Active and passive approaches in learning in nonstationary environments

According to [110], two different approaches can be followed for implementing adaptation algorithms in the learning in nonstationary environments area. The learning algorithms can follow an active approach based on a change detector that triggers an updating mechanism. Thus, this type of algorithm only performs a model update when a change is detected in the system. This strategy is also known as "detect & react" approaches [111] and their results are helpful when the process presents abrupt changes in its behavior.

Several mechanisms have been considered in the literature to carry out the algorithm's change detection phase [110, 112, 113, 114], which can be grouped into four prominent families: hypothesis tests, change-point methods, sequential hypothesis tests, and change-detection tests. These mechanisms are based on the study of statistical features extracted from data streams, such as the sample mean, the variance, or the classification error. Then, once a change is detected, the active approaches of learning algorithms proceed with the adaptation mechanism.

On the other hand, learning algorithms that follow a passive approach are based on con-

tinuously adapting the model as soon as a new data sample arrives from the model process. Thus, this approach maintains an up-to-date model that can be adapted to the gradual changes that the process behavior may present. A review of the various learning algorithms classified in this approach has been made in [110].

With all of the evolving systems and learning approaches already mentioned, this chapter proceeds with a review of the final field to be covered by this thesis, the fault detection and diagnosis topic, which was considered as a possible application of the interval proposals. Below, the related works within this additional topic are covered.

2.3 Fault Diagnosis Systems

In many engineering applications, the reliability and safety of industrial processes have acquired great importance over time. These become critical features to consider during the operation of systems when they are subject to anomalies in their behavior and eventual component failures. Due to that, the possibility of detecting and identifying any kind of potential anomalies as soon as possible becomes relevant and useful for avoiding a dangerous situation.

In the literature, a fault has been defined as an unpermitted deviation of at least one characteristic of the system from an acceptable condition, as discussed in [115]. Different kinds of factors can produce the presence of anomalies, for example, malfunction of actuators, loss of data in sensors, a sudden disconnection of system components, etc. Therefore, the faults are often classified as actuator faults, sensor faults, and plant faults [7]. Actuator faults usually cause interruptions in control action, while sensor faults produce substantial measurement errors that can affect the decision-making block where the control signal is obtained. This, coupled with the plant faults, which are usually associated with changes in the system's dynamic, leads to a performance degradation of the operation of the process, with the corresponding risk of eventual damage and collapse of the whole system.

The design of fault diagnosis systems (FDS) arises as a response to this problem, with the main objective of verifying the consistency of the real-time information which is measured from the system to achieve early detection and proper response against these eventual dangerous faults. Implementing fault diagnosis systems comprises three elements: fault detection, isolation, and identification. Fault detection consists of detecting a system malfunction and determining the time when the fault occurs. On the other hand, the fault isolation corresponds to the block, which determines the location of the faulty component inside the system. Finally, fault identification represents the task of obtaining detailed information about the faults, such as their type, shape, and size. The fault diagnosis system has been widely implemented and reported in the past decades, with several surveys papers written from 1976 to date, including [116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135]. Several fault diagnosis methods have been designed according to different classes of approaches. The classification of these methods used in the surveys [7, 136] is briefly explained below.

According to [7, 136] there are four main categories for the type of fault diagnosis systems. These categories are model-based FDS, signal-based FDS, knowledge-based FDS, and other FDS based on hybrid and active methods. The main ideas for each type of fault diagnosis

system are presented next.

2.3.1 Model-Based Fault Diagnosis Systems

The idea of model-based fault diagnosis methods was originated by Beard in [137]. This kind of method requires the availability of an identified model of the system process, which is used to monitor the consistency between the system measurements and the predicted outputs. Taking into account the different techniques available in the literature to characterize system behavior, several model-based methods have been developed over time. The survey [7] distinguishes four subcategories of methods, depending on the kind of methodology followed: the deterministic FDS, the stochastic FDS, the methods developed for discrete systems and those designed for distributed systems.

As a brief explanation, the deterministic FDS corresponds to those methods which use typical observer filters based on the system dynamic. Using as an example a linear state-space model, the observer usually can be expressed as

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + Kr(k), \tag{2.3}$$

where $\hat{x}$ is the estimated state, A, B are the system matrices previously identified and K is the observer gain. The observer residual signal $r(k)$ is given by the difference between the measurement $y(k)$ and the predicted output $\hat{y}(k)$, such that

$$r(k) = y(k) - \hat{y}(k), \tag{2.4}$$

$$\hat{y}(k) = C\hat{x}(k). \tag{2.5}$$

Here, the residual signal $r(k)$ is subjected to the identification errors of the matrices that define the dynamic of the system and the effect of hypothetical faults (including actuator, component, and sensor faults). Having a proper observer gain is one of the main objectives of the design of this kind of method. To achieve this, the residual signal must be sensitive to faults but robust against disturbances; thus, the observer gain is obtained by solving an optimization performed in the frequency domain of $r(k)$. This approach has been applied to different types of models, from linear systems (state-space models) to nonlinear systems (Lipschitz nonlinear systems, Takagi-Sugeno systems, time-delay systems, etc.). For example, in [138] a fault diagnosis sensor was designed for nonlinear systems based on linear matrix inequality, while that formulation was complemented in [139] with the inclusion of concepts from fuzzy logic.

On the other hand, the stochastic approach of this methodology has been mainly developed due to the use of Kalman filters, including its several variants proposed over time, such that extended, unscented, adaptive, and augmented state versions. Fault detection, isolation, and identification are performed in these methods by using the residuals generated from these Kalman filters. As an example, in [140] this kind of Kalman filters-based fault detection and diagnosis methods was studied for a distillation process, in [141] it was applied to a hydraulic servo system, in [142] it was considered to detect faults in sensors / actuators installed in an unmanned air vehicle, and in [143] it was used in a synchronous motor.

The third and fourth subcategories presented in [7] for this model-based technique follow a similar pattern as shown by the deterministic and stochastic approaches, a filter is designed

by taking into account the specific consideration of the system dynamic (such as discrete-event systems and networked distributed systems). Then, the residuals obtained are used as fault symptoms and are processed by the FDS method. A more detailed explanation of these methods is presented in the survey paper [7]. Additionally, a general schematic of this type of method is included in Figure 2.1.

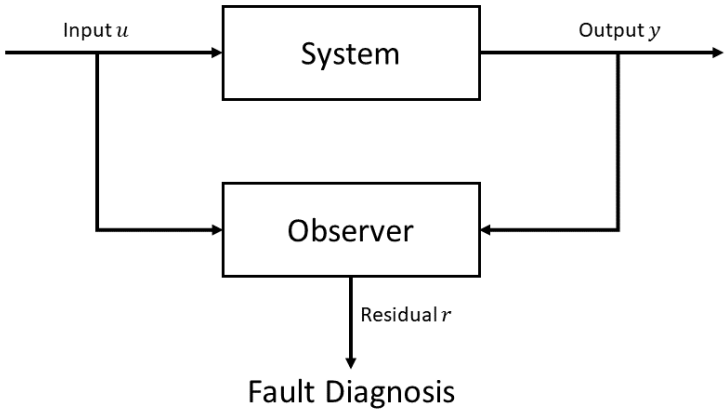


Figure 2.1: Block diagram of a model-based fault diagnosis method. A model or observer is identified for the system dynamics. Then, the fault is detected based on the difference between the measured and observed output, represented by the residual value r.

2.3.2 Signal-Based Fault Diagnosis Systems

The signal-based fault diagnosis systems correspond to the second main category of methods developed in the literature. These methods are based on the fact that the effects of faults can be reflected in measured signals. Therefore, the problem to be solved during the design of these FDS methods is to select a proper feature to be extracted from the measured signal and then perform a symptom analysis of it. Here it is important to remark on the importance of the prior knowledge available of the system (like normal operation data, examples of past faults in similar industrial processes, etc.) to perform an accurate fault diagnosis from the previous symptom analysis.

In this approach, there are several features that can be extracted from measured signals for symptom analysis, which can be either from the time domain (e.g., mean, variance, trends, etc.) or from the frequency domain (e.g. spectrum). This kind of signal-based method has been widely used in the literature in various types of applications, which have been covered in the survey paper [7]. Some examples of these applications are [131] where this approach is used to detect faults in induction motors, in [132] it was applied to electrical motors, and in [134] a review of this type of technique based on time-frequency analysis in machinery operation was performed. Furthermore, [144] implements a diagnosis system for cage rotor induction machines based on the stator current spectrum, while [145] applies a signal analysis for fault diagnosis in a wind turbine.

This type of method is represented by the general schematic presented in Figure 2.2.

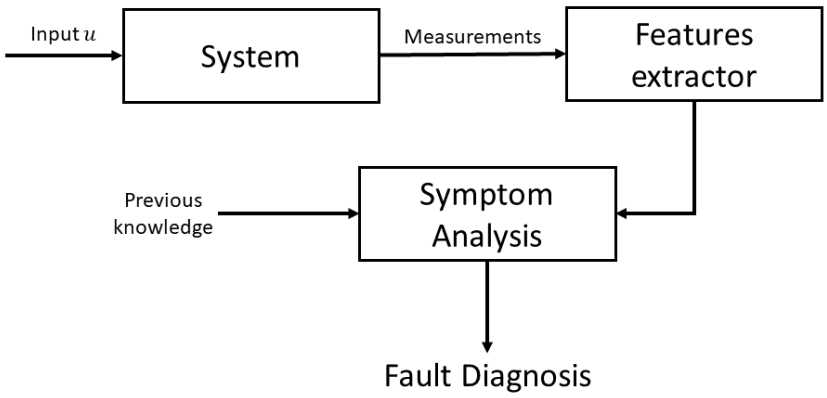


Figure 2.2: Block diagram of a signal-based fault diagnosis method. In this kind of methods, features are extracted from the measured output signal. Then, based on previous knowledge regarding the behavior of the system, the symptom analysis block establish that abnormal values for the features are considered a fault.

2.3.3 Knowledge-Based Fault Diagnosis Systems

The third category of fault detection methods, which is covered in [136], corresponds to knowledge-based methods. In this approach, the FDS does not require prior modeling or knowledge of signal patterns and begins only with large amounts of historical data. Due to that, this kind of method can also be referred to as a data-driven fault diagnosis method.

In the implementation of these methods, the tools available in the area of expert systems and computational intelligence become relevant. Similar to previous methods, fault detection is performed by checking the consistency between the system behavior and the historical data. To achieve that, the use of classifier techniques over the historical data aids in determining different operation modes of the system, which allows the consistency checker to decide whether the new measurement corresponds to a fault or not. Also, the inclusion of classifier techniques in this type of algorithm is helpful in differentiating the various faults that the system can present.

Knowledge-based methods have been widely applied in the literature, from a qualitative approach with the use of expert systems, such as in [146] where it was applied for machine fault diagnosis, while in [147] it was considered to monitor the operation of microprocessors and microcontroller boards. Alternative knowledge-based methods have been implemented from a quantitative approach with the solution of a pattern recognition problem. This pattern recognition problem has been approached in literature from a statistical framework with the use of principal component analysis, applied in [148] for a diesel engine operation, in [149] for a continuous stirred tank reactor and in [150] to check vibration sensors placed on a twin shaft gas turbine. Furthermore, the statistical framework has been approached with the use of partial least squares in [151] for an industrial hot strip mill and in [152, 153] for batch processes. A different approach based on independent component analysis has been implemented in [154] for an electrofused magnesia process, in [155] for solar modules and in [156] for monitoring rolling element bearings. Finally, support vector machines have been used in [157] to diagnose faults of rolling element bearings, induction motors and climatization

machines, in [158] to power transformers and in [159] to complex industrial circuit boards.

Alternatively, this knowledge-based fault diagnosis problem has been addressed from a non-statistical framework with the use of neural networks in [160] for internal combustion engines, in [161] for a nuclear process, in [162] for short circuit diagnosis in a wound-rotor inductor generator, in [163] for diagnosis in inductor motors and in [164] for monitoring power quality disturbances in a power system. The nonstatistical framework has also been addressed by the use of fuzzy logic in [165] for a fault diagnosis in a micro steam power unit and in an abnormal diagnosis of temperature in a chemical process, and in [166] for detecting fault modes in a pulse width modulation inverter.

Finally, a combined framework was proposed, where statistical and non-statistical methods are used jointly [167, 168] for fault detection in induction motors. These methods are represented by the general schematic presented in Figure 2.3.

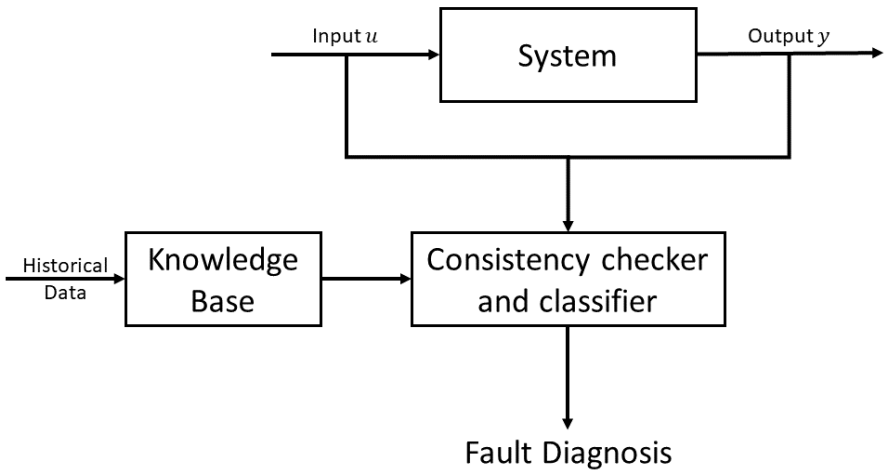


Figure 2.3: Block diagram of a knowledge-based fault diagnosis method. In this method, the fault is detected by the "Consistency checker and classifier" block when the values of the tuple that contains the input and output signals does not show consistency with the historical measurements of the system.

2.3.4 Others Fault Diagnosis Systems

In the literature, other types of methods can be found that do not necessarily fall into one of the previous categories. On the one hand, there are hybrid methods that have been developed by the combination of at least two different fault diagnosis techniques. For example, in [169, 170, 171, 172] some hybridizations of signal-based with knowledge-based methods are presented. In [169] a hybridized method was applied for plastic bearings, while in [170, 171] a diagnosis in induction motors was considered based on stator current espectrum and vibration analysis, respectively. Furthermore, in [172] a fault diagnosis was implemented for magnet synchronous motors based on the analysis of stator currents. Alternatively, in [173] a combination of a model-based method with a knowledge-based method was developed to monitor chemical reactors.

Within the fourth category, there are different types of methods in the literature that take a more invasive approach to the system. This methodology, called active fault diagnosis, consists of the injection of controlled input into the system dynamic to enhance the fault diagnosis by analyzing its response and determining its faulty nodes. This approach has been implemented in [174] for a theoretical reference case of study, in [175] the active fault diagnosis is handled by a closed-loop system with a linear quadratic regulator, in [176] a three-phase pulse width modulated converter is monitored based on the estimation of the equivalent series resistance, while in [177] an active diagnosis is performed for a battery system based on the response of a finite-state machine.

2.3.5 Comparative analysis of the methods

The different methods mentioned in this section have their own advantages and constraints. For example, the model-based methods can perform fault diagnosis with small amounts of data; however, its implementation is subject to the availability of a previously identified model.

On the other hand, the signal and knowledge-based methods are more suitable for the complex dynamic process because they do not require the exact model for its operation. However, the signal-based model lacks comprehensive monitoring of the input variables due to its focus on extracting features from the output signals of the system. While the knowledge-based methods have high computational costs, due to their dependence on large amounts of historical data for their training.

Finally, the hybrid methods share most of the advantages and disadvantages of the previous methods, based on which original methods were combined in the proposed hybridization. While the active diagnosis methods have the advantage of an enhanced fault diagnosis procedure, at the cost of being an invasive approach, which depending on the sensitivity of the system, its implementation could affect the normal behavior of the system.

2.4 Discussion

In this chapter, a review of the state-of-the-art related to the prediction interval modeling was performed, showing the vast number of alternatives available to successfully characterize the behavior of nonlinear systems, while including an uncertainty characterization. In the literature review, a comparison of the characteristics of the methods was made from a theoretical point of view, showing the advantages of the type-2 TS models and the joint supervision method to characterize any type of nonlinearities. Additionally, the sequential method based on fuzzy numbers was highlighted in this review, and due to its simpler structure and versatility, it can be applied over a wide range of nonlinear systems with uncertainty behaviors. Based on these conclusions, the sequential version of the fuzzy numbers method, the joint supervision, and the type-2 fuzzy models were selected as the initial candidates to be considered in the new interval design proposed in this work.

Next, a brief review of evolving systems was performed in this chapter, focusing on the applications of evolving methods that use fuzzy and neuro-fuzzy models as a basis. Considering that most evolving fuzzy and neuro-fuzzy models share similar update mechanisms in

their design, the general concepts of these methods were presented. Here, cluster updating, cluster addition, cluster merging, cluster splitting, and cluster removal were highlighted as the main mechanisms that have been developed for evolving models in the literature. Also, these mechanisms will be considered for the new interval model proposed in this work.

Additionally, from this review of evolving systems, it can be appreciated that the development of evolving intervals based on fuzzy models has not been carried out in depth. This situation may be due to the high popularity of the self-evolving interval type-2 fuzzy neural network. Considering the usefulness of fuzzy prediction intervals for characterizing a wide range of nonlinear systems with uncertainties, their incorporation into developing a new evolving interval design could provide new tools for implementing making-decision frameworks and fault detection analysis.

Finally, in this chapter, a brief review of fault detection methods was carried out. The main inspirations from this framework are the previous work designed for model-based fault detection methods, which relied on an analysis of the model error behavior. Considering that the interval width is usually associated with prediction errors, a new interval-based fault detection algorithm can be designed based on the analysis of the interval performance.

This motivates the realization of the research proposed in this work, which seeks to design new fuzzy prediction interval models, by including the adaptability characteristic that comes from evolving systems. Then, the additional goal of this work is to use the new interval in an applied case for performing fault detection.

The details of the proposed methodology will be described in Chapter 3.

3 Methodology

The methodology of this thesis is composed of the following stages. First, an evaluation of fuzzy prediction interval methods is performed. Then, based on the selected method, the design of a novel evolving fuzzy prediction intervals is developed. The design of this new evolving interval model will include the updating mechanisms presented in Section 2.2 for evolving fuzzy models. Finally, a new interval-based fault detection algorithm, which relies on the performance analysis of the proposed intervals, is designed.

In this chapter, Section 3.1 describes the background of the fuzzy prediction interval methods to be evaluated, while Section 3.2 explains the procedures to take into account for the adjustment of the model during the online operation of the first proposed evolving algorithm. Then, Section 3.3 presents an alternative novel proposal for achieving the self-evolving fuzzy prediction interval. Section 3.4 describes the main background for a model-based fault detection algorithm, and Section 3.5 presents the proposed methodology for achieving the fault detection based on the information provided by the prediction intervals. Finally, Section 3.6 presents a brief discussion about the proposed evolving prediction interval methods.

3.1 Background: Fuzzy prediction intervals

The Takagi–Sugeno formulation for fuzzy systems is considered for representing the base model that will be used to construct the fuzzy prediction intervals. The specific fuzzy model to be used in this work is defined by a set of p inputs $\{z_i(k)\}_{i=1}^{p}$, with $Z(k) = [z_1(k), \ldots, z_p(k)]^T$, and a set of q rules $\{R_j\}_{j=1}^{q}$ each of which has its corresponding output $\hat{y}_j(k)$. A rule in the Takagi–Sugeno model is expressed as follows:

$$R_j : \text{if } z_1(k) \text{ is } F_1^j \text{ and } \ldots \text{ and } z_p(k) \text{ is } F_p^j \tag{3.1}$$

$$\text{then: } \hat{y}_j(k) = \theta_0^j + \theta_1^j z_1(k) + \ldots + \theta_p^j z_p(k), \tag{3.2}$$

where $F_j(Z(k)) \in \mathbb{R}^+$ is the activation degree of each rule. Then, the output of the fuzzy system is:

$$\hat{y}(k) = \sum_{j=1}^{q} \beta^j(Z(k))\hat{y}_j(k), \tag{3.3}$$

where $\beta_j(Z(k)) := \frac{F_j(Z(k))}{\sum_{h=1}^{q} F_h(Z(k))}$ is the normalized activation degree.

Among the fuzzy prediction intervals based on Takagi-Sugeno models, the method based on fuzzy numbers (sequential version), the joint supervision method, and the type-2 fuzzy models, were highlighted in Section 2.1 as the main candidates to be included in the evolving interval design. These methods are briefly described below.

Fuzzy prediction interval based on fuzzy numbers

In this interval method, the parameters of the local models described by Equation (3.2) are defined by fuzzy numbers, i.e., the parameters values θ_p^j are replaced with the range $[\underline{\theta}_p^j, \overline{\theta}_p^j] = [g_p^j - \underline{s}_p^j, g_p^j + \overline{s}_p^j]$. Then, the upper and lower bounds defined for each rule R_j are the following:

$$\overline{y}_j(Z(k)) = \sum_{i=1}^{p} g_i^j z_i(k) + g_0^j + \sum_{i=1}^{p} \overline{s}_i^j |z_i(k)| + \overline{s}_0^j, \tag{3.4}$$

$$\underline{y}_j(Z(k)) = \sum_{i=1}^{p} g_i^j z_i(k) + g_0^j - \sum_{i=1}^{p} \underline{s}_i^j |z_i(k)| - \underline{s}_0^j, \tag{3.5}$$

where g_i^j are the mean values and $(\overline{s}_i^j, \underline{s}_i^j)$ are the spread values of the original parameters θ_i^j. Then, based on the output of the fuzzy model (3.3), the global bounds of the interval are obtained as

$$\overline{y}(k) = \sum_{j=1}^{q} \beta^j(Z(k)) \overline{y}_j(Z(k)), \tag{3.6}$$

$$\underline{y}(k) = \sum_{j=1}^{q} \beta^j(Z(k)) \underline{y}_j(Z(k)). \tag{3.7}$$

The interval width and coverage level are important characteristics to be considered at the moment of performing the interval identification. In [178, 4], the Prediction Interval Normalized Averaged Width (PINAW) and the Prediction Interval Coverage Probability (PICP) are defined respectively as metrics for representing those factors. These metrics are expressed for a dataset of size N by the following expressions:

$$\text{PINAW} = \frac{1}{NR} \sum_{k=1}^{N} \left(\overline{y}(k) - \underline{y}(k) \right), \tag{3.8}$$

where

$$R = \max_k \{y(k)\} - \min_k \{y(k)\} \tag{3.9}$$

is the range of values of the training data, and

$$\text{PICP} = \frac{1}{N} \sum_{k=1}^{N} c_k, \tag{3.10}$$

where

$$c_k = \begin{cases} 1, & \text{if } \underline{y}(k) \leq y_k \leq \overline{y}(k) \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

indicates if the interval given by the bounds $[\underline{y}(k), \overline{y}(k)]$, contains the real measure of $y(k)$.

The interval identification of this method consist in obtaining the spread values ($\overline{s}_r^{(i)}$, $\underline{s}_r^{(i)}$) from solving the following optimization problem:

$$\begin{aligned} \min_{\overline{s}_r, \underline{s}_r} \quad & \text{PINAW} \\ s.t. \quad & \text{PICP} = \alpha, \end{aligned} \tag{3.12}$$

where $\alpha \in (0, 1)$ is the target value to be obtained by the coverage level on the training dataset.

The equality constraint PICP $= \alpha$ is a hard constraint, and due to the nonlinear characteristic of the optimization problem in Equation (3.12), it might be difficult to solve with the typical optimization algorithms in the literature. With the purpose of relaxing this equality constraint, a version of the optimization problem is proposed in [2], which includes that constraint as a barrier function that increases to infinity when the solution is far from fulfilling the imposed condition. Such that, the interval identification can also be defined by the following optimization problem:

$$\min_{\overline{s}_i, \underline{s}_i} \quad J = \eta_1 \text{PINAW} + exp\{-\eta_2 \left[\text{PICP} - \alpha\right]\}, \tag{3.13}$$

where the exponential term tends to have large values when PICP$> \alpha$ because the argument of that function becomes positive. In contrast, that term becomes small when PICP$\leq \alpha$ because the exponential argument is negative. This optimization problem can be solved by using a suitable heuristic algorithm, such as evolutionary algorithms. For example, this problem is solved using Particle Swarm Optimization (PSO) in [2].

Joint supervision method

Based on the formulation presented in [54], this method uses a fuzzy model with 3 different outputs, where two of them are associated with the upper and lower interval bounds, while the third output corresponds to the expected target value. Similarly to what has been done with the fuzzy numbers method, the joint supervision method tries to handle with two conflicting goals: predictions must be as close as possible to the expected value, and the prediction interval must be wide enough to comply with imposed coverage probability restrictions.

The training procedure of this method consists in the implementation of 3 optimization problems, where a different cost function is used for each output ($\overline{y}, \hat{y}, \underline{y}$). First, all model outputs share a cost component (usually the mean square error) included to minimize the prediction error. On the other hand, the outputs associated with the interval bounds introduce a second term to the cost function that is applied only when data points fall outside

the predicted interval. Thus, the cost functions for each output are defined as follows

$$\overline{L} = \frac{1}{N}\sum_{k=1}^{N}(y(k) - \overline{y}(k))^2 + \lambda\frac{1}{N}\sum_{k=1}^{N} ReLU^2(y(k) - \overline{y}(k)), \tag{3.14}$$

$$\hat{L} = \frac{1}{N}\sum_{k=1}^{N}(y(k) - \hat{y}(k))^2, \tag{3.15}$$

$$\underline{L} = \frac{1}{N}\sum_{k=1}^{N}(y(k) - \underline{y}(k))^2 + \lambda\frac{1}{N}\sum_{k=1}^{N} ReLU^2(\underline{y}(k) - y(k)), \tag{3.16}$$

where

$$ReLU(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0, \end{cases} \tag{3.17}$$

and $\lambda \in \mathbb{R}^+$ is a parameter that must be tuned for each individual experiment. The interval identification considered for this method consists in an iterative process, where the minimization of Equations (3.14)-(3.17) is performed several times, by using a suitable solver. Then, a new interval model is obtained for each iteration, using different values for λ. The identification procedure continues changing the value of λ until the interval model complies with the desired interval coverage level.

Type-2 fuzzy models

As mentioned in [33], the interval type-2 (IT2) fuzzy systems can be implemented by using type-2 fuzzy sets as antecedents of the rules. Thanks to this model, two activation degrees can be obtained for each rule, $\underline{\beta}^j(Z(k))$ and $\overline{\beta}^j(Z(k))$, which stands for as the lower and upper limits of the membership functions, respectively. Also, in the A2-C1 variant of this kind of models, the consequences of the rules are defined by the parameter vector θ_p^j that corresponds to an interval type-1 fuzzy number, as similar as done in Equations (3.4)-(3.5).

Based on the limits identified for each rule, there are two ways to compute the global bounds of the A2-C1 IT2 fuzzy model. The unnormalized way considers the following global bounds for the model

$$\overline{y}(k) = \sum_{r=1}^{m} \overline{\beta}^j(Z(k))\overline{y}_j(Z(k)), \tag{3.18}$$

$$\underline{y}(k) = \sum_{r=1}^{m} \underline{\beta}^j(Z(k))\underline{y}_j(Z(k)), \tag{3.19}$$

while the normalized version computes the global limits as follows

$$\overline{y}(k) = \frac{\sum_{r=1}^{U} \underline{\beta}^j(Z(k))\overline{y}_j(Z(k)) + \sum_{r=U+1}^{m} \overline{\beta}^j(Z(k))\overline{y}_j(Z(k))}{\sum_{r=1}^{L} \underline{\beta}^j(Z(k)) + \sum_{r=L+1}^{m} \overline{\beta}^j(Z(k))}, \tag{3.20}$$

$$\underline{y}(k) = \frac{\sum_{r=1}^{L} \overline{\beta}^j(Z(k))\underline{y}_j(Z(k)) + \sum_{r=L+1}^{m} \underline{\beta}^j(Z(k))\underline{y}_j(Z(k))}{\sum_{r=1}^{L} \overline{\beta}^j(Z(k)) + \sum_{r=L+1}^{m} \underline{\beta}^j(Z(k))}, \tag{3.21}$$

where L, U are the switch points given by the Karnik-Mendel Algorithm [179].

Under this formulation, the prediction interval can be finally obtained by solving an optimization problem, similar to what has been performed in (3.12). There, the parameters of the membership functions $\underline{\beta}_r(z_k)$, $\overline{\beta}_r(z_k)$, and the parameters of $\underline{y}_j(Z(k))$, $\overline{y}_j(Z(k))$, are the optimization variables. Considering the high complexity of the model structure described by (3.20)-(3.21), only the unnormalized version of this model is selected to be compared with the previous fuzzy interval methods.

Having explained the fuzzy prediction interval methods, the first step of this work can be performed. This first stage consists of carrying out an evaluation of the aforementioned fuzzy prediction intervals, testing those methods over some applied cases. Then, based on the most suitable method identified from this evaluation, the novel design of an evolving fuzzy prediction interval will be developed. General details of the proposed evolving fuzzy prediction interval algorithm will be presented in the next Section.

3.2 Proposal 1: Evolving fuzzy prediction interval modeling based on learning in nonstationary environments

Modeling a nonlinear process in nonstationary conditions can present difficulties because usually a fixed model previously trained with historical data can become obsolete over time. Hence, a model capable of adapting to the changes in the system is needed. For example, a system may present changes in its parameters, which can occur due to internal factors (replacement of elements/actuators or faults produced) and external influences (changes in the process's environment). To address this challenge, the second main goal of this work is to combine the selected fuzzy prediction interval with the main ideas of evolving fuzzy models [87] and the learning in nonstationary environments.

The field of learning in nonstationary environments aims to define machine and deep learning models capable of tracking and adapting to nonstationary conditions [110]. According to [110], this area is organized into two leading families of algorithms, called the active and passive approaches. In the active approach, adaptive solutions rely on change detection mechanisms aiming at triggering the model adaptation; while in the passive algorithms, the adaptive solution continuously adapts the model over time as soon as new data arrive, even though a change has occurred or not.

The proposed evolving fuzzy prediction interval starts with an offline training phase where the initialization of the interval structure is determined. The algorithm then proceeds with the online phase, where it verifies if the model coverage level computed over the information contained on the newly acquired samples is large enough. During this phase, the algorithm performs an online update of the model already in use, which includes an updating, merging, and removal mechanisms of clusters based on a passive adaptation of the model, and the creation of new clusters based on an active learning approach. Then, the proposed algorithm proceeds to check the performance of the updated interval, where a complete retraining of the model could be considered if the new interval fails this checking test. A summary of the main stages of this proposal is shown in Figure 3.1.

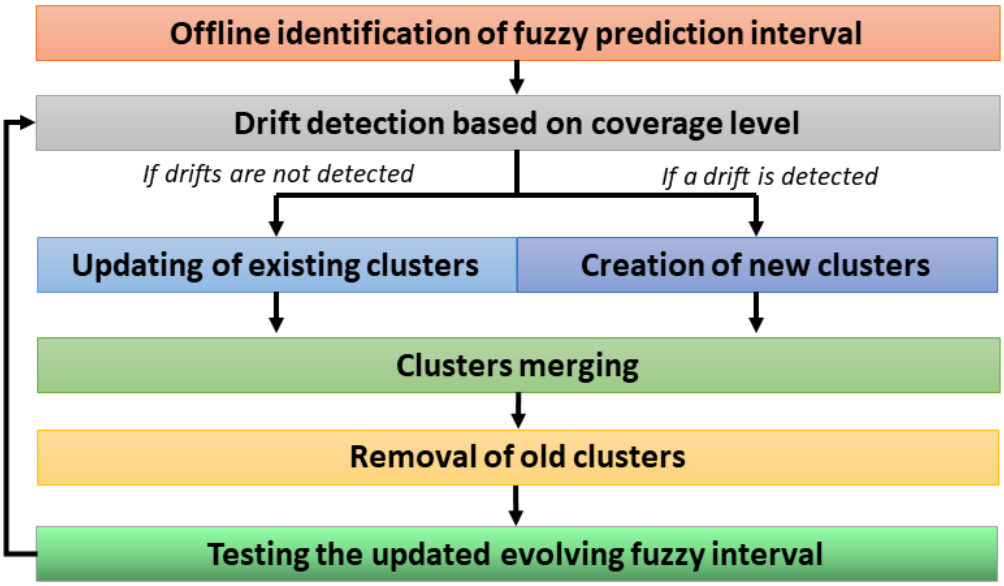


Figure 3.1: Summary of the proposed evolving fuzzy interval algorithm.

The algorithm requires a set of historical data $\mathcal{D} := \{(\boldsymbol{x}(k), y(k))\}_{t=1}^{T}$ that come from the process $P = P_1$. During the first stage shown on Figure 3.1, the algorithm performs an offline classical fuzzy prediction interval identification $FPI(\mathcal{D})$ on the dataset $\mathcal{D}$, generating the cluster set $\mathcal{C}$. After that, the algorithm proceeds with the first online step, which corresponds to the drift detection stage, where an estimate $\hat{\alpha}$ of the coverage level is computed over the training data using a cross-validation approach. This value will be used in the online phase to assess if the model, represented by $\mathcal{C}$ is still providing satisfactory performances. More specifically, the dataset $\mathcal{D}$ is partitioned into T/N blocks $\mathcal{B}_i$ of length N, formally:

$$\mathcal{B}_i := \{(\boldsymbol{x}(h), y(h)) \in \mathcal{D} \text{ s.t. } h \in \{(N-1)i+1, \ldots, Ni\}\}, \tag{3.22}$$

where, for the sake of simplicity, we assume that N is a factor of T and should be large enough to have sufficient samples to properly perform the FPI identification on each $\mathcal{B}_i$, i.e., based on previous works in machine learning, N should be at least ten times the number of parameters to be identified [180]. Training $FPI(\mathcal{D} \backslash \mathcal{B}_i)$, the algorithm estimates the coverage level $\hat{\alpha}_i$ of each partition $\mathcal{B}_i$, formally:

$$\hat{\alpha}_i \ := \frac{1}{N} \sum_{(\boldsymbol{z}(k), y(k)) \in \mathcal{D}_i} \mathbb{1}\{\underline{y}(k) \leq y(k) \leq \overline{y}(k)\}. \tag{3.23}$$

The overall coverage level $\hat{\alpha}$ is computed as follows:

$$\hat{\alpha} := \frac{N}{T} \sum_{i=1}^{T/N} \hat{\alpha}_i. \tag{3.24}$$

During this drift detection stage in the online phase ($k > T$) the algorithm uses the currently available model $\mathcal{C}$ to produce predictions $\hat{y}(k)$, as well as the prediction interval

$[\underline{y}(k),\ \overline{y}(k)]$ over the incoming input $\boldsymbol{x}(k)$. Based on the performance of the system in terms of coverage level, in this stage it is decided if the model is updated or the model undergoes a more extensive retraining. For instance, if P does not show drifts in its behavior, the algorithm updates the existing clusters including the information provided by the data last seen. Conversely, if P presents a change in its dynamic, the algorithm considers creating new clusters.

The next steps of the algorithm diagram presented in Figure 3.1 consider the application of the merging and removal mechanism of existing clusters, to maintain an aceptable complexity of the model. The final step of an online iteration considers the testing procedure of the updated interval performance, with the purpose of verifying that the model is still valid for representing the system. If at this point of the algorithm, the updated interval does not meet the expected performance, a retrain of the entire model is performed in this stage, using the same identification procedure previously used in the offline stage. In what follows, we detail all the procedures shown in Figure 3.1.

3.2.1 Drift detection based on coverage level

To detect changes in a system process, a new batch of data $\mathcal{D}_t$ is constructed as soon as a new measurement is obtained, based on the past N values of the process P, following a sliding window approach. Then, for each new instant t, the batch of data is defined as follows:

$$\mathcal{D}_t := \{(\boldsymbol{x}(h), y(h)) \in \mathcal{D} \text{ s.t. } h \in \{k-N+1, \ldots, k\}\}. \tag{3.25}$$

For detecting changes in the process P, the algorithm computes the corresponding estimated coverage level $\hat{\alpha}_t$ as described in Equation (3.23). This value is compared with the one generated by a Bernoulli trial [181] with a parameter of success $\hat{\alpha}$. Formally we compare $\hat{\alpha}_t$ with:

$$\alpha_{i,M} := \frac{1}{N}\sum_{i=1}^{N} z_i, \tag{3.26}$$

where $z_i \sim Be(\hat{\alpha})$ are realizations of Bernoulli random variables with parameter $\hat{\alpha}$. If the difference between this value and the coverage level $\hat{\alpha}_t$ computed from the newly collected set of sample is above a predefined threshold $Th \in \mathbf{R}^+$, the algorithm increase the value of n_α that counts the number of consecutive instants where the desired coverage is not met.

If the count n_α reaches the value of the size of $\mathcal{D}_t$, the algorithm decides that a change occurred in the process dynamics and starts the creation mechanism of new clusters. Otherwise, it resets the counter n_α to zero and proceeds with the updating mechanism of existing clusters. Next, the updating mechanism to be applied in the case that the drift detection stage does not detect a change is explained.

3.2.2 Updating mechanism of existing clusters

The last processed sample $(\mathbf{x}(k), y(k))$ is inserted in the closest cluster according to the Mahalanobis distance [182] measured from the sample to the center of each cluster $\boldsymbol{\mu}_j(k-1)$. A graphical example of this procedure is shown in Figure 3.2, where the sample vector $z(k)$ is compared with the different existing clusters, using the values of its distance metrics from

the center of the clusters. In this example, the sample vector joins the red cluster because it had the lowest distance value d_2 among the existing clusters.

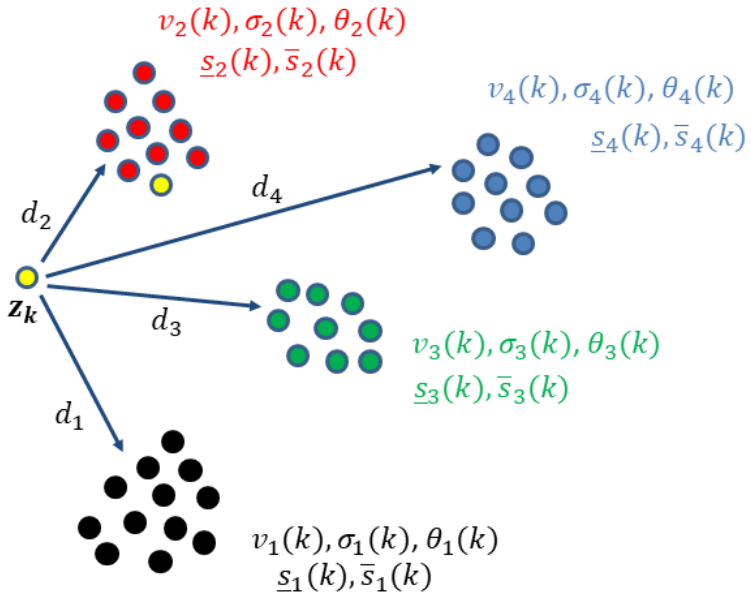


Figure 3.2: Clustering in evolving fuzzy prediction interval models.

Formally, the distance of a sample input $\boldsymbol{x}$ from a cluster C_j is defined as follows:

$$d_j(\boldsymbol{x}(k)) = \sqrt{\boldsymbol{\rho}_j(k)^\top \Sigma_j^{-1}(k-1)\boldsymbol{\rho}_j(k)}, \tag{3.27}$$

where $\boldsymbol{\rho}_j(k) := \boldsymbol{x}(k) - \boldsymbol{\mu}_j(k-1)$, and $\Sigma_j^{-1}(k-1)$ is the covariance matrix corresponding to the j-th cluster. If the minimum (over the clusters) of the distances is less than a given threshold $d_{\max} \in \mathbb{R}^+$, the sample is assigned to the corresponding cluster, and their elements are updated; otherwise, it is considered to be an outlier.

In the former case, the algorithm performs an update of the mean, covariance matrix, and parameters corresponding to cluster j. Formally, using the Welford's algorithm [183], it updates the mean and covariance:

$$\boldsymbol{\mu}_j(k) = \boldsymbol{\mu}_j(k-1) + \frac{\boldsymbol{x}(k) - \boldsymbol{\mu}_j(k-1)}{n_j(k-1)}, \tag{3.28}$$

$$\boldsymbol{\Sigma}_j(k) = \frac{n_j(k-1)\boldsymbol{\Sigma}_j(k-1) + \frac{n_j(k-1)}{n_j(k)}\boldsymbol{\delta}_{\mu_j}\boldsymbol{\delta}_{\mu_j}^T}{n_j(k)}, \tag{3.29}$$

$$\boldsymbol{\delta}_{\mu_j} = \boldsymbol{x}(k) - \boldsymbol{\mu}_j(k), \tag{3.30}$$

$$n_j(k) = n_j(k-1) + 1, \tag{3.31}$$

where $n_j(k-1)$ is the cardinality of the j-th cluster.

The parameters of the j-th cluster are updated using the recursive least-squares method, as shown in [87]. Formally, given the vector of mean parameters $\boldsymbol{g}^j(k) := [g_0^j(k), \ldots, g_p^j(k)]^\top$

for the local rule j, its update is:

$$\boldsymbol{g}^j(k) = \boldsymbol{g}^j(k-1) + Q_j(k-1)\boldsymbol{\psi}_j(k)\left(y_j(k) - \beta_j \hat{y}_j(k)\right), \tag{3.32}$$

where $\boldsymbol{\psi}_j(k) := \beta_j(\boldsymbol{x}(k))\boldsymbol{x}(k)$ and $y_j(k) := \beta_j(\boldsymbol{x}(k))y(k)$ are the projected input and output for the rule R_j, the matrix $Q_j(k)$ is defined as:

$$Q_j(k) := \frac{1}{\lambda_j}\left(Q_j(k-1) - \frac{Q_j(k-1)\boldsymbol{\psi}_j(k)\boldsymbol{\psi}_j(k)^\top Q_j(k-1)}{\lambda_j + \boldsymbol{\psi}_j(k)^\top Q_j(k-1)\boldsymbol{\psi}_j(k)}\right), \tag{3.33}$$

λ_j is the forgetting factor, and $Q_j(0) := \delta \boldsymbol{I}$, where $\delta \in \mathbb{R}^+$.[1]

However, since evolving fuzzy models such as [87] do not include prediction intervals, the update of the corresponding parameters, i.e., $\underline{\boldsymbol{s}_i}$ and $\overline{\boldsymbol{s}_i}$, has not been designed. Based on the constraint for the coverage level considered in the optimization problem in Equation (3.12), we update the spread parameter vectors as follows:

$$\overline{\boldsymbol{s}}^j(k) = \overline{\boldsymbol{s}}^j(k-1) + \eta_{up}\left(\text{PIPC} - \hat{\alpha}\right)|\boldsymbol{x}(k-1)|\overline{\boldsymbol{s}}^j(k-1), \tag{3.34}$$

$$\underline{\boldsymbol{s}}^j(k) = \underline{\boldsymbol{s}}^j(k-1) + \eta_{lw}\left(\text{PIPC} - \hat{\alpha}\right)|\boldsymbol{x}(k-1)|\underline{\boldsymbol{s}}^j(k-1), \tag{3.35}$$

where η_{up} and η_{lw} are new tuning parameters. We remark that Equations (3.34)-(3.35) are inspired by the concept of stochastic gradient descent.[2]

With this, the explanation of the updating mechanism finishes. Next, the creation of new clusters mechanism to be applied in the case that the drift detection stage decides that there is an important change in the system behavior is explained.

3.2.3 Creation of new clusters

The first step of this procedure is to divide the batch $\mathcal{D}_t$ into two different datasets $D^{(1)}$ and $D^{(2)}$, each containing $N/2$ samples. Even in this case, we require that the $N/2$ samples should be sufficient to perform the FPI identification properly, i.e., based on previous works in machine learning, $N/2$ should be at least ten times the number of parameters to be identified [180]. After that, the algorithm applies $FPI(D^{(1)})$ the identification procedure used in the offline stage of the algorithm on the dataset $D^{(1)}$, generating a new set of clusters $\mathcal{G}$ that comply with the expected value of coverage level $\hat{\alpha}$.

After this fuzzy identification is performed, the new set of clusters $\mathcal{G}$ is incorporated into the previous model. More specifically, each cluster $G_i \in \mathcal{G}$ is compared with the existing clusters $C_j \in \mathcal{C}$, using a distance metric. Based on [87], in this work is proposed the following comparison between two clusters using the Mahalanobis distance:

$$\bar{d}_{ij} := d_{ij} + d_{ji}, \tag{3.36}$$

where $\bar{d}_{ij}$ considers the two distance values that are obtained between clusters i and j, by using Equation (3.37).

[1] Recall that, according to [184], λ_j should be chosen very close to 1 and δ should be a large value.

[2] We provide the details of this procedure in the appendix.

If there is a specific distance $d_{j,k}$ measured between C_j and G_k that results lower than a predefined threshold $\varepsilon_{d_{jk}}$, the clusters C_j and G_k are merged using Eq. (3.38)-(3.43). The clusters C_j and G_k are removed from $\mathcal{C}$ and $\mathcal{G}$, while $\mathcal{C}'$ is joined to $\mathcal{C}$. Then, after the checking process of close clusters is done, the new set of clusters $\mathcal{G}$ joins the previous set $\mathcal{C}$.

After the process of updating or the cluster creation mechanism is completed, the algorithm must check if the actual clusters in $\mathcal{C}$ are not redundant or useless to model the process P. Therefore, the algorithm proceeds with the following cluster merging, and cluster removal procedures.

3.2.4 Cluster Merging

Two clusters are merged if their distance is below a given threshold $\varepsilon \in \mathbb{R}^+$. More specifically, the distance between cluster i and j with respect to cluster i is defined as:

$$d_{ij} = \sqrt{(\boldsymbol{\kappa}(k)^\top \Sigma_i^{-1} \boldsymbol{\kappa}(k)}, \tag{3.37}$$

where $\boldsymbol{\kappa}(k) := \boldsymbol{\mu}_i(k) - \boldsymbol{\mu}_j(k))$.

The new cluster C' is defined by the following centers, covariance matrix, parameters and spreads:

$$n'(k) := n_i(k) + n_j(k), \tag{3.38}$$

$$\boldsymbol{\mu}'(k) := \frac{n_i(k)\boldsymbol{\mu}_i(k) + n_j(k)\boldsymbol{\mu}_j(k)}{n'(k)}, \tag{3.39}$$

$$\boldsymbol{\Sigma}'(k) = \frac{1}{n'(k)}\Big(n_i(k)\boldsymbol{\Sigma}_i(k) + n_j(k)\boldsymbol{\Sigma}_j(k) + \frac{n_i(k)n_j(k)}{n'(k)}\boldsymbol{\kappa}(k)^\top\boldsymbol{\kappa}(k)\Big) \tag{3.40}$$

$$\boldsymbol{g}'(k) := \frac{n_i(k)\boldsymbol{g}_i(k) + n_j(k)\boldsymbol{g}_j(k)}{n'(k)}, \tag{3.41}$$

$$\underline{\boldsymbol{s}}'(k) := \frac{n_i(k)\underline{\boldsymbol{s}}_i(k) + n_j(k)\underline{\boldsymbol{s}}_j(k)}{n'(k)}, \tag{3.42}$$

$$\overline{\boldsymbol{s}}'(k) := \frac{n_i(k)\overline{\boldsymbol{s}}_i(k) + n_j(k)\overline{\boldsymbol{s}}_j(k)}{n'(k)}. \tag{3.43}$$

We remark that the update formulas are based on the unsupervised method described in [87] and in [185] for the covariance update, except for the spread formulas in Equations (3.42) and (3.43) which constitute a novel contribution of this work. The cluster set $\mathcal{C}$ is modified removing the old clusters C_i and C_j and adding the new one.

At this point, an additional mechanism should be included to avoid an increase in the complexity of the $FPI(k)$ structure. Next, the removal mechanism of clusters that become obsolete is described.

3.2.5 Removal of obsolete clusters

During the online operation of the prediction interval, some clusters will become obsolete at a certain point, thanks to the evolution of the rest of the existing clusters. Due to that,

an additional evolving procedure needs to be considered in the proposed algorithm, which verifies the cluster activation across time. When a cluster has null activation degrees for an extended period, that cluster must be purged from the model because that situation indicates its local prediction interval model has become obsolete.

For that purpose, an index γ^j is used, which counts the number of consecutive instants where the activation degree of cluster j had null values, i.e., those consecutive instants with $\beta^j(\boldsymbol{Z}(k)) = 0$. Then, a cluster is removed when γ^j surpasses a threshold value. In this work, this condition is implemented as

$$\gamma^j \geq k_{max}, \tag{3.44}$$

where k_{max} represents the maximum period allowed for a cluster to remain with null values for its activation degree.

Finally, the online iteration of the algorithm finishes with the testing stage of the updated evolving fuzzy prediction interval. To test the resulting interval, the behavior of some metrics, such as the coverage level, must be checked. This point of the algorithm proposes to compute the coverage level obtained by the new interval when modeling the second half of data $D^{(2)}$. The measurement of this metric is denoted as $\hat{\alpha}_{new}$, and is obtained using (3.10). To determine if the new interval is accurate or not, the value $\hat{\alpha}_{new}$ is compared with the coverage level estimator $\hat{\alpha}$, which was previously estimated for the original training dataset using Equation (3.24). In this work, it is proposed to rely on a standard two-tailed z-test for defining if $\hat{\alpha}_{new}$ is significantly different from $\hat{\alpha}$, using the normal approximation for the binomial distribution. Based on this, the following z-score function is considered

$$z = \frac{\hat{\alpha}_{new} - \hat{\alpha}}{\sqrt{\hat{\alpha}_{new}(1 - \hat{\alpha}_{new})/n}}, \tag{3.45}$$

where n is the number of samples used to compute $\hat{\alpha}_{new}$, i.e., $n = N/2$ based on the size of $D^{(2)}$.

These statistics should be tested with the quantile of order $\beta/2$ and $1-\beta/2$ of the Normal distribution. Suppose the test provides evidence that $\hat{\alpha}_{new}$ is different from $\hat{\alpha}$. In that case, the algorithm decides to discard the interval and, from now on, it uses only the clusters $\mathcal{G}$ identified using $D^{(1)}$, which are known to have reached a proper value of coverage level when tested on $D^{(2)}$. Otherwise, the algorithm proceeds to its next step without introducing new changes to the interval model.

At this point, all of the details of the first proposed algorithm are covered. The high-level description of the proposal is provided in Algorithm 1, incorporating all of those mechanisms able to adjust its structure to the changes presented by the system process P over time.

Algorithm 1 High-Level Algorithm

Input: Dataset $\mathcal{D}$
Offline Phase
Generate the cluster set $\mathcal{C}$ using $FPI(\mathcal{D})$
Compute $\hat{\alpha}$ according to Eq. (3.24)
Online Phase
for $t \in \{T+1, T+2, \ldots\}$ **do**
    Create a new dataset $\mathcal{D}_t$ according to Eq. (3.25)
    Compute the coverage level $\hat{\alpha}_t$ for $\mathcal{D}_t$
    Compute the coverage level $\alpha_{M,t}$ according to Eq. (3.26)
    if $\alpha_{t,M} - \hat{\alpha}_t \geq Th$ **then**
        $n_\alpha \leftarrow n_\alpha + 1$
    else
        $n_\alpha \leftarrow 0$
    end if
    Compute $d_j(\mathbf{x}(k))$ according to Eq. (3.27)
    if $\min_j d_j(\mathbf{x}(k)) \leq d_{\max}$ **then**
        Update mean $\boldsymbol{\mu}_j(k-1)$ and variance $\boldsymbol{\sigma}_j^2(k-1)$
        Update cluster parameters $\boldsymbol{g}_j(k-1)$
        Update spreads $\overline{\boldsymbol{s}}^j(k-1)$ and $\underline{\boldsymbol{s}}^j(k-1)$
    else
        The data point $\mathbf{x}(k)$ is considered as an outlier
    end if
    for $k, j \in \{1, \ldots, q\}, k \neq j$ **do**
        if $d_{k,j} \leq \varepsilon$ **then**
            Merge C_i and C_j with Eq. (3.38)-(3.43) in C'
            $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_i, C_j\}$
            $\mathcal{C} \leftarrow \mathcal{C} \cup C'$
        end if
    end for
    if $n_\alpha \geq N$ **then**
        Divide D_t into $D^{(1)}$ and $D^{(2)}$
        Run FPI on $D^{(1)}$ to get a cluster set $\mathcal{G}$
        for $C_j \in \mathcal{C}, G_i \in \mathcal{G}$ **do**
            if $d_{j,k} \leq \varepsilon$ **then**
                Merge C_j and G_i with Eq. (3.38)-(3.43) in C'
                $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_j\}$
                $\mathcal{G} \leftarrow \mathcal{G} \setminus \{G_i\}$
                $\mathcal{C} \leftarrow \mathcal{C} \cup C'$
            end if
        end for
        $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{G}$
        Compute $\hat{\alpha}_{new}$ of the FPI system on $D^{(2)}$
        if $\hat{\alpha}_{new} \leq \hat{\alpha}_t$ **then**
            $\mathcal{C} \leftarrow \mathcal{G}$
        end if
    end if

45: **for** $C_j \in \mathcal{C}$ **do**
46: **if** $\sum_{h=k-T}^{t} \mathbb{1}\{\beta_j(\mathbf{x}(h)) < tol\} = T$ **then**
47: $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_j\}$
48: **end if**
49: **end for**
50: **end for**

This part of the proposal was presented in the journal paper:

- **O. Cartagena**, F. Trovò, M. Roveri and D. Sáez, "Evolving Fuzzy Prediction Intervals in Nonstationary Environments," in IEEE Transactions on Emerging Topics in Computational Intelligence, (Early Access), doi: 10.1109/TETCI.2023.3296486. **[Journal Impact Factor 2022: 5.3 - Q2 Computer Science - published]**.

Considering that the basis of this method is mainly inspired by the previous offline fuzzy prediction interval identification, it is expected to achieve good results when modeling nonstationary systems. Furthermore, this proposal applies a complete checking process of the interval performance to verify the actual interval's validity. However, including all of the described mechanisms makes the algorithm more complex and produces a high computational cost in its implementation. Due to this reason, in the next section a second proposal that seeks a low-cost implementation of evolving fuzzy prediction intervals is described.

3.3 Proposal 2: Self-evolving fuzzy prediction interval modeling

In order to achieve a simpler implementation of evolving intervals that achieve competitive results with respect to the first proposal, a new evolving fuzzy prediction interval design is presented in this section.

This novel proposal starts with a structure similar to that included in the design of the previous method presented in Section 3.2. Significant algorithm changes are associated with the removal of the offline identification and verification (and possible retraining of the model) stages, which produced the high computational cost of the previous proposal. Additionally, a cluster splitting mechanism is included this time in the method proposal to try to avoid large increments in prediction error. Similar as done in Figure 3.1 with the previous proposed evolving fuzzy prediction interval, a summary of the main stages of this new self-evolving proposal is presented in Figure 3.3.

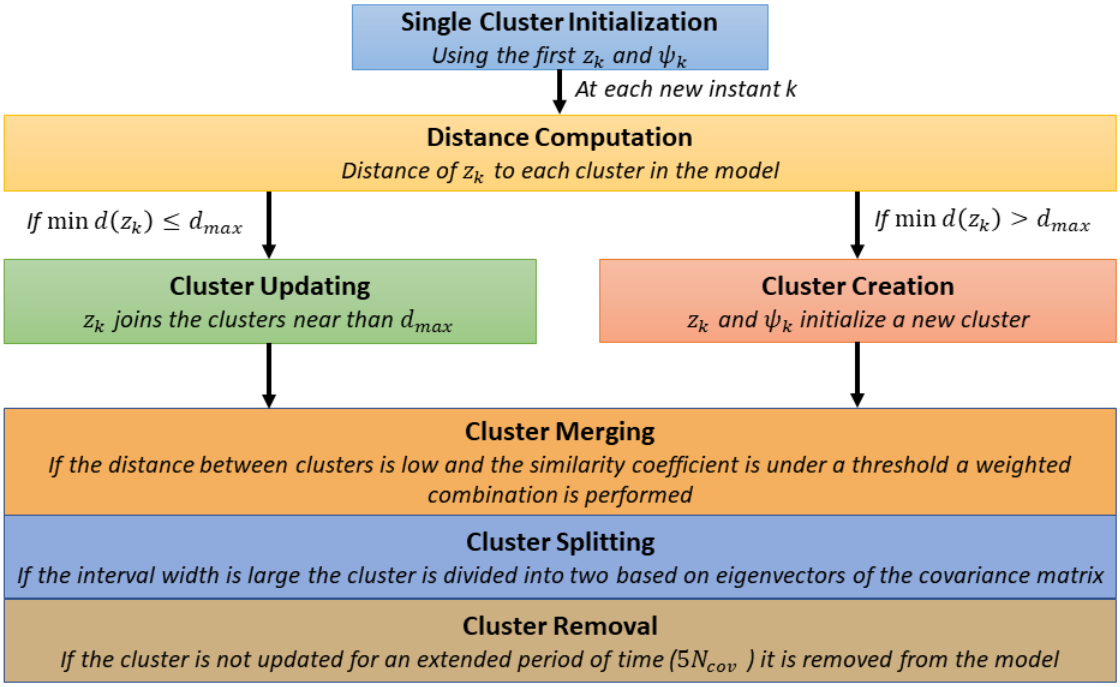


Figure 3.3: Summary of the proposed self-evolving fuzzy interval algorithm.

The proposed evolving fuzzy prediction interval starts with the first stage shown in Figure 3.3 which corresponds to a single cluster initialization. Here, the proposed algorithm will obtain the elements associated with the antecedents of the fuzzy rules, such as the center v and the covariance matrix $\boldsymbol{\Sigma}$, from the sample vector $\boldsymbol{z}(k)$ which is defined as:

$$\boldsymbol{z}(k) = [y(k-1), u(k-1), \omega(k-1)]^T, \tag{3.46}$$

where $y(k-1), u(k-1), \omega(k-1)$ are the previous values of the output, the input signal, and the measured disturbances that affect the system, respectively.

On the other hand, the parameter vector $\boldsymbol{\theta}$ associated with the consequences of the rules will be given by the structure of the regressor vector $\boldsymbol{\psi}(k)$. In this proposal, the regressor vector is constructed as follows:

$$\begin{aligned}\boldsymbol{\psi}(k) =&[1, y(k-1), \ldots, y(k-m), u(k-1), \ldots, u(k-m),\\ &\omega(k-1), \ldots, \omega(k-m)]^T,\end{aligned} \tag{3.47}$$

where m is the order considered for the local ARX models associated with each cluster.

The initialization of the model is done by creating a single cluster, following the idea from evolving fuzzy models. Considering that there is no prior information available for the parameter vector, it is proposed to initialize this element with the value $\boldsymbol{\theta}_c = \vec{1}$. Finally, to adjust the magnitude of the evolution of each spread parameter associated with the interval width, $\underline{\boldsymbol{s}}_c$ and $\overline{\boldsymbol{s}}_c$ start with the value:

$$\underline{\boldsymbol{s}}_c = \overline{\boldsymbol{s}}_c = \eta_s \vec{1} \cdot (|\psi(1)| + 1)^{-1}, \tag{3.48}$$

where η_s is a tuning parameter.

With the first cluster initialized, the user needs to provide the values of six different parameters to operate the learning phase of the algorithm. One of these is the number of data N_{cov} considered to estimate the interval coverage level. In this work, this value is defined based on the number of parameters to be identified, following the same procedure to determine the necessary data to perform a model identification with neural networks, i.e., based on previous works in machine learning, N_{cov} should be at least ten times the number of parameters to be identified [180]. The second parameter is the number of steps considered for making the predictions N_{pred}. This value should be selected based on the system's time response so that the prediction horizon can cover the system's transitory behavior. The third parameter is the target value for the coverage $\hat{\alpha}$, which should usually be selected within the range $[0.95, 0.999]$, based on the typical values of significance level for the coverage of the interval [186]. In addition, previous works in the literature on prediction intervals also used $\alpha = 0.9$ [6]. The other parameters to be selected are d_{Th} and κ_{join}, representing the distance threshold and the minimal overlap ratio considered for merging clusters. In this work, d_{Th} is obtained in a heuristic manner based on the values contained in the sample vectors $\boldsymbol{z}(k)$; however, it is recommended that this parameter not exceed the value $0.1 \max\{\boldsymbol{z}\}$. On the other hand, this work considers the fixed value $\kappa_{join} = 1.2$, based on the results reported on [187]. Finally, the last parameter to be defined is the maximum interval width tolerated for each cluster w_{max}. This parameter has a fixed value in this work ($w_{max} = 2$) and is decided similarly to the distance threshold. However, the value w_{max} could also be determined based on the behavior of the standard deviation presented by the modeling error.

Based on the parameters provided by the user, the learning phase of the evolving fuzzy prediction interval consists of the same mechanisms mentioned in Section 3.2 to perform the model update. Therefore, as soon as a new measurement $\boldsymbol{z}(k)$ is available, the algorithm starts with the second stage shown in Figure 3.3, which corresponds to the distance computation to select the closest cluster. This selection is made based on the values of the Mahalanobis distance between the sample $\boldsymbol{z}(k)$ and each existing cluster in the model. This is done by applying equation (3.27). Then, if the minimum value of $d_j(\boldsymbol{z}(k))$ is less than d_{Th}, the algorithm proceeds with the cluster update path (third stage shown in Figure 3.3), where the antecedents of the closest cluster are updated. Thus, for the cluster j associated with $\min d_j$, the center and covariance matrix are updated using (3.28)-(3.31), based on the Welford algorithm.

In the same stage of cluster update shown in Figure 3.3, after updating the antecedents of the closest clusters, the consequences of all nearby clusters should also be updated. To perform this update, only clusters with activation degrees $\beta_j(\boldsymbol{z}(k))$ greater than 0.1 are included in this process. This threshold is defined with the purpose of being less strict in selecting clusters when the number of clusters in the model c increases. Then, for the clusters that meet the threshold, the parameter vector $\boldsymbol{\theta}_j$ is updated using equations (3.32)-(3.33), which corresponds to the recursive least squares method.

At this point, the clusters selected for the process of updating its consequences must adapt the spread values s_i associated with the width of the interval. To do that, a count x_i is proposed to estimate the past coverage level of the intervals associated with each cluster.

Thus, x_i is defined for each cluster i as:

$$x_i(k) = \begin{cases} x_i(k-1) + \frac{1}{N_{cov}} & \text{if} \quad y_i(k) \in [\underline{y}_i(k), \overline{y}_i(k)] \\ x_i(k-1) - \frac{1}{N_{cov}} & \text{otherwise,} \end{cases} \tag{3.49}$$

and is saturated within the range $[0, 1]$. Then, for each cluster i that complies with the condition $\beta_i > (1/(c+1))$, its spread parameters are updated as

$$\overline{\boldsymbol{s}}_i(k+1) = \overline{\boldsymbol{s}}_i(k) - (x_i(k) - \hat{\alpha})|\psi_i(k)|\overline{\boldsymbol{s}}_i(k), \tag{3.50}$$

$$\underline{\boldsymbol{s}}_i(k+1) = \underline{\boldsymbol{s}}_i(k) - (x_i(k) - \hat{\alpha})|\psi_i(k)|\underline{\boldsymbol{s}}_i(k), \tag{3.51}$$

where $\hat{\alpha}$ iss the target value of the coverage level. Additionally, the user can consider applying equations (3.50)-(3.51) only if the cardinality of the cluster exceeds half of the value N_{cov} defined to estimate the coverage level. The user can include that condition to avoid updating the interval width during the initial instants of a cluster when the model still does not converge to an optimal value of the parameters $\boldsymbol{\theta}$. Here, the spread parameters associated with different prediction steps up to N_{pred} could be defined. Still, in that case, the new parameters associated with the additional prediction steps have to be handled by separate coverage estimations using the corresponding application of (3.49).

On the other hand, if $\min d_j > d_{Th}$ happens during the selection process of the closest cluster, the second path considered in Figure 3.3 for the algorithm's third stage is followed. In this alternative stage, a new cluster is created from scratch following the same procedure of initialization of the model (first stage of the algorithm in Figure 3.3). Here, the variable c that indicates the number of clusters available in the model must increase by one. Furthermore, in this part of the algorithm, the user can consider using the parameter vector $\boldsymbol{\theta}_{c-1}$ of an existing cluster as the initial parameter vector $\boldsymbol{\theta}_c$ of the new cluster c.

After all the previous procedures are performed to update existing clusters or create a new cluster, the algorithm progresses with the fourth stage of the diagram shown in Figure 3.3. Here, the algorithm needs to check if a merging process is required. To make this decision, as same as the previous proposal, the Mahalanobis distance between the center of clusters k and j computed using (3.37) can be considered as a first criterion for applying the merging process. Additionally, this method proposes to include here a second criterion presented previously on [187], where the clusters must comply with a similarity condition before performing the merging process. To evaluate this additional condition, for each candidate j the volume V_j of its hyperellipsoid in m dimensional hyperspace is computed as [187]:

$$V_j = \frac{2\pi^{m/2}}{m\Gamma(m/2)} \prod_{l=1}^{m} \lambda_l, \tag{3.52}$$

where Γ is the gamma function, m is the length of the sample vector $\boldsymbol{z}(k)$ and λ_l are the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}_j$.

The new combined cluster ij is obtained by a weighted combination of the elements from clusters i and j. Therefore, the new combined cluster is defined by the centers, covariance matrix, and parameters computed using (3.38)-(3.43).

Here, from the new covariance matrix $\boldsymbol{\Sigma}_{ij}$ its hyperellipsoid V_{ij} in m dimensional hyperspace is computed using equation (3.52). Then, the overlapping of clusters (i, j) is measured

by calculating the ratio between the joined volume V_{ij} and the sum of both cluster volumes as follows [187]:

$$\kappa_{ij} = \frac{V_{ij}}{V_i + V_j}. \tag{3.53}$$

If the ratio κ_{ij} is less than the predefined joint threshold value κ_{join} the new combined cluster ij replaces the previous clusters i and j in the model.

Conversely to merging clusters, the fifth stage shown in Figure 3.3 consists on the splitting procedure of clusters that have become too large. This procedure is usually applied only if the cardinality n_j of the cluster j and its local predictive error become too large. However, in this part of the algorithm, the user can consider that the local width acts similarly to an integrator of the past error based on the adaptation of the prediction interval. Thus, the proposed algorithm considers two conditions for splitting a cluster, a threshold for the minimum cardinality n_j and a threshold for the maximum interval width w_{max} admitted for each cluster. If for a sample vector $\boldsymbol{z}(k)$ both thresholds are surpassed, the cluster j is divided into two new clusters i and $c+1$. Based on an equal division of the cluster j, the cardinalities must comply with the condition $n_i = n_{c+1} = n_j/2$ and the parameters $\boldsymbol{\theta}, \underline{\boldsymbol{s}}, \overline{\boldsymbol{s}}$ will be the same vectors as the original cluster. To obtain the new centers $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_{c+1}$ and the new covariance matrices $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_{c+1}$, the following formulas are used:

$$\boldsymbol{\mu}_i = \boldsymbol{\mu}_j - \eta v_{max}, \quad \boldsymbol{\mu}_{c+1} = \boldsymbol{\mu}_j + \eta v_{max}, \tag{3.54}$$

$$\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{c+1} = \boldsymbol{\Sigma}_j - \eta^2 v_{max} v_{max}^T, \tag{3.55}$$

where v_{max} is the eigenvector associated with the highest eigenvalue of the covariance matrix $\boldsymbol{\Sigma}_j$, and $\eta \in (0, 1]$ is a parameter to determine the separation of the new clusters i and $c+1$. In this work we use the value $\eta = 0.5$ for performing the splitting. In this process, equations (3.54)-(3.55) are originally derived from the merging process defined by (3.38)-(3.43), thus, if the two new clusters generated in this step are instantly merged, we can recover the original cluster j.

Finally, the algorithm continues with the sixth and last stage shown in Figure 3.3, which consists of the removal of obsolete clusters. In the proposed algorithm, a cluster j can be considered obsolete if the variable o_j becomes large. Here, o_j is defined as a count of the number of iterations that have passed since the last update of the parameters. Thus, formally the variable o_j has the following two possible values:

$$o_j(k) = \begin{cases} 0 & \text{if } \boldsymbol{\theta}_j \text{ is updated at instant } k, \\ o_j(k) + 1 & \text{otherwise.} \end{cases} \tag{3.56}$$

With this explanation of the removal process completed, all the evolving mechanisms considered in this proposal are already presented. The high-level description of this second proposal is provided in Algorithm 2.

Algorithm 2 Learning phase - Evolving FPI

1: **Inputs:** Measurements $\boldsymbol{z}(k)$ and regressor vector $\boldsymbol{\psi}(k)$
2: **User-defined parameters:** N_{pred}, N_{cov}, $\hat{\alpha}$, d_{Th}, κ_{join}, w_{max}
Initialization:
3: $c \leftarrow 1$, $\boldsymbol{\mu}_c \leftarrow z(1)$, $k \leftarrow 1$, $\boldsymbol{\theta}_c \leftarrow \vec{1}$, $n_c \leftarrow 1$, $x_c \leftarrow 0$, $\underline{\boldsymbol{s}}_c, \overline{\boldsymbol{s}}_c \leftarrow \vec{1} \cdot (|\psi(1)| + 1)^{-1}$
Learning phase:
4: **repeat** $k \leftarrow k + 1$
5: Measure $\boldsymbol{z}(k)$ and compute $\boldsymbol{\psi}(k)$
6: **for** $i = 1, \ldots, c$ **do**
7: Measure the distances d_i to each v_i in the model
8: **end for**
9: **if** $\min d_i < d_{Th}$ **then**
10: $i_c \leftarrow$ index j of cluster with $d_j = \min d_i$
11: **else**
12: $i_c \leftarrow 0$
13: **end if**
14: **if** $i_c > 0$ **then**
15: Update $\boldsymbol{\mu}_{i_c}$, $\boldsymbol{\Sigma}_{i_c}$ and n_{i_c} using Welford's algorithm
16: **for** $i = 1, \ldots, c$ **do**
17: Compute the activation degrees β_i
18: **if** $\beta_i > 1/(c+1)$ **then**
19: Update $\boldsymbol{\theta}_i$ using RLS algorithm
20: Compute $\hat{y}_i(k), \underline{y}_i(k), \overline{y}_i(k)$
21: **if** $\underline{y}_i(k) \leq \beta_i \cdot y(k) \leq \overline{y}_i(k)$ **then**
22: $x_i \leftarrow x_i + 1/N_{cov}$
23: **else**
24: $x_i \leftarrow x_i - 1/N_{cov}$
25: **end if**
26: x_i is saturated within the range $[0, 1]$
27: **if** $n_i > N_{cov}/2$ **then**
28: $\underline{\boldsymbol{s}}_i, \overline{\boldsymbol{s}}_i \leftarrow s_i(k) - (x_i - \hat{\alpha})|\boldsymbol{\psi}(k)|s_i(k)$
29: **end if**
30: $o_i \leftarrow 0$
31: **else**
32: $o_i \leftarrow o_i + 1$
33: **end if**
34: **end for**
35: **else**
36: $c \leftarrow c + 1$
37: Initialize cluster c as done in Line 3
38: $\boldsymbol{\theta}_c \leftarrow \boldsymbol{\theta}_{c-1}$
39: **end if**

Compute the distance d_{ij} between clusters i and j
$\bar{d}_{ij} \leftarrow \min(d_{ij}, d_{ji})$
if $\bar{d}_{ij} < d_{Th}$ **then**
  Merge clusters i and j by a weighed combination
  Obtain the volume of the cluster combination V_{ij}
  Compute the overlapping ratio κ_{ij} using V_i, V_j and V_{ij}
  if $\kappa_{ij} < \kappa_{join}$ **then**
    New cluster ij replaces clusters i and j
  end if
end if
for $i = 1, \ldots, c$ **do**
  if $n_i > 5N_{cov}$ and $|\overline{y}_i(k) - \underline{y}_i(k)| > w_{max}$ **then**
    $c \leftarrow c + 1$
    Split the cluster i in two clusters of size $n_i/2$
    Use the eigenvectors of $\mathbf{\Sigma}_i$ to obtain v_i, $\boldsymbol{\mu}_c$
  end if
  if $o_i/n_i > N_{cov}$ **then**
    Remove cluster i
  end if
end for
until End of learning phase

This part of the proposal was presented in the journal paper:

- **O. Cartagena**, M. Ožbot, D. Sáez, I. Škrjanc, "Evolving fuzzy prediction interval for fault detection in a heat exchanger," in Applied Soft Computing, Volume 145, 110625, 2023, doi: 10.1016/j.asoc.2023.110625. **[Journal Impact Factor 2022: 8.7 - Q1 Computer Science - published]**.

The following section proceeds with an explanation of the background for the last part of the proposals reported in this thesis, which consist of the design of a fault detection method based on the information provided by prediction intervals.

3.4 Background: Model-based fault detection system

The deterministic model-based fault detection algorithm corresponds to those methods which use typical observer filters based on the system's dynamic. For example, if a nonlinear model is used, a simple observer usually can be expressed as:

$$\hat{x}(k+1) = f(\hat{x}(k), u(k)) + Kr(k), \tag{3.57}$$

$$\hat{y}(k) = C\hat{x}(k), \tag{3.58}$$

where $\hat{x}$ is the estimated state, f is the nonlinear model previously identified, and K is the observer gain. The observer residual signal $r(k)$ is given by the difference between the measurement $y(k)$ and the predicted output $\hat{y}(k)$, such that:

$$r(k) = y(k) - \hat{y}(k). \tag{3.59}$$

Here, the residual signal $r(k)$ is usually subjected to a residual evaluation function computed over a time window of N samples. A typical evaluation function is denoted as:

$$J(r(k)) = \sqrt{\sum_{k=0}^{N} r^T(k)r(k)}, \tag{3.60}$$

which is used in the denoted Root Mean Square Evaluation method (RMS). Then, an alarm can be activated based on a logical rule, where a fault is detected if $J(r(k))$ exceeds a threshold value J_{Th} [188]. Usually, the value J_{Th} is determined using past data of the system without faults.

Taking this idea of fault detection as a basis, we next present the proposal for the new algorithm that uses the prediction interval generated by the evolving algorithms.

3.5 Proposal 3: Interval-based fault detection algorithm

The last proposal to be presented in this thesis is the design of a new fault detection algorithm based on the information of the system's uncertainty provided by the prediction intervals. This proposal is based on the fact that a fault in the system will produce an increment of the modeling error, which will not coincide with the original uncertainty of the system's measurements previously characterized.

The proposed algorithm has the following structure: first, it starts with the identification of the prediction interval to characterize the behavior of a process. Then, the coverage level of the interval is constantly checked for a certain time window, and finally a fault is detected when this performance metric falls below a threshold value. A summary of this procedure is presented in Figure 3.4.

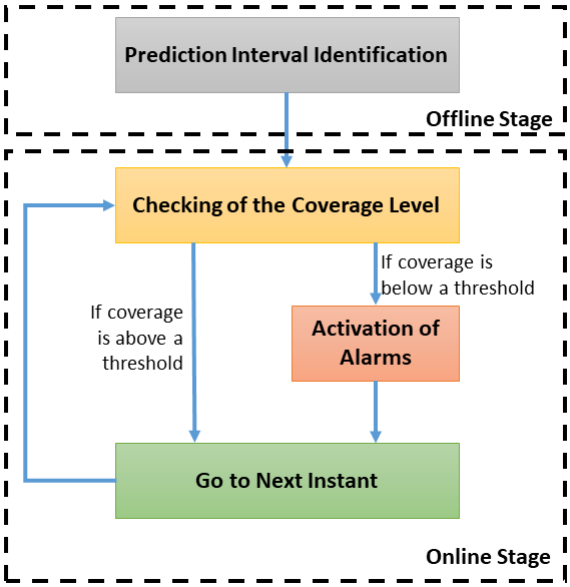


Figure 3.4: Summary of the proposed interval-based fault detection algorithm. The proposed method starts with an offline stage where the prediction interval is identified for representing the system dynamics. After that, the algorithm continues with the online stage, where the coverage level of the interval is monitored and the alarms are activated when the interval decrease its performance.

To run this new fault detection algorithm, the user has to define the same parameters N_{pred}, N_{cov}, $\hat{\alpha}$ and the distance threshold d_{Th} used during the learning stage of the second evolving prediction interval proposal. Then, by using the measurements on the fixed final prediction interval model obtained at the end of the learning stage (or the offline identification stage of the prediction interval, as shown in Figure 3.4), the algorithm can detect the system changes by identifying the moments when the model drops in performance.

The proposal proceeds with the first online stage shown in Figure 3.4, which consists of the checking process of the coverage level. At this stage, the algorithm can quantify this performance drop in terms of the coverage level, by using a failure index associated with the past instants when the interval failed to contain the measurements. This index can be estimated following a similar approach to that previously done by the evolving algorithm when having to estimate the coverage level. To do this, the accumulated failure count c_f is defined as follows:

$$c_f(k) = \begin{cases} c_f(k-1) - \frac{1}{N_{cov}} & \text{if} \quad y(k) \in [\underline{y}(k), \overline{y}(k)] \\ c_f(k-1) + \frac{1}{N_{cov}} & \text{otherwise,} \end{cases} \tag{3.61}$$

where the c_f value is capped to the range $[0, 1]$. Here, the algorithm can decide that a fault is occurring if $c_f(k)$ increases significantly more than $(1 - \hat{\alpha})$. If this is the case, the algorithm proceeds with the second online stage shown in Figure 3.4, which consists of the activation of the alarms. Otherwise, the algorithm's actual iteration finishes and goes directly to the next instant (as shown in the last part of the diagram presented in Figure 3.4).

The problem with this fault detection method is that the algorithm must check the failure

count for prediction steps greater than one. This requirement is because the prediction interval becomes more sensitive to the changes in a model for further prediction steps, adding some delays to the fault detection. Additionally, there would be more delays in fault detection if this proposed method is used because the algorithm relies on an estimation of failure for past N_{cov} instants. However, when a fault is detected at instant k using the predictions N_{pred} steps ahead, i.e., when comparing $y(k)$ with the output of the interval from N_{pred} instants before ($\hat{y}(k+N_{pred})$, $\underline{y}(k+N_{pred})$ and $\overline{y}(k+N_{pred})$), the algorithm can estimate how much the detection delay was. So, at the instant k when the algorithm activates the alarm, the beginning of the fault can be estimated as $k - N_{pred} - c_f(k) \times N_{cov}$.

With this explanation, the new fault detection method based on the information provided by prediction intervals is covered. The high-level description of this newly proposed method is provided in Algorithm 3.

Algorithm 3 Fault Detection

Inputs: Measurements $z(k)$ and regressor vector $\psi(k)$
User-defined parameters: N_{pred} N_{cov}, $\hat{\alpha}$, d_{Th}
for $i = 1, \ldots, c$ **do**
    if $n_i < N_{cov}$ **then**
        Remove cluster i from the model
    end if
end for
$F_c \leftarrow 0$
repeat $k \leftarrow k + 1$
    Measure $z(k)$ and compute $\psi(k)$
    Compute $\hat{y}(k), \underline{y}(k), \overline{y}(k)$ using the FPI model
    if $\underline{y}(k) \leq y(k) \leq \overline{y}(k)$ **then**
        $c_f \leftarrow c_f - 1/N_{cov}$
    else
        $c_f \leftarrow c_f + 1/N_{cov}$
    end if
    Saturate the count c_f within the range $[0, 1]$
    if $c_f - (1 - \hat{\alpha}) > Th$ **then**
        Identify $z(k)$ as a fault
        Estimate the beginning of the fault at instant $k - N_{pred} - c_f \times N_{cov}$
        $F_c \leftarrow 1$
    else
        $F_c \leftarrow 0$
    end if
until End of operation

This part of the proposal was presented in the journal paper:

- **O. Cartagena**, M. Ožbot, D. Sáez, I. Škrjanc, "Evolving fuzzy prediction interval for fault detection in a heat exchanger," in Applied Soft Computing, Volume 145, 110625, 2023, doi: 10.1016/j.asoc.2023.110625. **[Q1-IF 10.16 - published]**.

The following section presents the discussion of the proposals presented in this chapter.

3.6 Discussion

In this chapter, two evolving prediction interval techniques were proposed on the basis of the structure of a general fuzzy prediction interval method. Considering that most of the elements of the interval can be updated with methods that have been used successfully in previous works related with evolving intelligence systems, the uncertainty of the efficiency of this algorithm lies in the novel updating procedure of the interval width proposed in this thesis.

The first proposal mainly relies on methods inspired by the offline fuzzy interval identification. Thus, its design and operation have a high computational cost. For that reason, the second alternative proposed in this chapter relies on approximations of the coverage level for handling the updating of the model in a completely recursive manner. By doing that, the implementation of the algorithm results in less computational effort, facilitating the applicability of the model without sacrificing the interval's performance.

Additionally, thanks to the universal approximator characteristic associated with fuzzy models and the adaptability given by evolving models, the proposed evolving prediction interval could theoretically be used without any problem in a wide range of applied cases. To confirm this statement, the validation process of this proposal is very important, so this algorithm must be tested over several types of nonlinear systems with controlled changes entered into its dynamics during operation.

Finally, this chapter presented a novel application of this interval modeling methodology for the fault detection framework. Based on the information provided by the interval width regarding the modeling error, it is expected to achieve a fault detection algorithm that can compete with other model-based methods.

The next chapter shows the evaluation of the aforementioned evolving fuzzy prediction interval methods for some applied cases. Furthermore, the subsequent validation of the proposals based on one of the previous methods will be approached through numerical simulations using synthetic and real data.

4 Cases of study

This chapter presents the experiments carried out for evaluating the performance of fuzzy prediction intervals and the proposals presented in this work, including their corresponding results. First, this chapter presents the fuzzy prediction interval methods implemented for modeling the modified Chen series (simulated data) and solar power generation (real data). Then, the proposed evolving fuzzy interval methods are tested over a benchmark case and altered solar power generation data. Finally, the evolving intervals and the interval-based fault detection algorithm are tested over real measurements extracted from an experimental heat exchanger plant which operates with some controlled changes induced in its dynamics.

4.1 Evaluation of fuzzy prediction intervals

In this work, two benchmark cases are used for evaluating the methods described in Section 3.1: the modeling of the modified Chen series [189] and the forecast of solar power generation data from the Politecnico di Milan [190]. To make a proper comparison, all methods must be trained with a target value α =90% for the coverage level, across the selected prediction horizon.

The evaluation of the selected fuzzy prediction interval methods is first performed. This is done to justify the subsequent selection of the method based on fuzzy numbers as a basis of the proposed evolving fuzzy prediction interval design presented in Section 3.2.

In the following, the different cases of study considered for this work are explained in detail.

4.1.1 Modeling a nonlinear system

The evaluation of the fuzzy prediction interval methods described in Section 3.1, has been carried out on a first case of study corresponding to the modeling of a nonlinear dynamical system. The system considered is the modified Chen series, which is described by the following dynamical equation [55]:

$$\begin{aligned} y(k) = \; & [0.8 - 0.5\ exp\{-y^2(k-1)\}]\, y(k-1) - \\ & [0.3 - 0.9\ exp\{-y^2(k-1)\}]\, y(k-2) + \\ & u(k-1) + 0.2\ u(k-2) \\ & 0.1\ u(k-1)u(k-2) + e(k), \end{aligned} \tag{4.1}$$

where the noise depends on the previous state as follows

$$e(k) = 0.5\ exp\left\{-y(k-1)^2\right\}\beta(k), \tag{4.2}$$

and β is a white noise signal.

10,000 samples were obtained from the simulation of this series, separated into training, validation, and testing datasets in proportions of 60%, 20%, and 20%, respectively. Based on this, fuzzy models and their respective intervals were trained using the training set. The structure of the models, which is defined by the number of rules, the clustering method, and the selection of their input variables, was decided on the basis of the minimization of the prediction error in the validation dataset. Based on this, fuzzy models that considered the use of the Fuzzy C-Means algorithm for clustering resulted in a structure of 3 input variables (2 autoregressors for the output of the system $y(k)$ and 1 for the exogenous input $u(k)$) and 18 rules. Finally, the models and the corresponding prediction intervals were tested using the testing dataset, verifying that the trained intervals obtained a performance close to that obtained in the training set.

Table 4.1 shows the root mean square error (RMSE), PINAW and PICP (previously defined in equations 3.8 and 3.10, respectively) obtained with the selected methods (fuzzy numbers, joint supervision and type-2 TSK model) for one-, 8- and 16-prediction steps, all of them measured for the testing dataset. Additionally, the table includes the number of parameters given by the crisp model and the interval model.

Table 4.1: Performance metrics for the modified Chen series in the testing dataset.

Prediction Horizon	**Metrics**	**Fuzzy Models**		
		Fuzzy Numbers	**Joint Supervision**	**Type-2 TSK**
1 step ahead	RMSE	0.243	0.332	0.252
	PICP (%)	89.14	88.79	88.14
	PINAW (%)	4.27	7.18	5.70
8 steps ahead	RMSE	0.534	0.61	0.495
	PICP (%)	89.35	88.7	90.36
	PINAW (%)	11.36	15.42	11.59
16 steps ahead	RMSE	0.618	0.65	0.521
	PICP (%)	89.86	91.33	91.17
	PINAW (%)	12.68	16.25	12.47
Number of Parameters	Crisp model	180	-	-
	Interval model	108	69	306

As reported in Table 4.1, all methods achieve similar coverage level results, close to the target value defined for training (PICP = 90%). This indicates that these prediction intervals have been properly identified and maintain a coverage performance similar to that obtained during training when a different dataset is used. An important difference among the methods is the interval width (PINAW) obtained along the prediction horizon. The fuzzy numbers method and the type-2 fuzzy models show the best performances with the lowest values of

PINAW for more prediction steps due to the greater complexity of the interval structure associated with a large number of parameters.

As an example, Figure 4.1 shows the prediction interval obtained with the method based on fuzzy numbers for a 16-steps ahead prediction.

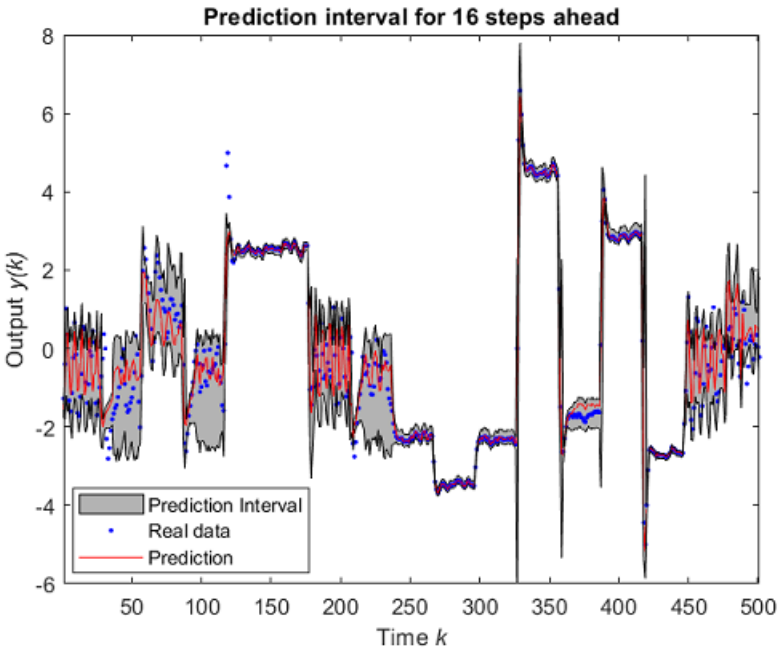


Figure 4.1: 16-steps ahead prediction interval based on fuzzy numbers. The red line representing the output of the fuzzy model for 16 prediction steps and the interval defined by the gray area are compared with the future measurements extracted later from the Chen time-series.

From these results, the method based on fuzzy numbers and the type-2 fuzzy models arise as the main candidates to be used in the later evolving interval design. Here, the fuzzy numbers method presents a small advantage thanks to its greater versatility due to its lower structure complexity compared to type-2 fuzzy models. To confirm this preliminary conclusion, a second case study is proposed to complement the evaluation of the fuzzy prediction interval methods and thus be able to choose the most appropriate method for the evolving interval design. The simulation results for the second benchmark case are presented in what follows.

4.1.2 Modeling solar power generation data

Solar power generation data from the Multi-Good Microgrid Laboratory of the Politecnico di Milano, Milan, Italy [190], are used as the second case study to test the prediction interval methods studied in this work. The available data correspond to measurements taken in the years 2017 and 2018, with a sample time of one hour. 11,688 samples are divided in the same proportions as in the previous case study: 60% for training, 20% for validation and 20% for testing.

The interval training process is performed using a target value for the coverage level of 90%. Similarly to the Chen series experiment, the optimization of the structure of the models was decided on the basis of the configuration that had the best performance of the metrics in the testing dataset. A total of 48 regressors, corresponding to data from two previous days, were used as initial candidate inputs, selecting the most relevant inputs by performing a sensitivity analysis. Based on this, the fuzzy models that use the Fuzzy C-Means algorithm for clustering resulted in a structure of 23 input variables and 2 rules.

Table 4.2 shows the performance metrics obtained with the selected methods (fuzzy numbers, joint supervision, and type-2 TSK model) for different prediction horizons, all of them measured on the testing dataset.

Table 4.2: Performance metrics in the testing dataset for the solar power generation.

Prediction Horizon	**Metrics**	**Fuzzy Models**		
		Fuzzy Numbers	**Joint Supervision**	**Type-2 TSK**
1 hour ahead	RMSE	13.48	14.51	13.85
	PICP (%)	89.65	90.57	89.05
	PINAW (%)	8.62	18.47	10.55
12 hours ahead	RMSE	25.95	26.01	27.01
	PICP (%)	89.08	90.17	89.04
	PINAW (%)	17.83	28.45	20.36
24 hours ahead	RMSE	25.91	26.02	27.37
	PICP (%)	88.76	91.13	88.63
	PINAW (%)	17.44	26.03	20.31
Number of Parameters	Crisp model	140	-	-
	Interval model	92	236	234

Table 4.2 shows that the models perform similarly to those obtained in the previously analyzed dataset (Chen time series). In these results, all the methods managed to comply with the desired coverage level (PICP) of 90%, while the fuzzy prediction intervals based on fuzzy numbers and type-2 models presented narrow intervals throughout the prediction horizon (see PINAW in Table 4.2). Additionally, Figure 4.2 shows the prediction intervals obtained with the fuzzy numbers method for a 24-hours ahead prediction.

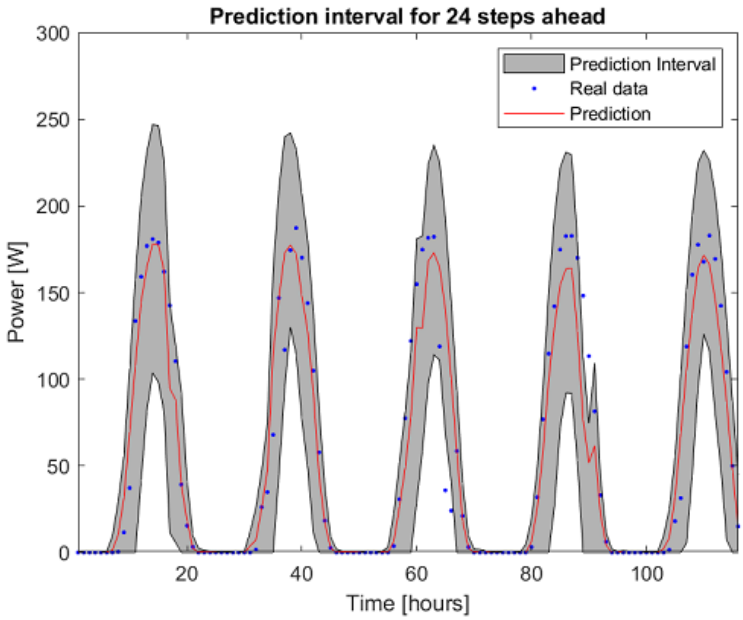


Figure 4.2: 24-steps ahead prediction interval based on fuzzy numbers for the solar power generation data. The red line representing the output of the fuzzy model for 24 prediction steps and the interval defined by the gray area are compared with the future measurements of solar power generation.

In Figure 4.2 it can be seen that the fuzzy prediction interval based on fuzzy numbers exhibits a smooth behavior throughout the plotted data.

Based on their performance, the methods based on fuzzy numbers and the type-2 fuzzy models remain the main candidates to be used in the later evolving interval design. The evaluation of the results obtained for the two cases of study and the definitive selection of the method to be used in the evolving interval design will be discussed below.

4.1.3 Comparison of methods

Based on the solar power generation modeling results, the previous statement mentioned during the Chen series modeling can be confirmed; the fuzzy numbers methods arise as the proper choice to develop the evolving interval design proposed in this work.

This decision is justified by the fact that the fuzzy number method has a simpler structure than the type-2 fuzzy model, which does not make more complex the development of the evolving interval. Additionally, the performance of the fuzzy number method was good for both benchmark cases studied. Here, the particular case of solar power generation modeling will be highlighted, where the fuzzy numbers method even presented a performance that exceeds the results achieved by the other interval methods shown in Tables 4.1 and 4.2, in terms of the lowest interval width. That trend was presented on the different prediction steps considered in the prediction horizon.

This part of the evaluation results reported in this section was presented in the journal

paper:

- **O. Cartagena**, S. Parra, D. Muñoz-Carpintero, L. G. Marín and D. Sáez, "Review on Fuzzy and Neural Prediction Interval Modelling for Nonlinear Dynamical Systems," in IEEE Access, vol. 9, pp. 23357-23384, 2021, doi: 10.1109/ACCESS.2021.3056003. **[Journal Impact Factor 2022: 3.9 - Q2 Computer Science - published]**.

With the definitive selection of the base fuzzy interval method for the development of the evolving algorithm, this chapter continues with testing the evolving fuzzy prediction interval designs over simulated data.

4.2 Simulation results of evolving fuzzy prediction intervals

In this section, a comparison between the performance of the proposals described in Sections 3.2 and 3.3 is shown. Two applied cases were considered to test the proposals: the modeling of synthetic data obtained from a generic nonlinear time-variant system and the forecasting of solar power generation data from the Politecnico di Milan. The prediction intervals were trained to comply with a 90% coverage level in the training dataset for both simulation cases. The analysis of the proposal is performed by comparing five different versions of the fuzzy prediction intervals.

- Classical fuzzy prediction interval (FPI) implemented with fuzzy numbers [2].
- Fuzzy prediction interval that only includes the proposed update mechanism for existing clusters (U-FPI) described in Section 3.2.
- Fuzzy prediction interval that only includes creating and merging new clusters (C-FPI) described in Section 3.2.
- First proposal: evolving fuzzy prediction interval (E-FPI), which considers all the previous evolving mechanisms described in Section 3.2.
- Second proposal: self-evolving fuzzy prediction interval (SE-FPI), which considers the evolving mechanisms described in Section 3.3.

This strategy is used for simulation tests to observe separately the effects obtained by each proposed mechanism when handling time-variant data. The simulation results for both cases of study considered for this work are shown below.

4.2.1 Modeling of synthetic data

For obtaining the synthetic data to be used for testing the methods, a nonlinear system given by

$$y(k) = sin\left(a_1 y(k-1) + a_2 y(k-2) + b_1 u(k-1)\right) + d(k), \tag{4.3}$$

is implemented. There, $d(k)$ is a Gaussian noise signal with mean zero and standard deviation $\sigma = 0.0001$. Additionally, the input signal is computed as

$$u(k) = b_u u(k-1) + e(k), \tag{4.4}$$

where $b_u = 0.4$ and $e(k)$ is a random signal with mean zero and standard deviation $\sigma = 1$.

Initially, 10,000 samples were simulated using the following fixed values for the system parameters: $a_1 = 0.1$, $a_2 = 0.2$, and $b_1 = -0.1$. These samples are used for obtaining the initial prediction intervals using the 50% of data as the training dataset, the next 25% as a validation dataset, and the final 25% as a testing dataset. The initial prediction interval is then identified using a particle swarm optimization algorithm.

The evaluation of the initial interval is carried out in terms of three different metrics. First, the root mean square error (RMSE) is measured for the base prediction model. Additionally, the PINAW and PICP, previously defined on equations (3.8) and (3.10), are used for evaluating the interval width and the coverage level. These interval metrics obtained in the training and testing dataset are reported in Table 4.3. These results show that the trained interval can perform accurately when tested with different data. The simulation results for both cases of study considered for this work are shown below.

For simulating the occurrence of an abrupt change in the system, 10,000 new samples were simulated using a second set of fixed parameters: $a_1 = 0.4$, $a_2 = 0.1$, and $b_1 = 0.2$. On the other hand, for simulating a gradual drift in the system, each of the parameters $p_i = \{a_1, a_2, b_1\}$ are defined for each new sample k as follows

$$p_i(k) = p_i^{(1)} + \left(\frac{p_i^{(2)} - p_i^{(1)}}{10000} \right) k, \tag{4.5}$$

where $p_i^{(1)}$ is the corresponding value that comes from the first set of parameters used for obtaining the initial intervals, while $p_i^{(2)}$ is the value extracted from the second set used for simulating the abrupt change.

In the case of the gradual drift simulation, an additional dataset of $10,000$ samples obtained with the second set of fixed parameters (with $a_1 = 0.4$, $a_2 = 0.1$) is considered after the previous measurements. This is done to include the study of the interval behavior when the system stops changing parameters.

The online evaluation of the interval is performed using the evolution of the following metrics: the RMSE obtained by the predictive model, the PICP, and the average of the interval width (PINAW). In this case, these metrics are measured at each instant k using a

Table 4.3: Interval metrics obtained for predictions 1-step ahead of the synthetic data.

Metrics	**Training**	**Testing**
RMSE	0.0995	0.0998
PICP [%]	90.358	89.996
PINAW [%]	36.897	43.090

time window of the N past measurements. Figure 4.3 shows the evolution of the coverage level and interval width when the system presents an abrupt drift in its dynamics, while Table 4.4 shows the average values for the metrics measured for 10 different simulations of this experiment with the standard deviation value enclosed between parentheses. On the other hand, Figure 4.4 and Table 4.5 show the same results for the case of a gradual drift in the behavior of the system.

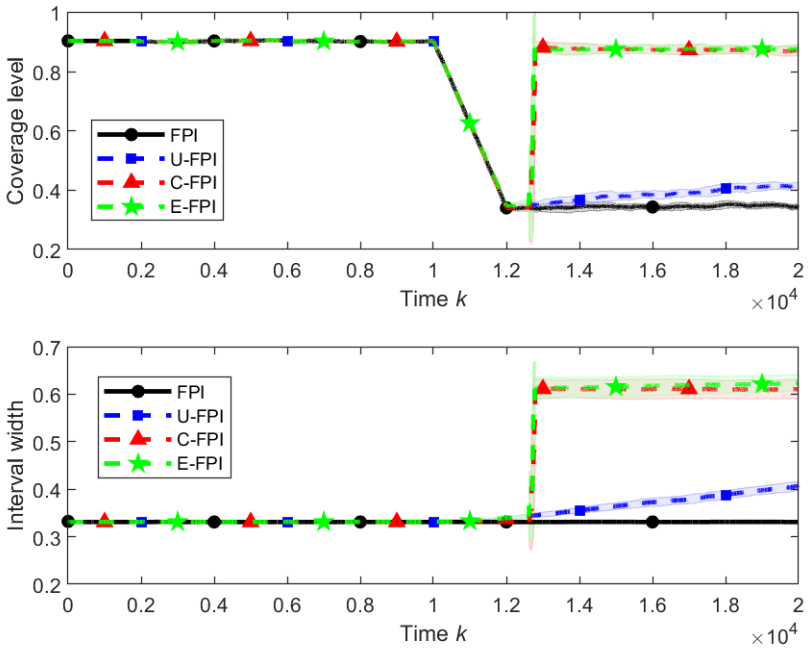


Figure 4.3: Comparison of the performance of the fuzzy prediction interval (FPI) with the effect of the proposed methods (U-FPI, C-FPI, and E-FPI) when modeling a nonlinear system with an abrupt drift. The average metric value and the corresponding interval that represents its standard deviation are obtained from 10 different simulations.

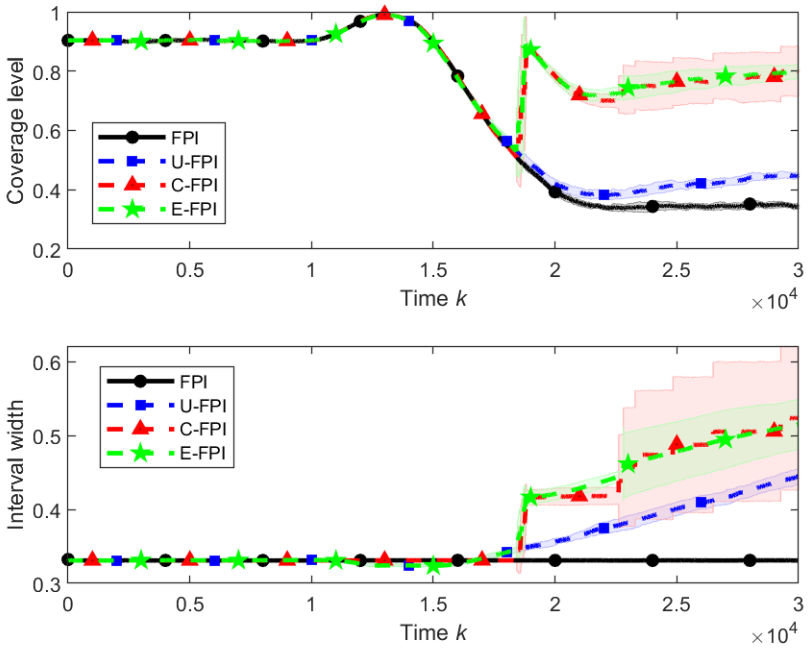


Figure 4.4: Comparison of the performance of the fuzzy prediction interval (FPI) with the effect of the proposed methods (U-FPI, C-FPI, and E-FPI) when modeling a nonlinear system with a gradual drift. The average metric value and the corresponding interval that represents its standard deviation are obtained from 10 different simulations.

Table 4.4: Metrics of the FPI methods when modeling an abrupt drift.

Metrics	**FPI**	**U-FPI**	**C-FPI**	**E-FPI**
RMSE	0.261 (0.002)	0.261 (0.002)	0.191 (0.002)	0.191 (0.002)
PICP	62.35 (0.29)	63.98 (0.26)	81.51 (0.62)	81.69 (0.56)
PINAW	18.76 (0.60)	19.71 (0.65)	24.51 (0.93)	24.72 (0.96)

Table 4.5: Metrics of the FPI methods when modeling a gradual drift.

Metrics	**FPI**	**U-FPI**	**C-FPI**	**E-FPI**
RMSE	0.240 (0.002)	0.239 (0.002)	0.159 (0.002)	0.159 (0.003)
PICP	66.01 (0.24)	68.29 (0.25)	81.18 (1.47)	81.46 (0.65)
PINAW	18.76 (0.60)	20.09 (0.59)	21.64 (0.98)	21.65 (0.60)

For Figures 4.3 and 4.4, the upper graph shows the average value of the Prediction Interval Coverage Level measured by the PICP at each instant k on the 10 simulations performed, including an interval representing the corresponding standard deviation. The lower graph presents the behavior of the average value measured at each instant k for the Prediction Interval Width on the 10 simulations performed, including the corresponding interval associated with the standard deviation.

From Figure 4.3 and Table 4.4 it can be seen that the proposed creating method of new clusters included in the C-FPI and E-FPI implementations can improve the coverage level of the interval when an abrupt drift occurs. Additionally, the inclusion of new clusters in the C-FPI and E-FPI methods reduced the modeling error compared to the original FPI. Due to the change presented by the system, the interval had to increment its width to achieve proper values of coverage level. Here, the new updating mechanism proposed for the spread parameters (included in the U-FPI and E-FPI implementations) only produced brief increments in the PICP, which shows that the proposal tried to comply with the coverage objective. However, this PICP improvement did not reach the expected value.

On the other hand, from Figure 4.4 and Table 4.5 it can be confirmed the improvement that the inclusion of the proposed methods generates in terms of the coverage level when a gradual drift occurs. In this case, the algorithm needed more time to achieve optimal behavior because the system constantly changes its parameters between the samples 10,000 and 20,000. Additionally, the behavior presented by the C-FPI method shows a higher variability of the results across the different simulations performed due to the problems of the particle swarm optimization algorithm to converge to an appropriate solution (this can be seen from the wide red interval achieved by the results of the C-FPI in Figure 4.4). By including all updating mechanisms proposed for the evolving interval, the results of the E-FPI show a tendency to converge to an acceptable interval with less variability than the C-FPI method.

Additionally, another experiment is proposed to check the effect of the window size N on the performance of the E-FPI method. According to [180], the value of $N/2$ should be at least ten times the number of parameters identified for each local model. However, theoretically, increasing the value of N will improve the model performance and, consequently, an increment of the running time of the algorithm. To evaluate this effect, we evaluated the proposed E-FPI method using different values for the time window size N. Each experiment was repeated 10 times. Figure 4.5 shows the average root mean square error, and Table 4.6 shows the average simulation time of the experiments with the standard deviation value enclosed between parentheses. For both Figure 4.5 and Table 4.6 results were reported for $N = \{500, 1000, 2000\}$.

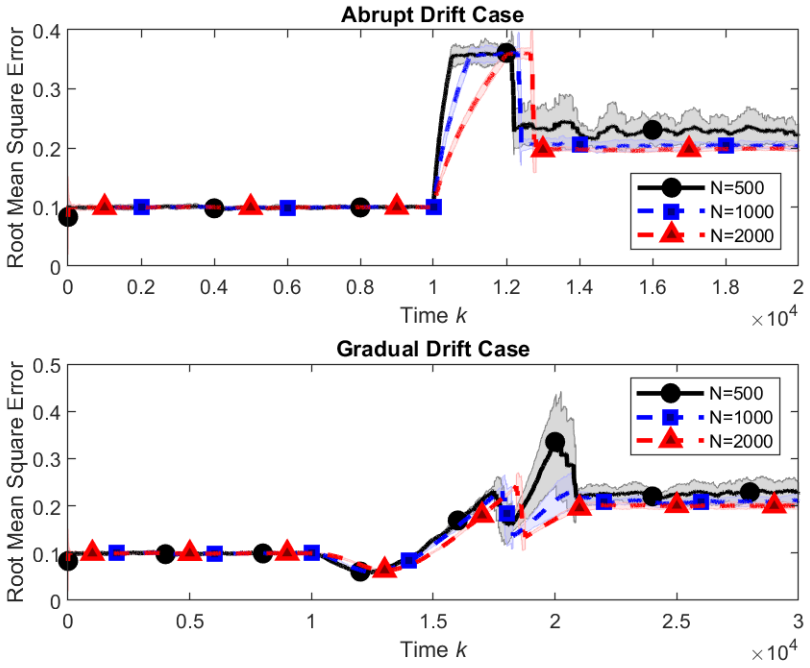


Figure 4.5: Comparison of the modeling error of the E-FPI in a time window of length N. The upper graph shows the average error measured at each instant k for an abrupt drift, while the lower graph shows the same for a gradual drift. The average metric value and the corresponding interval that represents its standard deviation are obtained from 10 different simulations.

Table 4.6: Simulation time in seconds of the E-FPI proposal when using different window lengths N. The abrupt and gradual drift results were obtained from 10 different simulations.

Window Length (N)	**500**	**1000**	**2000**
Abrupt Drift Case	57.56 (7.03)	83.26 (5.04)	151.37 (28.46)
Gradual Drift Case	112.66 (12.23)	163.23 (24.86)	236.57 (22.43)

Based on the reported results, Figure 4.5 confirms that increasing the value of N improves the model's performance by reducing the modeling error measured. However, this improvement is achieved at the cost of expanding the model delay for reacting to the system changes. Furthermore, according to Table 4.6, the increment of N means that the algorithm takes more time to process all of the mechanisms proposed, which is a consequence of the increasing running time that the model has in terms of N. These results show the trade-off between the running time and the model precision. Also, the performance of the E-FPI method presented when $N = 500$ confirms the importance of satisfying the condition of minimum data required to use the identification algorithm. In Figure 4.5, it can be seen that using a low value of N produces higher average values of prediction error and worsens the algorithm's convergence, as suggested by the larger standard deviation obtained when $N = 500$.

Finally, to finish with this synthetic case of study, a comparison between the classical

fuzzy prediction interval (FPI) and both evolving proposals presented in this work (E-FPI and SE-FPI) is performed when $N = 2000$. As done with the first comparison reported in this section, Figure 4.6 shows the evolution of the RMSE, coverage level, and interval width when the system presents an abrupt drift in its dynamics, while Table 4.7 presents the average values for the corresponding metrics measured for 10 different simulations of this experiment (here the standard deviation value is enclosed between parentheses). On the other hand, Figure 4.7 and Table 4.8 show the same results for the case of a gradual drift in the system behavior.

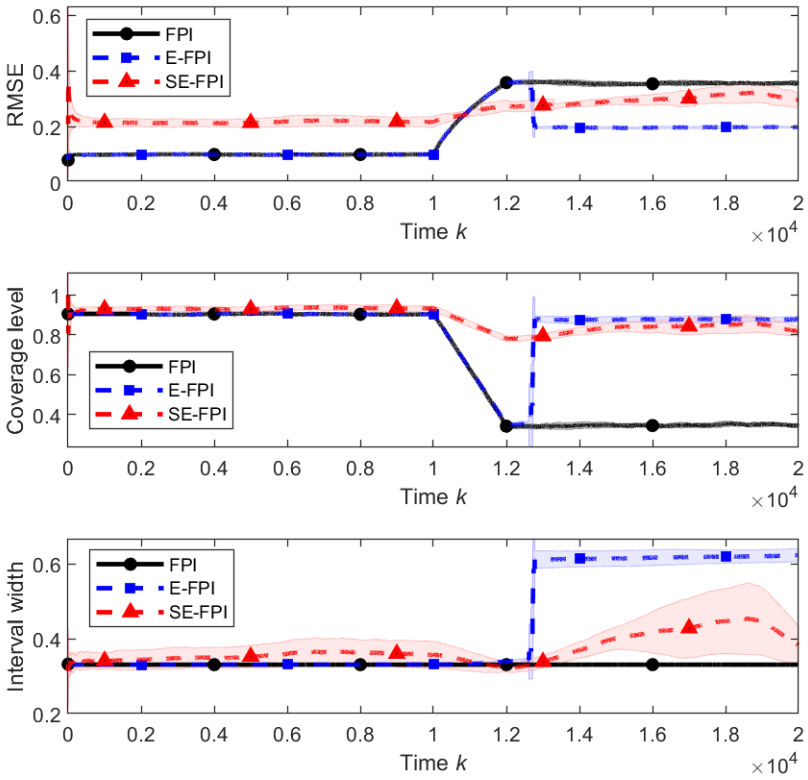


Figure 4.6: Performance comparison of the fuzzy prediction interval (FPI) with the proposed evolving methods (E-FPI and SE-FPI) when modeling a nonlinear system with an abrupt drift. The average metric value and the corresponding interval that represents its standard deviation are obtained from 10 different simulations.

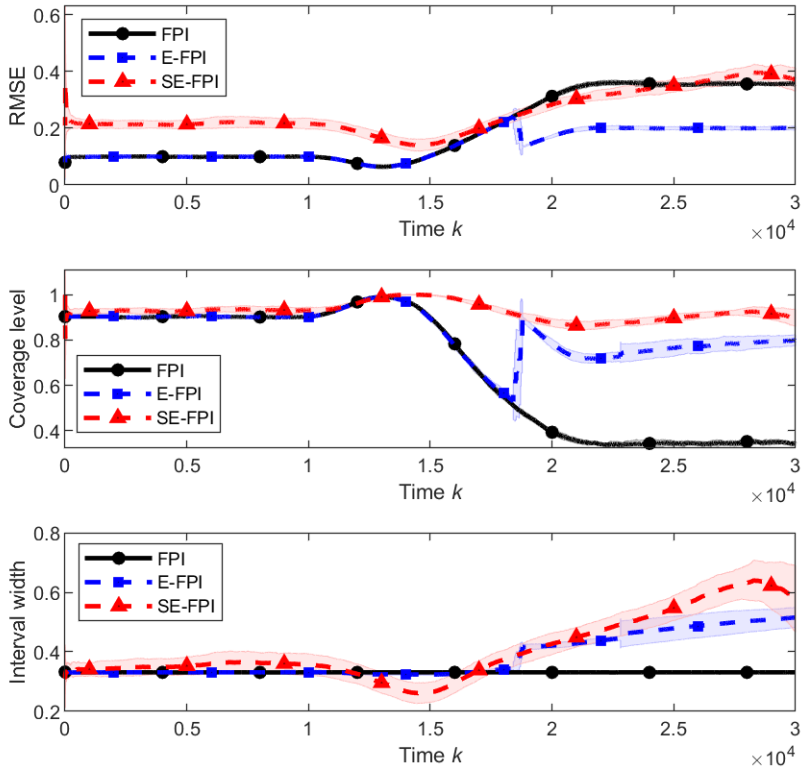


Figure 4.7: Performance comparison of the fuzzy prediction interval (FPI) with the proposed evolving methods (E-FPI and SE-FPI) when modeling a nonlinear system with a gradual drift. The average metric value and the corresponding interval that represents its standard deviation are obtained from 10 different simulations.

Table 4.7: Metrics of the proposed FPI methods when modeling a nonlinear system with an abrupt drift.

Metrics	**FPI**	**E-FPI**	**SE-FPI**
RMSE	0.261 (0.002)	0.191 (0.002)	0.161 (0.004)
PICP	62.35 (0.29)	81.69 (0.56)	74.90 (3.08)
PINAW	18.76 (0.60)	24.72 (0.96)	21.04 (1.75)

Table 4.8: Metrics of the proposed FPI methods when modeling a nonlinear system with a gradual drift.

Metrics	**FPI**	**E-FPI**	**SE-FPI**
RMSE	0.240 (0.002)	0.159 (0.003)	0.140 (0.002)
PICP	66.01 (0.24)	81.46 (0.65)	85.13 (1.91)
PINAW	18.76 (0.60)	21.65 (0.60)	23.56 (1.56)

Based on the results presented in Figures 4.6-4.7 and Tables 4.7-4.8, it can be appreciated

the same advantages of the evolving proposals identified in the first comparison made in this section. The evolving proposals can effectively adapt to the changes introduced to the system, reaching accurate values of coverage level and interval width based on the evolution presented by the prediction error. The most notable additional conclusion that can be achieved from the comparison made is that the proposed SE-FPI works as a good approximation of the first E-FPI method. The SE-FPI achieved this good behavior even though it requires less computational effort for training, thanks to avoiding the use of classical interval identification, which is based on solving a complex optimization problem. However, this saving on computational resources comes with a trade-off, which according to the standard deviations presented by the SE-FPI in Figures 4.6-4.7 and Tables 4.7-4.8, consists on some problems to achieve a consistent convergence of the results across the ten simulations tests made so far.

Additionally, as seen in Table 4.7, in some cases, the proposed SE-FPI can present more difficulties in reaching an acceptable coverage level due to the simplified way of updating the model compared to the E-FPI (which also includes a checking stage of the updated interval as part of the algorithm to avoid this issue). However, in terms of interval width and model error presented in all of the reported results, the performance of the SE-FPI was similar enough to those obtained by the first proposal (E-FPI).

Finally, to appreciate the applicability of both proposals, Table 4.9 reports the average, minimum, and maximum time taken for running each iteration of the simulation of an abrupt and gradual drift. These values are obtained from 10 simulations implemented for each method (E-FPI and SE-FPI).

Table 4.9: Time in seconds that take each iteration of E-FPI and SE-FPI when modeling an abrupt and gradual drift. The average values and their standard deviation reported in parentheses were obtained from 10 different simulations.

Interval method	**Average time**	**Minimum time**	**Maximum time**
E-FPI (abrupt drift)	0.0085	0.00027	51.5975
SE-FPI (abrupt drift)	0.0097	0.00042	0.0613
E-FPI (gradual drift)	0.0091	0.00027	97.9651
SE-FPI (gradual drift)	0.0056	0.00043	0.0479

From the values reported in Table 4.9 it can be seen that E-FPI and SE-FPI achieve a low average iteration time. This confirms the effectiveness of both proposals in terms of the average computational effort. However, there is a situation that can affect the implementability of the E-FPI, which consists of the high maximum iteration time reached for both types of drift. This high value was produced using the cluster creation and retraining mechanism for the interval, which is based on the offline identification of fuzzy prediction intervals (with a high computational effort). Despite this problematic, the E-FPI can be used without problems in cases where the sampling time is big enough (for example, for slow system dynamics), or if the offline identification procedure used to create clusters or retrain the model can be limited to a short iteration time.

In summary, all the results obtained for both types of drifts show the effectiveness of the

proposed mechanisms for performing an online adaption of the interval. Next, the second case of study considered in this work is presented with its results.

4.2.2 Modeling of solar power generation

Solar power generation data from the Multi-Good Microgrid Laboratory of the Politecnico di Milano, Milan, Italy, is the second case study to test the proposed methods. The available data corresponds to 11,688 measurements taken in 2017 and 2018, with a sampling time of one hour. The 6,168 samples from 257 days of 2017 are used to obtain the initial fuzzy prediction intervals. This data was divided in the same proportions as the previous case study: 50% for training, 25% for validation, and 25% for testing. The metrics obtained in the training and testing dataset are reported in Table 4.10. These results show that the trained interval can maintain an accurate performance for an entire year. Specifically, in this case, the testing dataset resulted in lower values of RMSE due to the lower variabilities of its measurements compared with the training data. For example, the training dataset includes data from winter to summer, while the testing dataset only includes measurements from spring. This reduction of the RMSE between both datasets provoked the appearance of lower values of PINAW while maintaining acceptable levels of PICP.

Table 4.10: Interval metrics obtained for predictions 1-step ahead. The training and testing datasets consider 50% and 25% of measurements of solar power generation from the year 2017.

Metrics	**Training**	**Testing**
RMSE	38.7529	30.2491
PICP [%]	90.425	86.144
PINAW [%]	25.420	19.837

Then, the proposed methods were tested with the other 5,520 samples corresponding to 230 days of 2018. The coverage level and the interval width are shown in Figure 4.8 to evaluate the proposed methods (E-FPI and SE-FPI).

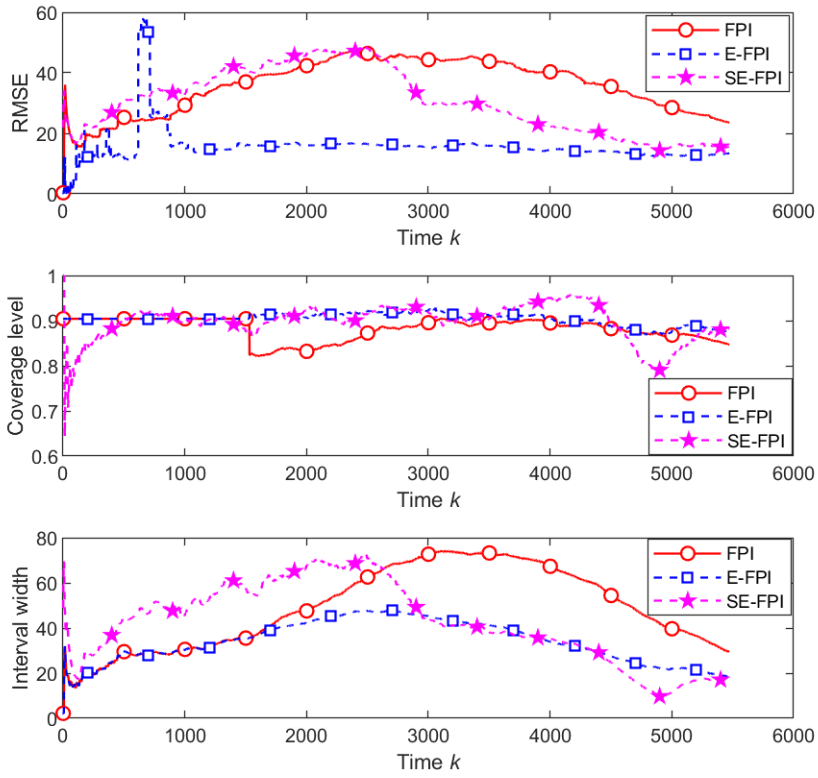


Figure 4.8: Comparison of the fuzzy prediction interval (FPI) and the proposed evolving fuzzy prediction intervals (E-FPI and SE-FPI) when modeling solar power data from 2018.

From Figure 4.8 a brief improvement can be appreciated regarding the coverage level when the proposals E-FPI and SE-FPI are used. However, a reduction of the interval width can be highlighted as the main advantage of the proposal. In this case study, the improvements in the interval performance are mainly due to the updating mechanism of the prediction model that reduced the modeling error and adjusted the spread parameters accordingly. These results show that the interval can be adequately adapted to the different behaviors presented by the measurements taken in 2018 compared with the data from 2017. Furthermore, the results shown in Figure 4.8 regarding RMSE, coverage, and interval width confirm that the proposed SE-FPI works as a good alternative to the E-FPI, reaching similar results despite requiring less computational effort.

For testing the methods with a different type of change of the signal behavior, reductions around 30% and 90% for the solar power generation are artificially introduced in the samples measured from 11:00 to 15:00. This reduction began from day 121 of the measured in 2018, i.e., from sample 2,880 in this dataset. This altered data tries to simulate the appearance of an object that obstructs solar radiation, making a shadow on the photovoltaic panels at certain hours. The behavior of the interest metrics measured for the different methods is shown in Figure 4.9 and Table 4.11.

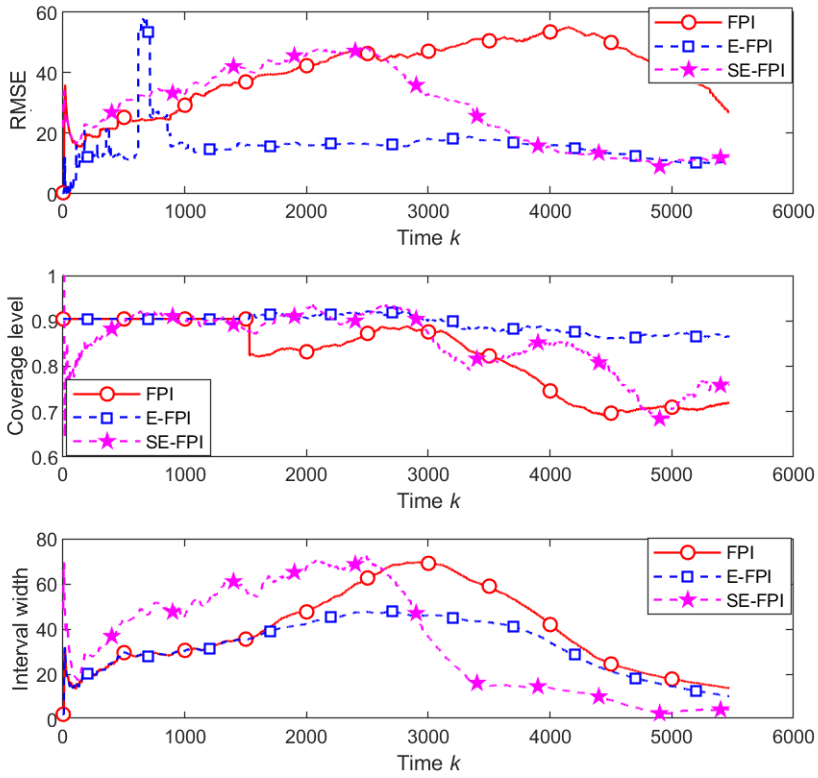


Figure 4.9: Comparison of the fuzzy prediction interval (FPI) and the proposed evolving fuzzy prediction intervals (E-FPI and SE-FPI) when modeling the 90% reduced solar power data from 2018.

Table 4.11: Performance metrics of the fuzzy prediction interval model (FPI) and the proposals (E-FPI and SE-FPI) when modeling solar power data from 2018. The three methods are compared in the original dataset and with measurements with 30% and 90% of reduction.

Tests	**Original Data**			**30% Reduction**			**90% Reduction**		
Metrics	**FPI**	**E-FPI**	**SE-FPI**	**FPI**	**E-FPI**	**SE-FPI**	**FPI**	**E-FPI**	**SE-FPI**
RMSE	36.45	18.51	15.06	36.83	18.13	14.83	41.44	18.79	16.21
PICP [%]	86.33	89.93	79.48	84.85	89.58	77.65	78.86	88.08	72.79
PINAW [%]	22.30	16.49	18.97	20.57	16.11	17.79	17.21	14.69	15.28

Based on the results shown in Figure 4.9 and Table 4.11, it can be seen the effectiveness of the proposed E-FPI to successfully correct the values of coverage level obtained by the FPI after the change in the measurements introduced from sample 2,880. Additionally, thanks to the updating mechanism of the predictive model, the modeling error presents a reduction of approximately 50% of the original RMSE measured with the FPI. Furthermore, including the new interval spread updating allowed the interval width to be reduced for the three experiments carried out, following the behavior of the new modeling error. According to Table 4.11, the reductions presented were between 2.5% and 6% of the PINAW measured in

the FPI case. On the other hand, in these results, it can be noted that the implementation of SE-FPI presented more problems in maintaining an accurate coverage level. This situation shows that the alternative SE-FPI proposal is more sensitive to greater system changes, affecting the consistency of the interval width required to reach the desired coverage level. However, it is expected that if the experiment continues, this proposed method will eventually converge to a proper interval width that meets the user's requirements.

This part of the results regarding the implementation of E-FPI was presented in the journal paper:

- **O. Cartagena**, F. Trovò, M. Roveri and D. Sáez, "Evolving Fuzzy Prediction Intervals in Nonstationary Environments," in IEEE Transactions on Emerging Topics in Computational Intelligence, (Early Access), doi: 10.1109/TETCI.2023.3296486. **[Journal Impact Factor 2022: 5.3 - Q2 Computer Science - published]**.

Considering that all of these simulated results confirm the advantages previously mentioned for the proposed methods compared to the classic FPI, this chapter continues with the report of the experimental implementations carried out in this work.

4.3 Experimental results of evolving fuzzy prediction intervals

An experimental study is proposed based on a real heat exchanger pilot plant [191] installed in a laboratory of the University of Ljubljana, Slovenia. The main purpose of this kind of plant is to transfer heat between two different water circuits. In this case, the system process is composed of a primary close circuit with a hot water flow, in which its temperature (T_h) is regulated by a thermostat. The secondary open circuit corresponds to the cold water flow from the local water supply network, which has its temperature value (T_{sp}). A summary of this system is presented in Figure 4.10.

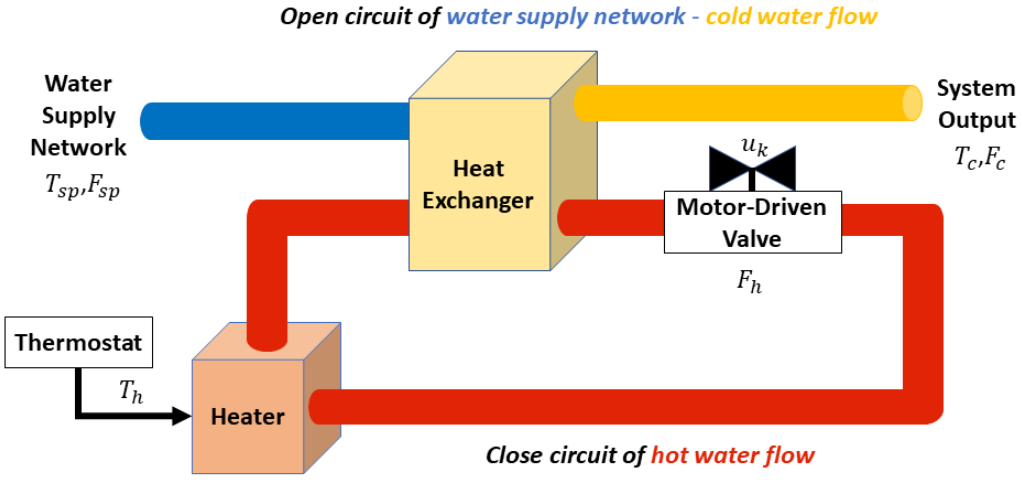


Figure 4.10: Diagram of the real heat exchanger plant considered for the experimental tests.

The system's output is the temperature (T_c) of the secondary circuit after the heat transfer from the hot water flow. The system's operator can handle this output value by manipulating the flow of hot water (F_h) using a motor-driven valve. In the real plant, the manipulated variable corresponds to the electrical signal (u_k) received by the motor-driven valve, which can have values in the range of 4-20 [mA]. Note that the flow value of the secondary circuit (F_c) is considered constant and with the same value (F_{sp}) that is provided by the local water supply network.

This work presents two experiments performed with the heat exchanger plant to test the proposed evolving fuzzy prediction interval and the fault detection algorithm.

In the first experiment, measurements from the real plant's output are taken after using a varying input signal $u(k)$. Furthermore, the signal $u(k)$ has to reach the plant's different operation modes for proper system identification. Based on this idea, this work used an APRBS signal and stepped functions as the values for the signal $u(k)$. Here, 7,750 output measurements extracted from the plant with a sampling time of 4 seconds were used for directly applying the learning phase of the evolving prediction interval algorithm. Next, this work proceeds with the validation of the prediction interval achieved at the end of the learning phase by comparing the model outputs with 7,000 new measurements taken from the system. This validation is carried out by evaluating the performance of the prediction interval model reached at the end of the learning phase, but without considering any model updating from the evolving mechanisms, i.e., after finishing the learning phase, the final interval is tested with a fixed structure and parameters using a different set of measurements. The evaluation of the prediction interval models will be done in terms of the modeling error (RMSE), coverage level (PICP), and interval width (PINAW).

Based on these metrics, the validation process of the interval provided by the learning phase of the algorithm consists of verifying that the coverage level is close to its expected value in a different dataset while the interval width results are consistent with the prediction errors achieved by the predictive model.

In the second experiment, the predictive interval model achieved in the learning phase is used for implementing the fault detection algorithm. In this case, the model structure and its parameters are fixed, so no evolving mechanisms are used in this experiment to prevent the adaptation of the interval to the changes presented by the heat exchanger plant. Here, 6,000 output measurements with a sampling time of 4 seconds were taken from the system.

During the running process of the experiment, two different changes were introduced in the operation of the plant. First, the hot water flow circuit's reference temperature decreased from 80°C to 60°C over 90 minutes. The second change was the reduction of the openness of the hot water valve, which reduced the hot water flow in the heat exchanger for 90 minutes.

An additional experiment was carried out to complement the previous test to be performed with real faults, which consists of introducing other faulty scenarios. However, due to the characteristics of the real heat exchanger plant, it was impossible to introduce new kinds of faults in a controlled environment that differed from those that had already been reported. Thus, the additional experiment was implemented using 6,000 output measurements taken from the heat exchanger plant during its normal operation, multiplying these values with

some factors during two different periods of time.

The proposed interval-based fault detection method is evaluated in this experiment in terms of the algorithm's detection delay, precision, and recall rates. Based on [192], precision and recall are formally defined as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}, \tag{4.6}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \tag{4.7}$$

The following section presents the experimental results of both experiments carried out in this part of the work.

4.3.1 Modeling of a heat exchanger plant

The first stage of the experimental test is the application of both proposals for the evolving fuzzy prediction interval using the measurements obtained from the heat exchanger plant after applying an APRBS signal and a staircase function as an input of the system.

For this experiment, the settling time of the heat exchanger is estimated first from a test where the plant is excited with a stair-case signal. The corresponding result is presented in Figure 4.11 and shows that the system's settling time can be estimated at around 100 seconds (25 samples if a sampling time of 4 seconds is considered).

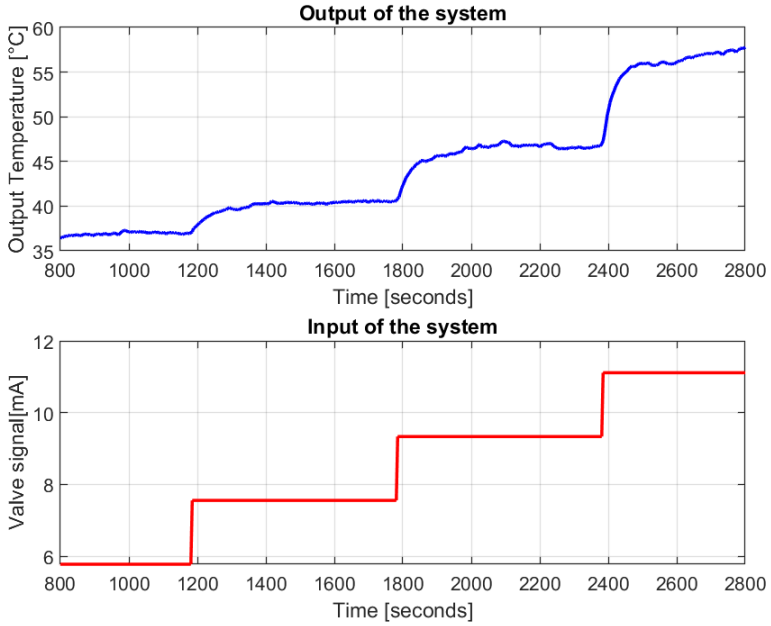


Figure 4.11: Response of the heat exchanger plant when excited with a staircase function.

Based on this information about the heat exchanger, the proposed learning phase is implemented using a training dataset, where the model parameters are obtained by comparing the 1-step predicted error.

Figure 4.12 shows the experiment's measurements considered as the training dataset for applying the learning phase of the proposed evolving prediction interval algorithms.

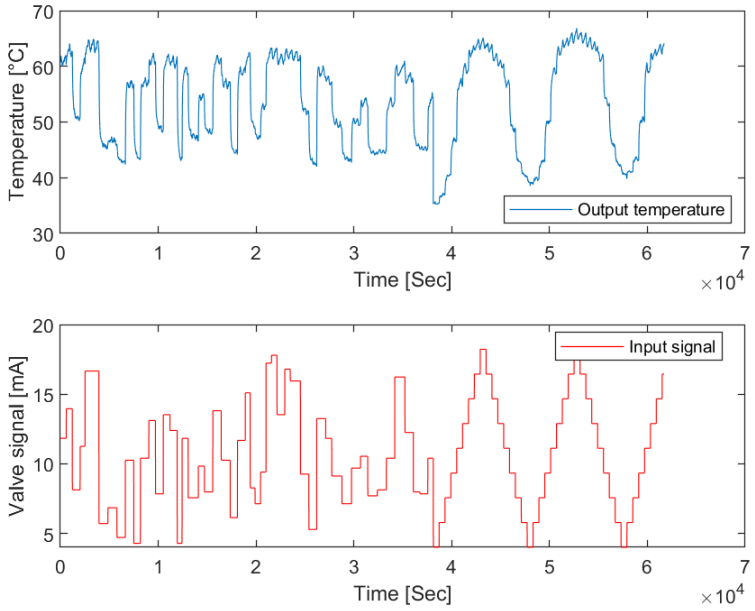
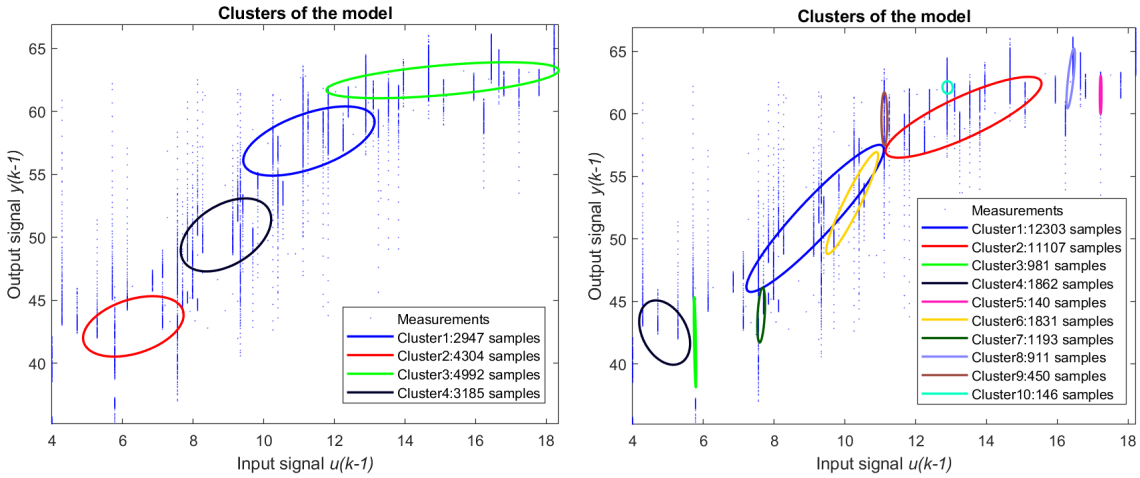


Figure 4.12: Measurements of the input variable (valve signal) and the corresponding output (output temperature) used for training the interval.

The training process of the E-FPI and the proposed learning phase of the SE-FPI method are implemented using the following user-parameters: first, a time window of past $N_{cov} = 500$ samples (equal to 2,000 seconds) was used to estimate the coverage level, while the prediction steps $N_{pred} = 1$ (4 seconds) and $N_{pred} = 30$ (120 seconds) were used for implementing the prediction intervals. The threshold for the cluster distance $d_{TH} = 3$ and the maximum width allowed for each cluster $w_{max} = 2$ were selected in a heuristic manner based on the results of preliminary tests. For the expected coverage level required for the evolving interval and the maximum overlapping ratio, the values $\hat{\alpha} = 0.95$ and $\kappa_{join} = 1.2$ were selected based on the typical values previously used in the literature. Figure 4.13 shows the final clusters achieved by the two proposed evolving algorithms at the end of the learning phase.



(a) Proposed E-FPI (b) Proposed SE-FPI

Figure 4.13: Resulting clusters of the model at the end of the learning phase. The shape of the clusters is constructed from the eigenvectors of the covariance matrix of each cluster. The blue points represent the measurements taken from the system. Figure 4.13a shows the final clusters obtained when applying the E-FPI method, while Figure 4.13b does the same for the case of the SE-FPI implementation.

Figure 4.13 shows that at least the shapes of the clusters obtained at the end of the learning phase follow the distribution of all the measurements collected from this first experiment. Clusters that finished with low quantities of samples (below N=500) in the SE-FPI case will be discarded in the following experimental tests.

At this point, the resulting interval obtained at the end of the learning phase has to be tested with a fixed structure and parameters, i.e., without using the adaptation mechanisms from the evolving systems field. Therefore, the measurements extracted from the system after applying a staircase function as input are used for this second stage of experimental tests. Figure 4.14 shows the measurements of the experiment used for validating the proposed evolving prediction interval algorithms.

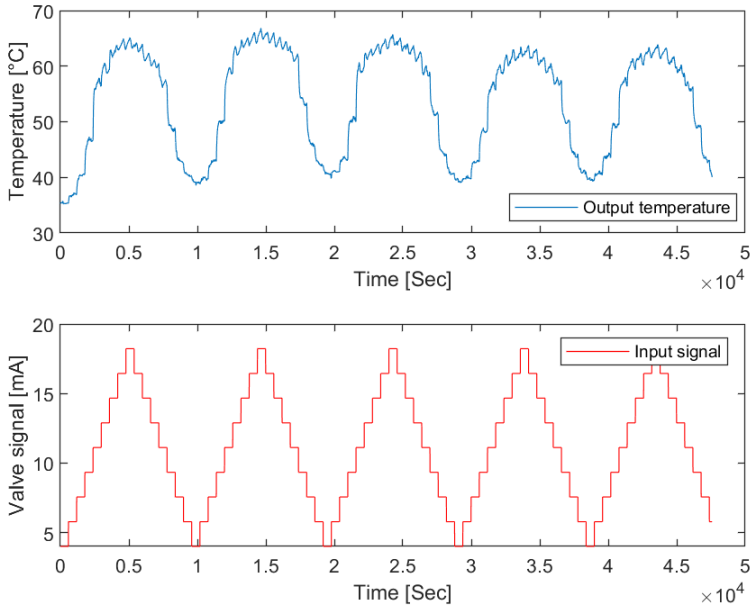
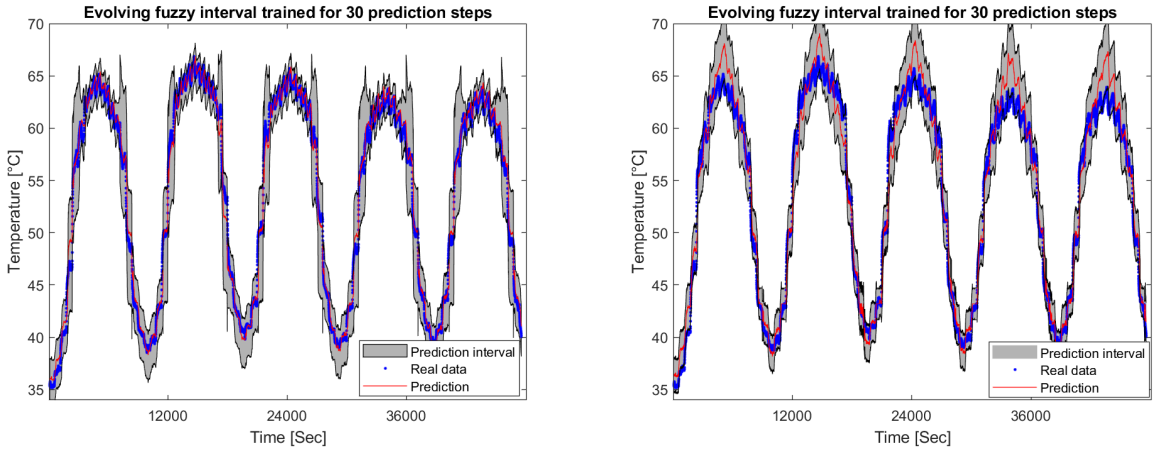


Figure 4.14: Measurements of the input variable (valve signal) and the corresponding output (output temperature) used for validating the interval.

The implementation of this validation test achieved the following results in terms of the 30-steps predicted output, shown in Figure 4.15 for each proposed algorithm. Additionally, Table 4.12 shows the performance metrics measured for both interval methods in the validation test.



(a) Proposed E-FPI (b) Proposed SE-FPI

Figure 4.15: The 30-step prediction interval output measured during the validation test. Figure 4.15a compares the real measurements (blue points) with the prediction interval obtained when applying the E-FPI method, while Figure 4.15b does the same for the case of the SE-FPI implementation.

Table 4.12: Performance metrics of the interval measured in the validation test of the real heat exchanger plant experiment.

Metrics	**1-step prediction**		**30-steps prediction**	
	E-FPI	**SE-FPI**	**E-FPI**	**SE-FPI**
RMSE	0.1100	0.1046	1.0792	1.6417
PINAW	0.0136	0.0205	0.1850	0.1622
PICP	0.9692	0.9854	0.9814	0.9207

From the results shown in Figure 4.15, it can be appreciated that the interval provided by the proposed algorithms achieves accurate modeling of the output temperature. However, the SE-FPI presented some problems when trying to contain the measurements in some moments of the experimental test. This situation occurs due to the lower temperatures presented by the heat exchanger, which were probably affected by the change of external factors such as the ambient temperature. But, despite these situations, Table 4.12 shows that the general performance of the intervals over the whole experiment achieved acceptable results, with a coverage level close to the expected value of 95% for the case of 30 prediction steps.

Based on metrics reported in Table 4.12, the good performance achieved by the SE-FPI, which resulted close to the E-FPI's performance, but using a significantly less amount of computational resources (due to avoiding the use of complex optimization algorithms for determining the initial values of the interval width) should be highlighted. Additionally, the SE-FPI achieved competitive results in terms of the interval width despite having a worse base predictive model in terms of the RMSE measured in this validation experiment.

Additionally, an important result that can be extracted from this initial experiment is related to the timings of the SE-FPI algorithm handle during the learning phase. Figure 4.16 shows the time spent by the algorithm for each iteration, compared to the number of clusters the model possesses at each instant.

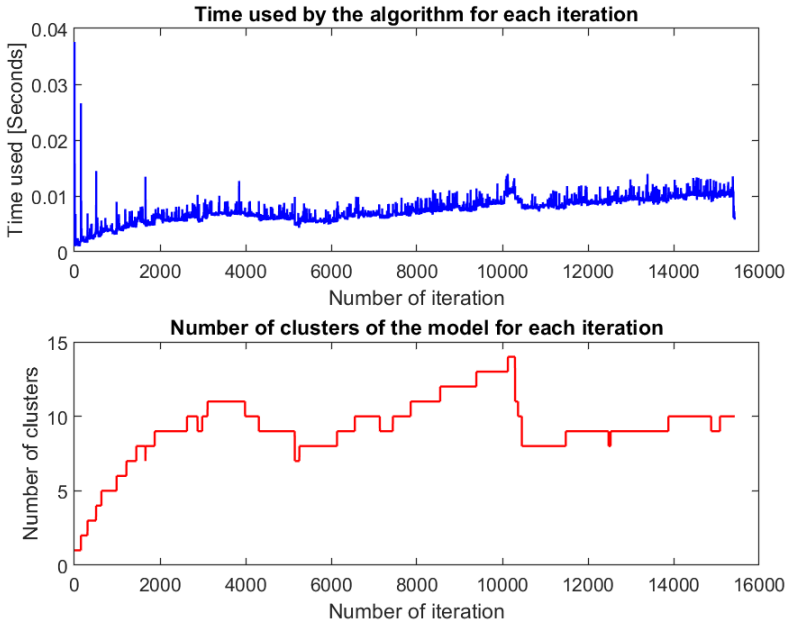


Figure 4.16: Time for running each iteration of the proposed SE-FPI algorithm during the learning phase. The iteration time is compared with the complexity of the model, represented by the number of clusters at each instant.

From Figure 4.16 it can be seen that the proposed algorithm can complete each iteration in a short period despite the increment of the model complexity due to the higher number of clusters. The average iteration time measured for the SE-FPI during the learning phase was 0.0073 seconds, with a standard deviation of 0.002. This result probes the viability of applying the SE-FPI in the real heat exchanger plant because the iteration time is considerably lower than the system's sampling time of 4 seconds.

On the other hand, the applicability of the E-FPI has the risk of not achieving a proper iteration time. This risk is present in the E-FPI because the interval identification process included in the cluster creation and retraining mechanisms can take longer than the heat exchanger sampling time, according to the results reported in Table 4.9 and discussed in the previous simulated experiment. For checking the aplicability of the E-FPI in this specific case, Figure 4.17 shows the time spent by the E-FPI algorithm for each iteration, compared to the number of clusters the model possesses at each instant, as done previously with the SE-FPI method.

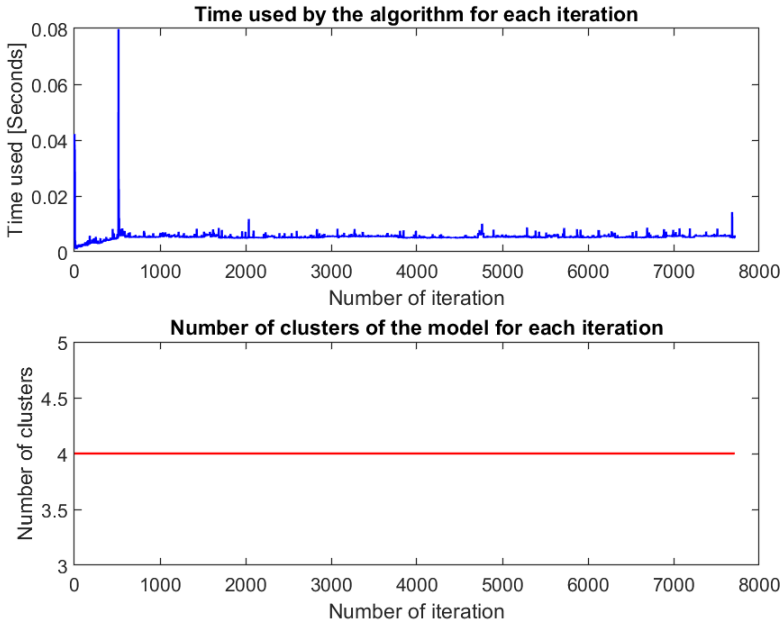


Figure 4.17: Time for running each iteration of the proposed E-FPI algorithm during the learning phase. The iteration time is compared with the complexity of the model, represented by the number of clusters at each instant.

In Figure 4.17 it can be seen that the E-FPI maintained a low iteration time throughout the experiment with the heat exhanger data, even considering the the instants where the algorithm took more time to run (specifically from Figure 4.17 the two instants when a single iteration of the E-FPI took 0.04 and 0.08 seconds). Furthermore, the average iteration time measured for the E-FPI during this experiment was 0.0053 seconds, with a standard deviation of 0.0013, which was considerably less than the time used by the SE-FPI. This result is because the E-FPI did not use the creation of clusters and the retraining mechanism during the experiment, as can be seen from Figure 4.17, where the E-FPI maintained a constant number of clusters for all iterations of the algorithm. This confirms that in this especific case the E-FPI can be used without problems while the cluster creation mechanism is not needed, i.e. if the heat exchanger does not present an important change in its behavior.

In this part of the work, an additional experiment was carried out to test the robustness of SE-FPI. The results that show that the SE-FPI is not robust enough to handle unexpected disturbances in the learning phase if the user wants to make further predictions of more than one step ahead are presented in the Annex B.

This part of the results regarding the implementation of the SE-FPI was presented in the journal paper:

- **O. Cartagena**, M. Ožbot, D. Sáez, I. Škrjanc, "Evolving fuzzy prediction interval for fault detection in a heat exchanger," in Applied Soft Computing, Volume 145, 110625, 2023, doi: 10.1016/j.asoc.2023.110625. **[Journal Impact Factor 2022: 8.7 - Q1 Computer Science - published]**.

With the quality of the obtained interval already validated and the identification of the main advantage of the SE-FPI over the E-FPI in terms of the applicability to the heat exchanger in this experiment, this work proceeds with the following experimental tests for testing the proposed fault detection algorithm. The results of these new experiments are reported below.

4.3.2 Testing of fault detection in a heat exchanger plant

The last stage of the experimental tests is the application of a fuzzy prediction interval in implementing a new model-based fault detection algorithm. In this section, the algorithm testing is carried out using a fixed interval obtained at the end of the learning phase of the SE-FPI proposal. The proposed algorithm is tested over the heat exchanger measurements shown in Figure 4.18. The results of the fuzzy prediction interval in terms of the output interval and the failure index are shown in Figures 4.19 and 4.20, respectively.

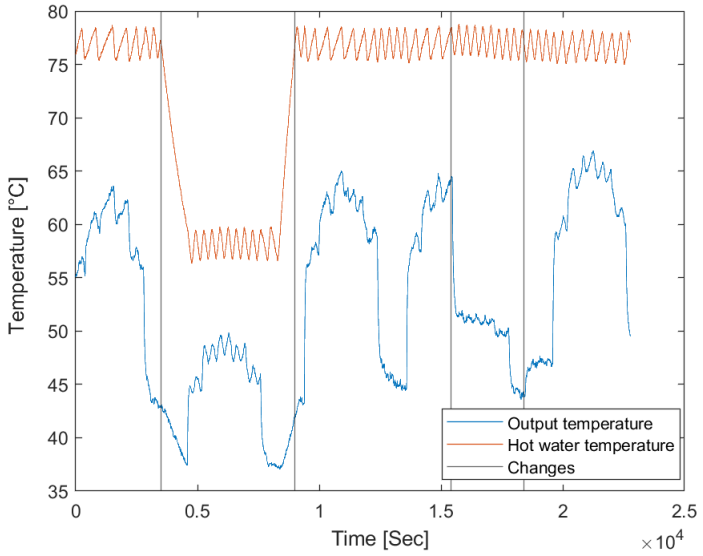


Figure 4.18: Measurements of the output and the hot water temperatures used to implement the interval-based fault detection algorithm.

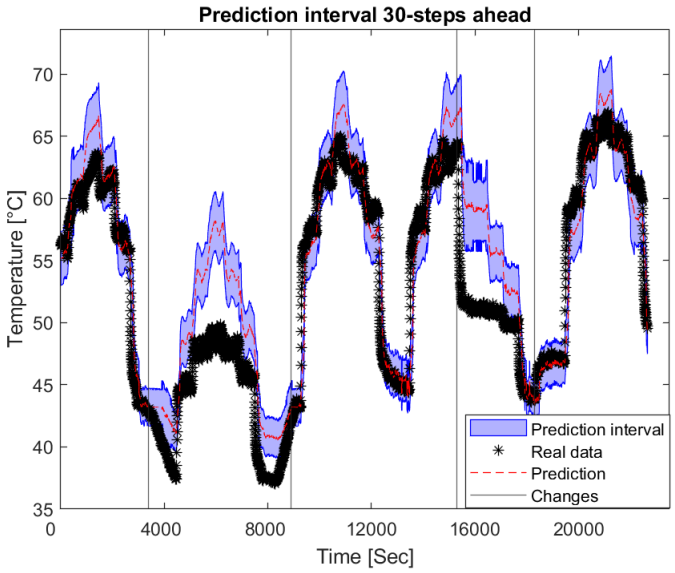


Figure 4.19: The 30-step prediction interval output measured during the fault detection testing experiment. The black lines represent the instants where the heat exchanger dynamics changed.

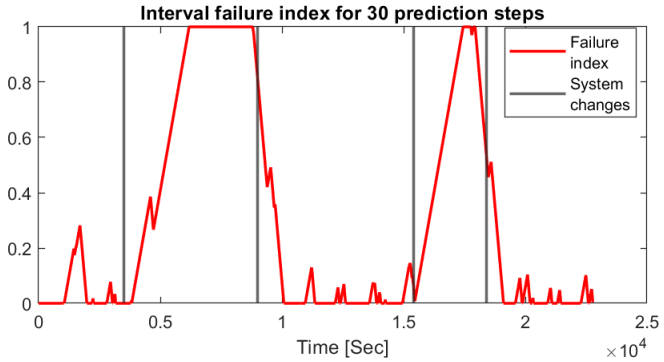


Figure 4.20: Interval failure level estimated at each instant k by considering a time windows of size $N = 500$. The black lines represent the instants where the heat exchanger dynamics changed.

The result presented in Figure 4.19 shows how the fixed interval obtained at the end of running the SE-FPI learning phase fails when trying to model the heat exchanger under the changes introduced in its system dynamics. This failure of the interval in the task of containing the real measurements of the systems produced the increments of the index shown in Figure 4.20. This result confirms that the index proposed in equation (3.61) would help estimate the occurrence of faults.

The proposed interval-based fault detection algorithm is implemented by establishing a threshold value of 0.5 for the failure index, representing that at least 50% of the samples within the time window (equal to 1,000 seconds in the performed tests) should be outside the interval for detecting a fault. The value of the user parameters are the same as those

used in the previous stage when the proposed SE-FPI was tested, i.e., $N_{cov} = 500$, $N_{pred} = 1$ (4 seconds) and $N_{pred} = 30$ (120 seconds), $d_{TH} = 3$ and $\hat{\alpha} = 0.95$. The results of the proposed algorithm are compared with a model-based fault detection algorithm based on the integration of the MSE (also called RMS Evaluation)[188], which follows the use of the evaluation function (3.60) presented in Section 3.4, and detects the fault when the integrated error over a time-window surpass a threshold. In this section, the RMS Evaluation method is implemented for the 1- and 30-step prediction errors.

Additionally, these results are compared with the implementation of two different fault detection methods, one based on the parameter clustering, as implemented in [193] (where clusters were implemented over several estimations of system parameters and the fault is detected when the new set of parameters is far from the existing clusters), and the other based on the evolving clustering applied over the input-output space, as presented in [194] (where a fault is detected when the creation mechanism of the evolving algorithm such as presented in Section 3.2.3 is activated). The results are shown in Figure 4.21, and the value of the performance metrics defined in equations (4.6)-(4.7) are reported in Table 4.13.

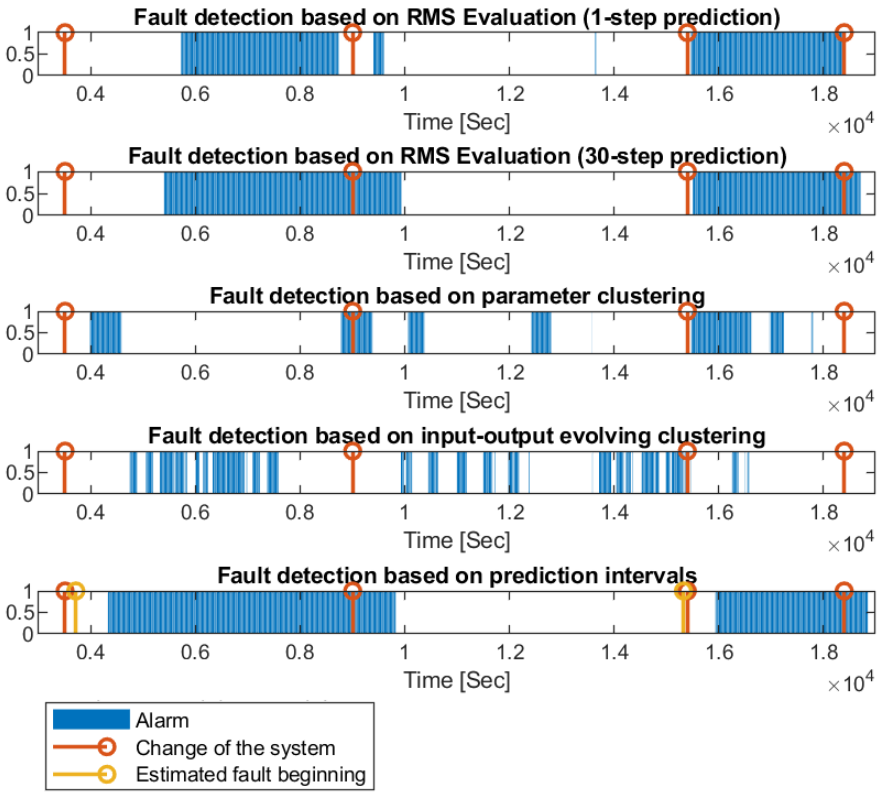


Figure 4.21: Comparison of the alarms produced by the different fault detection methods tested over the first fault scenario of the heat exchanger plant. The first graph shows the activation of alarms for the RMS evaluation method applied over the 1-step prediction, while the second graph does the same for the 30-steps prediction case. Third graph reports the alarm activation produced by the method based on the clustering of the parameter vector, and the fourth graph shows the alarms generated by the evolving clustering method applied over vectors that contain input-output measurements of the system. Finally, the last graph shows the alarms activated and the fault beginning estimated by the proposed interval-based method, which was tested with the 30-steps predictions.

Table 4.13: Fault detection metrics achieved by the five methods tested. These metrics were obtained considering the experiment with the fault due to the decrease in the reference hot water (Fault 1) and the fault due to the reduction of the oppeness of the hot water valve (Fault 2).

Metrics	**Delay Fault 1 [Sec]**	**Delay Fault 2 [Sec]**	**Precision**	**Recall**
1-step RMS Evaluation	2225	69	96.30%	69.79%
30-steps RMS Evaluation	1893	109	83.79%	76.38%
FD based on parameter clustering	480	80	67.30%	26.64%
FD based on input-output evolving clustering	1244	72	50.05%	24.14%
Interval based FD (Proposal of this work)	821	541	84.74%	83.91%

From the alarms shown in Figure 4.21, it can be appreciated that the proposed algorithm detected faster the first fault of the experiment, which consisted of the slow change of the hot water temperature. Furthermore, the information provided by the interval failure index allowed the algorithm to estimate the instant when the fault started. However, the algorithm presents a delay in the alarm activation because the proposed algorithm relies on comparing the interval and the future measurements of the system's output which becomes known 30 steps later. According to the values shown in Table 4.13 the delay of the proposal was considerably less than the RMS Evaluation methods in the case of the first fault (detecting with the proposal was over 1,000 seconds faster than the other methods based on RMS evaluation). The only exception is the result achieved by the fault detection method based on the parameter clustering, which presented the fastest detection, but with lower performance metrics because this method mainly detected when a change occurred and had a problem with determining the time when the fault was active.

In the case of the second fault, the proposal's delay resulted worse than those achieved by the other tested methods. Based on the precision and recall values reported in Table 4.13, the proposal maintained more consistent results during the experiment for both metrics. On the other hand, the RMS evaluation methods decreased their performance in terms of the recall value compared with the precision metric, which means that these methods presented higher numbers of false negatives during the experiment. Finally, the clustering-based fault detection algorithms reported in this experiment presented problems for achieving good performance metrics, due to the situation of not being able to maintain the alarm activated when a change was detected in the system, considerably increasing the number of false negatives. Also, the clustering-based fault detection algorithms had a larger number of false positives during the normal operation period of the plant between the faults, showing that these methods were more sensible to the noise signal presented in the measurements and the small changes in the plant behavior due to external disturbances.

With the purpose of complementing the previous results, an additional test was implemented for evaluating the methods in a different faulty scenario. Considering the limitations of the real heat exchanger plant and the difficulty of introducing other kinds of controlled faults, this additional experiment consists of modeling a case where the sensors of the system present a malfunction behavior. This situation was simulated by multiplying the output measurements taken during the plant's normal operation by 0.9 and 0.4 during two different periods. Figure 4.22 shows the measurements of this second experiment used for testing the fault detection algorithms, while Figure 4.23 presents the interval results achieved for 30-prediction steps.

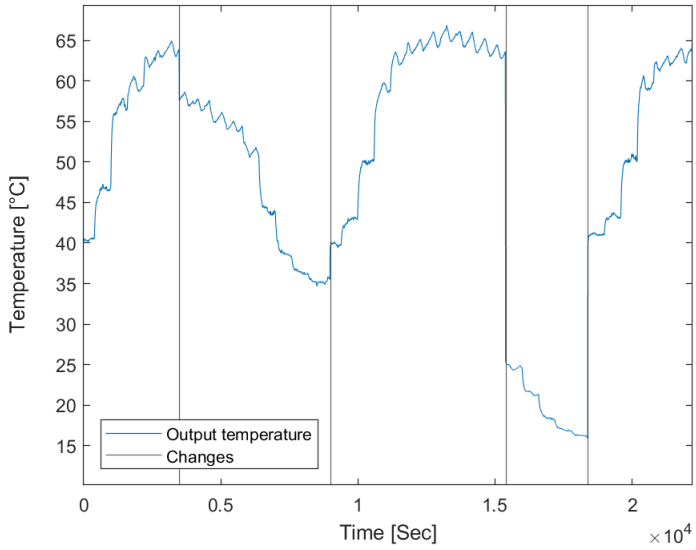


Figure 4.22: Measurements of the output used to implement the second evaluation test for the fault detection algorithms. The first fault corresponds to reducing measurements to 90% of the original value. After that, the second fault is the reduction of measurements to 40% instead.

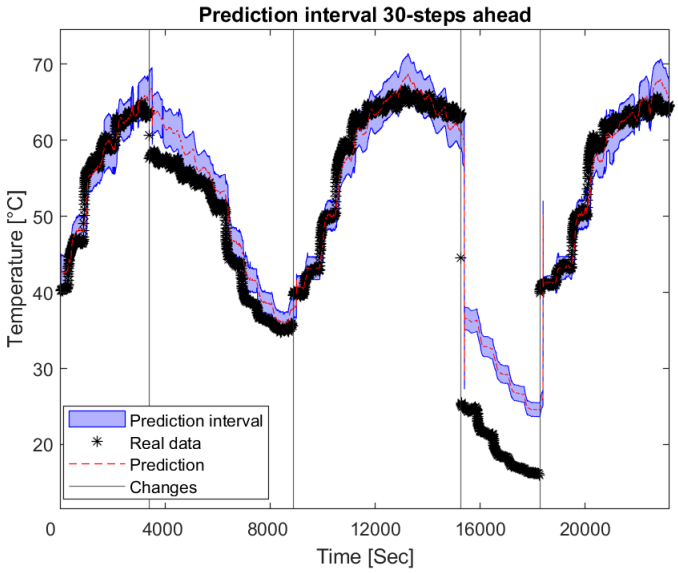


Figure 4.23: The 30-step prediction interval output measured during the fault detection testing experiment. The black lines represent the instants where the heat exchanger dynamics changed.

The results achieved by all of the fault detection methods are shown in Figure 4.24, and the value of the performance metrics previously defined in equations (4.6)-(4.7) are reported in Table 4.14.

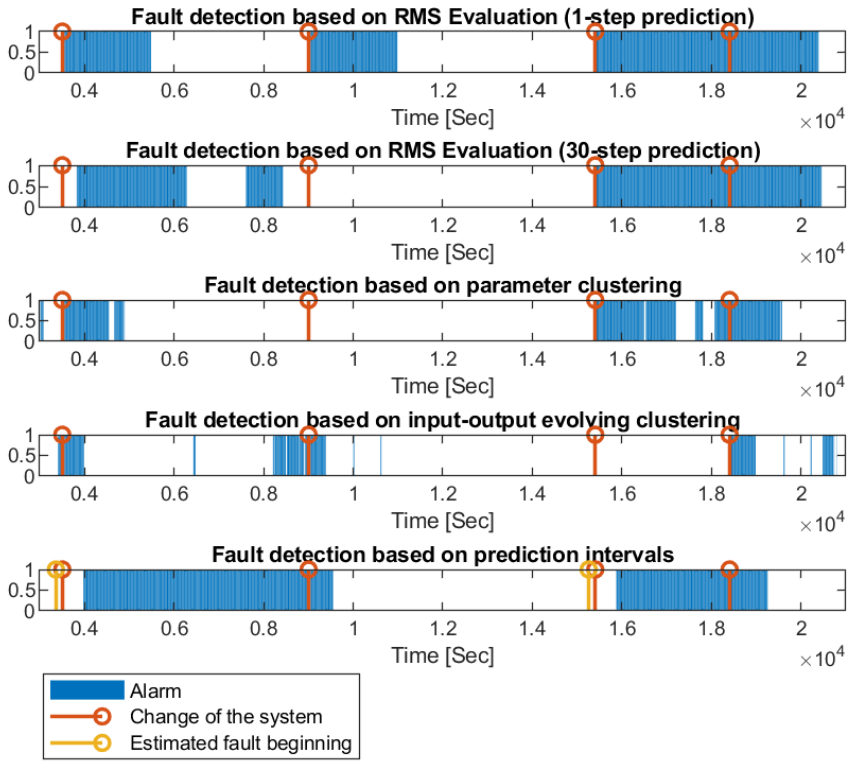


Figure 4.24: Comparison of the alarms produced by the different fault detection methods tested over the second fault scenario of the heat exchanger plant. The first graph shows the activation of alarms for the RMS evaluation method applied over the 1-step prediction, while the second graph does the same for the 30-steps prediction case. Third graph reports the alarm activation produced by the method based on the clustering of the parameter vector, and the fourth graph shows the alarms generated by the evolving clustering method applied over vectors that contain input-output measurements of the system. Finally, the last graph shows the alarms activated and the fault beginning estimated by the proposed interval-based method, which was tested with the 30-steps predictions.

Table 4.14: Fault detection metrics achieved by the five methods tested. These metrics were obtained considering the experiment with the fault due to the decrease of the output to a 90% (Fault 1) and 40% (Fault 2) of its original value.

Metrics	**Delay Fault 1 [Sec]**	**Delay Fault 2 [Sec]**	**Precision**	**Recall**
Interval based FD	499	495	86.76%	88.85%
1-step RMS Evaluation	3	3	65.57%	58.87%
30-steps RMS Evaluation	351	19	91.23%	73.93%
FD based on parameter clustering	3	3	83.19%	41.69%
FD based on input-output evolving clustering	15	3011	53.13%	14.78%

From the alarms presented in Figure 4.24 and the delays reported in Table 4.14, it can be appreciated that this time the methods which rely on the evaluation of the 1-step error and clustering presented a faster response against this new faults. This occurred because the faults considered in this experiment were introduced abruptly in the measurements. On the other hand, the methods that require to evaluate the system behavior for the 30 prediction steps (Interval based and 30-step RMS evaluation) maintained a higher delay because of their requirement to wait for the future measurements of the system to evaluate the prediction error and the interval. Regarding the performance metrics and the occurrence of false negatives for the first fault, the 1-step RMS evaluation and the clustering-based fault detection presented this problem because of the small change representing the introduced fault of multiplying the output by 0.9.

On the other hand, in the case of the second fault, almost all of the tested methods presented good results because this fault represented a major change in the system behavior. However, in this case, the bad response that the fault detection based on evolving clustering had to be highlighted. This situation occurred by the specific circumstances of this method, where the detection of the second fault was strongly affected by the previous presence of the first fault. So, in this case, the new clusters created due to the first fault interacted with the original clusters of the normal operation thanks to the evolving mechanisms, and their evolution increased the complexity of detecting the second fault. Thus, analyzing the overall results of this second test, the proposed interval-based fault detection method presented more consistent results in terms of the performance metrics, the same as occurred for the first test with the real faults, but with the trade-off of having a major delay in the detection.

Considering the two experimental tests performed with faults, the proposal is more robust for detecting, thanks to its consistent results with different types of faults. However, the operator has to consider that the proposal introduces delays in the alarms that the algorithm can not avoid. Despite this, the interval-based fault detection algorithm can estimate when the fault started as soon as the alarm is activated.

This part of the results regarding the implementation of the interval-based fault detection algorithm was presented in the journal paper:

- **O. Cartagena**, M. Ožbot, D. Sáez, I. Škrjanc, "Evolving fuzzy prediction interval for fault detection in a heat exchanger," in Applied Soft Computing, Volume 145, 110625, 2023, doi: 10.1016/j.asoc.2023.110625. **[Journal Impact Factor 2022: 8.7 - Q1 Computer Science - published]**.

4.4 Discussions

This chapter implemented simulation and experimental results to evaluate the previous interval methods, testing the two evolving interval proposals (E-FPI and SE-FPI) and the novel interval-based fault detection algorithm.

Based on the evaluation of intervals, the simulation results coincide with the theoretical analysis presented in Chapter 3, concluding that the sequential version of the fuzzy prediction interval based on fuzzy numbers is a good choice to implement the proposed evolving

intervals. This decision is taken based on the advantages of this interval method, which provides a good interpretability of the uncertainty associated with each input variable, due to the model structure (spread values considered for each parameter in the consequences of the fuzzy model). Additionally, the interval method based on fuzzy numbers produces a more straightforward implementation of the proposals thanks to the lower complexity of the interval identification compared to the other alternatives considered in this work (joint supervision method and type-2 fuzzy models).

Considering the results achieved by the proposals, this chapter confirms the applicability of E-FPI and SE-FPI to proper modeling of time-variant nonlinear systems with their uncertainty. Furthermore, based on the implementation of the E-FPI proposal, the interval performance in terms of model error, interval width, and coverage level shows good model behavior thanks to all of the mechanisms included for checking the accuracy of the updated interval.

On the other hand, the results of the SE-FPI show that this second proposal can adequately approximate its performance to that achieved by the E-FPI despite having a simplified method of performing the model updating, and the elimination of the additional mechanisms for checking the updated interval. However, this reduction of the interval complexity, which arises as a significant advantage over the first proposal, comes with a reduction of the final performance in some cases of the reported results, evidence of the trade-off between the model complexity and the robustness of its precision over different kinds of changes that the modeled system may present.

Finally, this chapter presented the results of the novel interval-based fault detection algorithm, which produced competitive performance metrics compared to some related methods, such as the RMS evaluation, the fault detection algorithm based on parameter clustering, and the fault detection based on evolving clustering. However, the use of prediction intervals introduced some delays in alarm activation because the proposed methodology has to wait until the real future measurement of the predicted output to start the interval evaluation. Also, the proposed algorithm has to wait until the interval consistently fails to contain the system measurements to detect a fault. Despite these problems with the delays in the activation of the alarms, at the end of the experimental test, the proposed fault detection method produces an accurate posteriori conclusion of when the fault started and finished, which can still be helpful for the system operator.

5 Conclusions

This thesis considered different methods and algorithms to develop a novel design of evolving fuzzy prediction intervals. The first stage of this work consisted of a review of the state-of-the-art of the topics of fuzzy prediction intervals and evolving intelligent systems. From this review, an evaluation of prediction interval methods is performed using nonlinear systems and measurements of signals as benchmark cases.

The first evaluation of these methods yielded as the main conclusion the advantages of fuzzy prediction intervals for modeling different types of nonlinear dynamical systems with uncertainties. This idea is based on the comparative analysis of the fuzzy interval methods performed in Chapter 2 and the coverage level and interval width results obtained. From this first stage, the simplicity of the model structure and the versatility associated with the fuzzy prediction interval method based on fuzzy numbers justify the decision to consider that method as a suitable choice for the later design of the novel evolving fuzzy prediction interval.

The second stage of this work was the design of novel evolving fuzzy prediction interval methods. At this point, some updating mechanisms previously developed for evolving intelligent systems were adapted to make them compatible with their inclusion in the structure of the fuzzy prediction interval model. Moreover, additional updating mechanisms had to be developed, such as the interval width updating procedure, which has not been developed in depth in the literature.

Regarding the first proposal, inspired by the main concepts of modeling in nonstationary environments, the proposed evolving fuzzy prediction interval (E-FPI) design considered two main changes a system could present: abrupt and gradual drifts. Therefore, the proposed algorithm was structured based on two separate paths. First, a passive approach where the existing elements of the model are constantly updated at each new instant. Second, an active approach where significant changes are introduced into the model if an abrupt drift is detected.

The proposed update mechanism included in the E-FPI design was initially tested in two case studies: modeling synthetic data from a generic nonlinear system with different types of change and forecasting solar power generation. Based on the tests performed with the synthetic data, the effectiveness of the mechanisms of creation and merging of new clusters was confirmed. On the other hand, for modeling the solar power generation data, the updating mechanism of the base model and the proposed novel updating of the spread parameters

of the interval showed their utility for considerably improving the performance of the interval in this applied case. Additionally, including the altered data in the simulation tests shows the robustness achieved by the proposed evolving interval to handle photovoltaic panel obstruction.

However, during the simulated tests of the E-FPI, a difficulty was identified regarding the high computational complexity of the online model identification process, which can compromise the possible implementation into real nonlinear time-variant dynamical systems. In response to this problem, the second proposal for the interval design, denoted as the self-evolving fuzzy prediction interval (SE-FPI), was formulated in this thesis. The advantage of this new proposal is that the identification process and updating mechanism are implemented recursively. Thus, the usual computational complexity of offline identification of the interval based on fuzzy numbers is avoided.

The results achieved by the SE-FPI in the simulated and experimental tests implemented in this thesis show that this second proposal can properly approximate the performance of the E-FPI despite having a simplified updating process. However, reducing the model complexity comes with a trade-off with the robustness of its precision for modeling systems with its uncertainty. This conclusion is verified by some reported results where the first proposal (E-FPI) worked fine, while the second algorithm design (SE-FPI) had more difficulties in reaching similar levels of performance in terms of the coverage level. Despite this situation, both proposed evolving intervals (E-FPI and SE-FPI) presented promising results for all the simulated and experimental tests carried out during the development of this thesis.

The last proposal of this thesis was the new interval-based fault detection algorithm, which relies on the behavior of the interval failure index to activate alarms. The experimental results reported in this thesis show the benefits of the proposal in achieving accurate precision and recall performance when slow and fast changes occur in the system. Additionally, the instant when the fault starts can be estimated from the information provided by the interval and its failure index. However, one disadvantage of this proposal is the forced delay introduced in the detection process because the algorithm has to wait until the measurement associated with the future prediction is already taken before deciding to activate the alarm.

5.1 Future work

Based on the results achieved by the methodologies proposed in this thesis, there are different paths to follow as future work.

For example, future work should consider the use of the evolving fuzzy prediction interval proposed here to improve the implementation of robust control strategies, such as robust model predictive control, on systems that tend to present changes in their behavior and its uncertainty.

On the other hand, future research directions should also consider new proposals to improve the performance of the interval-based fault detection algorithm designed in this work. In this topic, future research should consider trying to overcome the problem of delays presented in this thesis for alarm activations.

Finally, an alternative future work that can be considered is the extension of the methodologies presented in this thesis to higher-dimensional problems. With respect to this, the thesis proposals were applied on cases where the system had a single output. When trying to implement multiple intervals for different variables that can affect a predictive control strategy, the uncertainty modeling would not be optimal. Therefore, the proposals could be adapted to handle multiple output variables at the same time.

Bibliography

[1] Y. Xu, M. Zhang, Q. Zhu, and Y. He, "An improved multi-kernel rvm integrated with ceemd for high-quality intervals prediction construction and its intelligent modeling application," *Chemometrics and Intelligent Laboratory Systems*, vol. 171, pp. 151 – 160, 2017.

[2] L. G. Marín, N. Cruz, D. Sáez, M. Sumner, and A. Núñez, "Prediction interval methodology based on fuzzy numbers and its extension to fuzzy systems and neural networks," *Expert Systems with Applications*, vol. 119, pp. 128 – 141, 2019.

[3] H. M. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE Access*, vol. 6, pp. 36218–36234, 2018.

[4] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Transactions on neural networks*, vol. 22, no. 9, pp. 1341–1356, 2011.

[5] S. Jahandari, A. Kalhor, and B. N. Araabi, "A self tuning regulator design for nonlinear time varying systems based on evolving linear models," *Evolving Systems*, vol. 7, no. 3, pp. 159–172, 2016.

[6] O. Cartagena, S. Parra, D. Muñoz-Carpintero, L. G. Marín, and D. Sáez, "Review on fuzzy and neural prediction interval modelling for nonlinear dynamical systems," *IEEE Access*, vol. 9, pp. 23357–23384, 2021.

[7] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[8] R. J. Patton, F. J. Uppal, and C. J. Lopez-toribio, "Soft computing approaches to fault diagnosis for dynamic systems: A survey," *IFAC Proceedings Volumes*, vol. 33, no. 11, pp. 303–315, 2000. 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes 2000 (SAFEPROCESS 2000), Budapest, Hungary, 14-16 June 2000.

[9] J. M. Mendel, *Type-1 Fuzzy Systems: Design Methods and Applications*, pp. 161–244. Cham: Springer International Publishing, 2017.

[10] P. Melin and O. Castillo, "A review on the applications of type-2 fuzzy logic in classification and pattern recognition," *Expert Systems with Applications*, vol. 40, no. 13, pp. 5413–5423, 2013.

[11] O. Castillo and P. Melin, "A review on interval type-2 fuzzy logic applications in intelligent control," *Information Sciences*, vol. 279, pp. 615–631, 2014.

[12] J. M. Mendel, *Type-1 Fuzzy Systems*, pp. 101–159. Cham: Springer International Publishing, 2017.

[13] E. Kayacan and M. A. Khanesar, "Chapter 2 - fundamentals of type-1 fuzzy logic theory," in *Fuzzy Neural Networks for Real Time Control Applications*, pp. 13–24, Butterworth-Heinemann, 2016.

[14] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, pp. 116–132, Jan 1985.

[15] T. Heskes, "Practical confidence and prediction intervals," in *Advances in neural information processing systems*, pp. 176–182, 1997.

[16] I. Škrjanc, S. Blažič, and O. Agamennoni, "Identification of dynamical systems with a robust interval fuzzy model," *Automatica*, vol. 41, no. 2, pp. 327 – 332, 2005.

[17] S. Oblak, "Interval fuzzy modelling in fault detection for a class of processes with interval-type parameters," in *EUROCON 2005 - The International Conference on "Computer as a Tool"*, vol. 2, pp. 1867–1870, Nov 2005.

[18] S. Oblak, I. Skrjanc, and S. Blazic, "On applying interval fuzzy model to fault detection and isolation for nonlinear input-output systems with uncertain parameters," in *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*, pp. 465–470, Aug 2005.

[19] S. Oblak, I. Škrjanc, and S. Blažič, "Fault detection for nonlinear systems with uncertain parameters based on the interval fuzzy model," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 4, pp. 503 – 510, 2007.

[20] S. Oblak, I. Škrjanc, and S. Blažzič, "A new fault-detection system for nonlinear systems based on an interval fuzzy model," in *2007 European Control Conference (ECC)*, pp. 2956–2962, July 2007.

[21] Y.-W. Chen, J.-B. Yang, C.-C. Pan, D.-L. Xu, and Z.-J. Zhou, "Identification of uncertain nonlinear systems: Constructing belief rule-based models," *Knowledge-Based Systems*, vol. 73, pp. 124 – 133, 2015.

[22] D. Senthilkumar and C. Mahanta, "Identification of uncertain nonlinear systems for robust fuzzy control," *ISA Transactions*, vol. 49, no. 1, pp. 27 – 38, 2010.

[23] F. Valencia, J. Collado, D. Sáez, and L. G. Marín, "Robust energy management system

for a microgrid based on a fuzzy prediction interval model," *IEEE Transactions on Smart Grid*, vol. 7, pp. 1486–1494, May 2016.

[24] F. Valencia, D. Sáez, J. Collado, F. Ávila, A. Marquez, and J. J. Espinosa, "Robust energy management system based on interval fuzzy models," *IEEE Transactions on Control Systems Technology*, vol. 24, pp. 140–157, Jan 2016.

[25] G. M. T. Nguyen and K. Uchida, "An improved interval fuzzy modeling method: Applications to the estimation of photovoltaic/wind/battery power in renewable energy systems," *Energies*, vol. 11, p. 482, 03 2018.

[26] Z. Xu and C. Sun, "Interval t-s fuzzy model and its application to identification of nonlinear interval dynamic system based on interval data," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 4144–4149, Dec 2009.

[27] S. Zaidi and A. Kroll, "A novel approach to t-s fuzzy modeling of nonlinear dynamic systems with uncertainties using symbolic interval-valued outputs," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 1196 – 1201, 2015. 17th IFAC Symposium on System Identification SYSID 2015.

[28] S. Zaidi and A. Kroll, "Noe ts fuzzy modelling of nonlinear dynamic systems with uncertainties using symbolic interval-valued data," *Applied Soft Computing*, vol. 57, pp. 353 – 362, 2017.

[29] F. Veltman, L. G. Marin, D. Sáez, L. Guitierrez, and A. Núñez, "Prediction interval modeling tuned by an improved teaching learning algorithm applied to load forecasting in microgrids," in *2015 IEEE Symposium Series on Computational Intelligence*, pp. 651–658, Dec 2015.

[30] N. N. Karnik and J. M. Mendel, "Operations on type-2 fuzzy sets," *Fuzzy Sets and Systems*, vol. 122, no. 2, pp. 327 – 348, 2001.

[31] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy Systems*, vol. 8, pp. 535–550, Oct 2000.

[32] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—i," *Information Sciences*, vol. 8, no. 3, pp. 199 – 249, 1975.

[33] J. M. Mendel, *Interval Type-2 Fuzzy Systems*, pp. 449–527. Cham: Springer International Publishing, 2017.

[34] N. N. Karnik and J. M. Mendel, "Applications of type-2 fuzzy logic systems to forecasting of time-series," *Information Sciences*, vol. 120, no. 1, pp. 89 – 111, 1999.

[35] Qilian Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems," in *Ninth IEEE International Conference on Fuzzy Systems. FUZZ- IEEE 2000 (Cat. No.00CH37063)*, vol. 1, pp. 328–333 vol.1, 2000.

[36] K. Huarng and H.-K. Yu, "A type 2 fuzzy time series model for stock index forecasting," *Physica A: Statistical Mechanics and its Applications*, vol. 353, pp. 445 – 462, 2005.

[37] A. Khosravi, S. Nahavandi, and D. Creighton, "Short term load forecasting using interval type-2 fuzzy logic systems," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pp. 502–508, 2011.

[38] A. Khosravi, S. Nahavandi, D. Creighton, and D. Srinivasan, "Interval type-2 fuzzy logic systems for load forecasting: A comparative study," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1274–1282, 2012.

[39] M. S. Fadali, S. Jafarzadeh, and A. Nafeh, "Fuzzy tsk approximation using type-2 fuzzy logic systems and its application to modeling a photovoltaic array," in *Proceedings of the 2010 American Control Conference*, pp. 6454–6459, June 2010.

[40] S. Jafarzadeh, M. S. Fadali, and C. Y. Evrenosoglu, "Solar power prediction using interval type-2 tsk modeling," *IEEE Transactions on Sustainable Energy*, vol. 4, pp. 333–339, April 2013.

[41] A. Khosravi and S. Nahavandi, "Combined nonparametric prediction intervals for wind power generation," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 849–856, 2013.

[42] A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction intervals for short-term wind farm power generation forecasts," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 3, pp. 602–610, 2013.

[43] L. G. Marín, F. Valencia, and D. Sáez, "Prediction interval based on type-2 fuzzy systems for wind power generation and loads in microgrid control design," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 328–335, 2016.

[44] W. Zou, C. Li, and P. Chen, "An inter type-2 fcr algorithm based t–s fuzzy model for short-term wind power interval prediction," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 4934–4943, 2019.

[45] C. Juang and Y. Tsao, "A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1411–1424, 2008.

[46] C.-F. Juang and R.-B. Huang, "A mamdani-type recurrent interval type-2 fuzzy neural network for identification of dynamic systems with measurement noise," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8975 – 8980, 2011. 18th IFAC World Congress.

[47] Y. Lin, S. Liao, J. Chang, and C. Lin, "Simplified interval type-2 fuzzy neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 959–969, 2014.

[48] V. Sumati, P. Chellapilla, S. Paul, and L. Singh, "Parallel interval type-2 subsethood neural fuzzy inference system," *Expert Systems with Applications*, vol. 60, pp. 156 –

168, 2016.

[49] N. Baklouti, A. Abraham, and A. Alimi, "A beta basis function interval type-2 fuzzy neural network for time series applications," *Engineering Applications of Artificial Intelligence*, vol. 71, pp. 259 – 274, 2018.

[50] N. Anh, S. Suresh, M. Pratama, and N. Srikanth, "Interval prediction of wave energy characteristics using meta-cognitive interval type-2 fuzzy inference system," *Knowledge-Based Systems*, vol. 169, pp. 28 – 38, 2019.

[51] C.-M. Lin and T.-L. Le, "Pso-self-organizing interval type-2 fuzzy neural network for antilock braking systems," *International Journal of Fuzzy Systems*, vol. 19, no. 5, pp. 1362–1374, 2017.

[52] A. Kavousi-Fard, A. Khosravi, and S. Nahavandi, "A new fuzzy-based combined prediction interval for wind power forecasting," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 18–26, 2016.

[53] M. Han, K. Zhong, T. Qiu, and B. Han, "Interval type-2 fuzzy neural networks for chaotic time series prediction: A concise overview," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2720–2731, 2019.

[54] D. Muñoz-Carpintero, S. Parra, O. Cartagena, D. Sáez, L. G. Marín, and I. Škrjanc, "Fuzzy interval modelling based on joint supervision," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–8, July 2020.

[55] N. Cruz, L. G. Marín, and D. Sáez, "Neural network prediction interval based on joint supervision," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

[56] I. Škrjanc, "Fuzzy confidence interval for ph titration curve," *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 4083 – 4090, 2011.

[57] D. Sáez, F. Ávila, D. Olivares, C. Cañizares, and L. Marín, "Fuzzy prediction interval models for forecasting renewable resources and loads in microgrids," *IEEE Transactions on Smart Grid*, vol. 6, pp. 548–556, March 2015.

[58] R. Morales, D. Sáez, L. G. Marín, and A. Nuñez, "Microgrid planning based on fuzzy interval models of renewable resources," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 336–343, July 2016.

[59] S. F. Rafique, Z. Jianhua, R. Rafique, J. Guo, and I. Jamil, "Renewable generation (wind/solar) and load modeling through modified fuzzy prediction interval," *International Journal of Photoenergy*, vol. 2018, 2018.

[60] A. Bayas, I. Škrjanc, and D. Sáez, "Design of fuzzy robust control strategies for a distributed solar collector field," *Applied Soft Computing*, vol. 71, pp. 1009 – 1019, 2018.

[61] A. Núñez and B. De Schutter, "Distributed identification of fuzzy confidence intervals for traffic measurements," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 6995–7000, Dec 2012.

[62] O. Cartagena, D. Muñoz-Carpintero, and D. Sáez, "A robust predictive control strategy for building hvac systems based on interval fuzzy models," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–8, July 2018.

[63] M. Crispoltoni, M. Fravolini, F. Balzano, S. D'Urso, and M. Napolitano, "Interval fuzzy model for robust aircraft imu sensors fault detection," *Sensors*, vol. 18, no. 8, p. 2488, 2018.

[64] S. Tomažič, D. Dovžan, and I. Škrjanc, "Confidence-interval-fuzzy-model-based indoor localization," *IEEE Transactions on Industrial Electronics*, vol. 66, pp. 2015–2024, March 2019.

[65] P. Angelov, D. P. Filev, and N. Kasabov, *Evolving intelligent systems: methodology and applications*, vol. 12. John Wiley & Sons, 2010.

[66] I. Škrjanc, J. A. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey," *Information Sciences*, vol. 490, pp. 344–368, 2019.

[67] E. Lughofer, *Evolving Fuzzy and Neuro-Fuzzy Systems: Fundamentals, Stability, Explainability, Useability, and Applications*, ch. Chapter 4, pp. 133–234. World Scientific, 2022.

[68] M. Fazle Azeem, M. Hanmandlu, and N. Ahmad, "Structure identification of generalized adaptive neuro-fuzzy inference systems," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 5, pp. 666–681, 2003.

[69] J. Platt, "A resource-allocating network for function interpolation," *Neural computation*, vol. 3, no. 2, pp. 213–225, 1991.

[70] C.-F. Juang and C.-T. Lin, "An online self-constructing neural fuzzy inference network and its applications," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 1, pp. 12–32, 1998.

[71] N. K. Kasabov and Q. Song, "Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[72] E. D. Lughofer, "Flexfis: A robust incremental learning approach for evolving takagi–sugeno fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1393–1410, 2008.

[73] P. P. Angelov and D. P. Filev, "An approach to online identification of takagi-sugeno fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 484–498, 2004.

[74] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran, "Sequential adaptive fuzzy inference system (safis) for nonlinear system identification and prediction," *Fuzzy Sets and Systems*, vol. 157, no. 9, pp. 1260–1275, 2006. Fuzzy Concepts Applied to Food Control Quality Control.

[75] G. Leng, G. Prasad, and T. M. McGinnity, "An on-line algorithm for creating self-organizing fuzzy neural networks," *Neural Networks*, vol. 17, no. 10, pp. 1477–1493, 2004.

[76] N. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, pp. 902–918, Dec 2001.

[77] D. Leite, P. Costa, and F. Gomide, "Evolving granular neural networks from fuzzy data streams," *Neural Networks*, vol. 38, pp. 1–16, 2013.

[78] D. Leite, R. M. Palhares, V. C. S. Campos, and F. Gomide, "Evolving granular fuzzy model-based control of nonlinear dynamic systems," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 923–938, 2015.

[79] D. Leite, M. Santana, A. Borges, and F. Gomide, "Fuzzy granular neural network for incremental modeling of nonlinear chaotic systems," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 64–71, 2016.

[80] S. Wu and M. J. Er, "Dynamic fuzzy neural networks-a novel approach to function approximation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 2, pp. 358–364, 2000.

[81] S. Wu, M. J. Er, M. Ni, and W. E. Leithead, "A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks," in *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, vol. 4, pp. 2453–2457 vol.4, 2000.

[82] H. Soleimani-B, C. Lucas, and B. N. Araabi, "Recursive gath–geva clustering as a basis for evolving neuro-fuzzy modeling," *Evolving Systems*, vol. 1, no. 1, pp. 59–71, 2010.

[83] P. Angelov and D. Filev, "Simpl_ets: a simplified method for learning evolving takagi-sugeno fuzzy models," in *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pp. 1068–1073, 2005.

[84] P. Angelov, *Evolving Takagi-Sugeno Fuzzy Systems from Streaming Data (eTS+)*, ch. 2, pp. 21–50. John Wiley & Sons, Ltd, 2010.

[85] E. Lughofer, J.-L. Bouchot, and A. Shaker, "On-line elimination of local redundancies in evolving fuzzy systems," *Evolving systems*, vol. 2, no. 3, pp. 165–187, 2011.

[86] E. Lughofer, C. Cernuda, S. Kindermann, and M. Pratama, "Generalized smart evolving fuzzy systems," *Evolving systems*, vol. 6, no. 4, pp. 269–292, 2015.

[87] D. Dovžan, V. Logar, and I. Škrjanc, "Implementation of an evolving fuzzy model (efumo) in a monitoring system for a waste-water treatment process," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1761–1776, 2015.

[88] D. Dovžan, "Evolving fuzzy model in fault detection system," in *2017 Evolving and Adaptive Intelligent Systems (EAIS)*, pp. 1–8, IEEE, 2017.

[89] D. Leite, P. Costa, and F. Gomide, "Granular approach for evolving system modeling," in *Computational Intelligence for Knowledge-Based Systems Design*, (Berlin, Heidelberg), pp. 340–349, Springer Berlin Heidelberg, 2010.

[90] D. Leite, P. Costa, and F. Gomide, "Interval approach for evolving granular system modeling," in *Learning in Non-Stationary Environments: Methods and Applications*, (New York, NY), pp. 271–300, Springer New York, 2012.

[91] M. Pratama, J. Lu, and G. Zhang, "Evolving type-2 fuzzy classifier," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 3, pp. 574–589, 2016.

[92] C.-F. Juang, C.-F. Lu, and Y.-W. Tsao, "A self-evolving interval type-2 fuzzy neural network for nonlinear systems identification," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 7588–7593, 2008. 17th IFAC World Congress.

[93] C. Juang and Y. Tsao, "A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1411–1424, 2008.

[94] C. Juang, R. Huang, and Y. Lin, "A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1092–1105, 2009.

[95] Y. Lin, J. Chang, N. R. Pal, and C. Lin, "A mutually recurrent interval type-2 neural fuzzy system (mrit2nfs) with self-evolving structure and parameters," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 492–509, 2013.

[96] Y. Lin, J. Chang, and C. Lin, "A tsk-type-based self-evolving compensatory interval type-2 fuzzy neural network (tscit2fnn) and its applications," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 447–459, 2014.

[97] K. Subramanian, A. K. Das, S. Sundaram, and S. Ramasamy, "A meta-cognitive interval type-2 fuzzy inference system and its projection based learning algorithm," *Evolving Systems*, vol. 5, no. 4, pp. 219–230, 2014.

[98] S. F. Toloue and M. Akbarzadeh-T, "A hierarchical fuzzy approach for adaptation of pre-given parameters in an interval type-2 tsk fuzzy neural structure," in *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 425–430, 2014.

[99] M. Pratama, E. Lughofer, M. J. Er, W. Rahayu, and T. Dillon, "Evolving type-2 recurrent fuzzy neural network," in *2016 International Joint Conference on Neural*

Networks (IJCNN), pp. 1841–1848, 2016.

[100] M. Pratama, J. Lu, E. Lughofer, G. Zhang, and M. J. Er, "An incremental learning of concept drifts using evolving type-2 recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1175–1192, 2017.

[101] C.-M. Lin, T.-L. Le, and T.-T. Huynh, "Self-evolving function-link interval type-2 fuzzy neural network for nonlinear system identification and control," *Neurocomputing*, vol. 275, pp. 2239–2250, 2018.

[102] C. Luo, C. Tan, X. Wang, and Y. Zheng, "An evolving recurrent interval type-2 intuitionistic fuzzy neural network for online learning and time series prediction," *Applied Soft Computing*, vol. 78, pp. 150–163, 2019.

[103] H. Wang, C. Luo, and X. Wang, "Synchronization and identification of nonlinear systems by using a novel self-evolving interval type-2 fuzzy lstm-neural network," *Engineering Applications of Artificial Intelligence*, vol. 81, pp. 79–93, 2019.

[104] J. Zhao, Y. Liu, W. Pedrycz, and W. Wang, "Spatiotemporal prediction for energy system of steel industry by generalized tensor granularity based evolving type-2 fuzzy neural network," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 7933–7945, 2021.

[105] Y. Liu, J. Zhao, W. Wang, and W. Pedrycz, "Prediction intervals for granular data streams based on evolving type-2 fuzzy granular neural network dynamic ensemble," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 4, pp. 874–888, 2021.

[106] J. Tavoosi, A. A. Suratgar, M. B. Menhaj, A. Mosavi, A. Mohammadzadeh, and E. Ranjbar, "Modeling renewable energy systems by a self-evolving nonlinear consequent part recurrent type-2 fuzzy system for power prediction," *Sustainability*, vol. 13, no. 6, 2021.

[107] L. Maciel, F. Gomide, and R. Ballini, "Evolving possibilistic fuzzy modeling for financial interval time series forecasting," in *2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC)*, pp. 1–6, 2015.

[108] E. Lughofer and M. Pratama, "Online active learning in data stream regression using uncertainty sampling based on evolving generalized fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 292–309, 2018.

[109] I. Škrjanc, "An evolving concept in the identification of an interval fuzzy model of wiener-hammerstein nonlinear dynamic systems," *Information Sciences*, vol. 581, pp. 73–87, 2021.

[110] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, 2015.

[111] C. Alippi, *Learning in Nonstationary and Evolving Environments*, pp. 211–247. Cham:

Springer International Publishing, 2014.

[112] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132–156, 2017.

[113] D. J. D'Souza and K. R. Uday Kumar Reddy, "Anomaly detection for big data using efficient techniques: A review," in *Advances in Artificial Intelligence and Data Engineering* (N. N. Chiplunkar and T. Fukao, eds.), (Singapore), pp. 1067–1080, Springer Singapore, 2021.

[114] K. Shaukat, T. M. Alam, S. Luo, S. Shabbir, I. A. Hameed, J. Li, S. K. Abbas, and U. Javed, "A review of time-series anomaly detection techniques: A step to future perspectives," in *Advances in Information and Communication* (K. Arai, ed.), (Cham), pp. 865–877, Springer International Publishing, 2021.

[115] D. van Schrick, "Remarks on terminology in the field of supervision, fault detection and diagnosis," *IFAC Proceedings Volumes*, vol. 30, no. 18, pp. 959 – 964, 1997. IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS 97), Kingston upon Hull, UK, 26-28 August 1997.

[116] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1976.

[117] R. Isermann, "Process fault detection based on modeling and estimation methods: A survey," *Automatica*, vol. 20, no. 4, pp. 387 – 404, 1984.

[118] J. J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Control Systems Magazine*, vol. 8, no. 6, pp. 3–11, 1988.

[119] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459 – 474, 1990.

[120] E. Alcorta García and P. Frank, "Deterministic nonlinear observer-based approaches to fault diagnosis: A survey," *Control Engineering Practice*, vol. 5, no. 5, pp. 663 – 670, 1997.

[121] R. Isermann and P. Ballé, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control Engineering Practice*, vol. 5, no. 5, pp. 709 – 719, 1997.

[122] P. M. Frank and X. Ding, "Survey of robust residual generation and evaluation methods in observer-based fault detection systems," *Journal of process control*, vol. 7, no. 6, pp. 403–424, 1997.

[123] R. Isermann, "Model-based fault-detection and diagnosis â€“ status and applications," *Annual Reviews in Control*, vol. 29, no. 1, pp. 71 – 85, 2005.

[124] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 293 – 311, 2003.

[125] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part iii: Process history based methods," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 327 – 346, 2003.

[126] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, July 2009.

[127] C. Angeli and A. Chatzinikolaou, "On-line fault detection techniques for technical systems: A survey.," *IJCSA*, vol. 1, no. 1, pp. 12–30, 2004.

[128] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.

[129] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418–6428, 2014.

[130] H. Dong, Z. Wang, S. X. Ding, and H. Gao, "A survey on distributed filtering and fault detection for sensor networks," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[131] M. El Hachemi Benbouzid, "A review of induction motors signature analysis as a medium for faults detection," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 984–993, 2000.

[132] S. Nandi, H. A. Toliyat, and X. Li, "Condition monitoring and fault diagnosis of electrical motors—a review," *IEEE Transactions on Energy Conversion*, vol. 20, no. 4, pp. 719–729, 2005.

[133] S. Mortazavizadeh and S. Mousavi, "A review on condition monitoring and diagnostic techniques of rotating electrical machines," *Physical Science International Journal*, pp. 310–338, 2014.

[134] Z. Feng, M. Liang, and F. Chu, "Recent advances in time-frequency analysis methods for machinery fault diagnosis: A review with application examples," *Mechanical Systems and Signal Processing*, vol. 38, no. 1, pp. 165 – 205, 2013. Condition monitoring of machines in non-stationary operations.

[135] L. Qin, X. He, and D. Zhou, "A survey of fault diagnosis for swarm systems," *Systems Science & Control Engineering: An Open Access Journal*, vol. 2, no. 1, pp. 13–23, 2014.

[136] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part ii: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3768–3774, 2015.

[137] R. V. Beard, *Failure accomodation in linear systems through self-reorganization.* PhD thesis, Massachusetts Institute of Technology, 1971.

[138] A. Pertew, H. Marquez, and Q. Zhao, "Lmi-based sensor fault diagnosis for nonlinear lipschitz systems," *Automatica*, vol. 43, no. 8, pp. 1464 – 1469, 2007.

[139] D. Zhang, H. Wang, B. Lu, and Z. Wang, "Lmi-based fault detection fuzzy observer design with multiple performance constraints for a class of non-linear systems: comparative study," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 1, pp. 633–645, 2012.

[140] R. Li and J. H. Olson, "Fault detection and diagnosis in a closed-loop nonlinear distillation process: application of extended kalman filters," *Industrial & engineering chemistry research*, vol. 30, no. 5, pp. 898–908, 1991.

[141] H. Liu, D. Liu, C. Lu, and X. Wang, "Fault diagnosis of hydraulic servo system using the unscented kalman filter," *Asian Journal of Control*, vol. 16, no. 6, pp. 1713–1725, 2014.

[142] C. Hajiyev and H. E. Soken, "Robust adaptive kalman filter for estimation of uav dynamics in the presence of sensor/actuator faults," *Aerospace Science and Technology*, vol. 28, no. 1, pp. 376 – 383, 2013.

[143] G. H. B. Foo, X. Zhang, and D. M. Vilathgamuwa, "A sensor fault detection and isolation method in interior permanent-magnet synchronous motor drives based on an extended kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3485–3495, 2013.

[144] G. M. Joksimović, J. Riger, T. M. Wolbank, N. Perić, and M. Vašak, "Stator-current spectrum signature of healthy cage rotor induction machines," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4025–4033, 2013.

[145] X. Gong and W. Qiao, "Bearing fault diagnosis for direct-drive wind turbines via current-demodulated signals," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3419–3428, 2013.

[146] M. Bo, J. Zhi-nong, and W. Zhong-qing, "Development of the task-based expert system for machine fault diagnosis," *Journal of Physics: Conference Series*, vol. 364, p. 012043, may 2012.

[147] D. V. Kodavade and S. D. Apte, "A universal object oriented expert system frame work for fault diagnosis," *International Journal of Intelligence Science*, vol. 2, no. 3, pp. 63–70, 2012.

[148] X. Wang, U. Kruger, G. W. Irwin, G. McCullough, and N. McDowell, "Nonlinear pca with the local approach for diesel engine fault detection and diagnosis," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 122–129, 2008.

[149] L. M. Elshenawy and H. A. Awad, "Recursive fault detection and isolation approaches of

time-varying processes," *Industrial & Engineering Chemistry Research*, vol. 51, no. 29, pp. 9812–9824, 2012.

[150] Y. Zhang, C. Bingham, M. Gallimore, *et al.*, "Fault detection and diagnosis based on extensions of pca," *Advances in Military Technology*, vol. 8, no. 2, pp. 27–41, 2013.

[151] S. X. Ding, S. Yin, K. Peng, H. Hao, and B. Shen, "A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2239–2247, 2013.

[152] R. Vitale, O. E. de Noord, and A. Ferrer, "A kernel-based approach for fault diagnosis in batch processes," *Journal of Chemometrics*, vol. 28, no. 8, pp. S697–S707, 2014.

[153] Z. Xiaoqiang, X. Yongfei, and W. Tao, "Fault detection of batch process based on multi-way kernel t-pls," *Journal of Chemical and Pharmaceutical Research*, vol. 6, no. 7, pp. 338–346, 2014.

[154] Y. Zhang, N. Yang, and S. Li, "Fault isolation of nonlinear processes based on fault directions and features," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1567–1572, 2014.

[155] D. Tsai, S. Wu, and W. Chiu, "Defect detection in solar modules using ica basis images," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 122–131, 2013.

[156] Y. Guo, J. Na, B. Li, and R.-F. Fung, "Envelope extraction based dimension reduction for independent component analysis in fault diagnosis of rolling element bearing," *Journal of Sound and Vibration*, vol. 333, no. 13, pp. 2983 – 2994, 2014.

[157] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560 – 2574, 2007.

[158] Z. B. Sahri, U. T. Malaysia, *et al.*, "Support vector machine-based fault diagnosis of power transformer using k nearest-neighbor imputed dga dataset," *Journal of Computer and Communications*, vol. 2, no. 09, p. 22, 2014.

[159] F. Ye, Z. Zhang, K. Chakrabarty, and X. Gu, "Board-level functional fault diagnosis using multikernel support vector machines and incremental learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 2, pp. 279–290, 2014.

[160] Y. Shatnawi and M. Al-khassaweneh, "Fault diagnosis in internal combustion engines using extension neural network," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1434–1443, 2014.

[161] O. Elnokity, I. I. Mahmoud, M. K. Refai, and H. M. Farahat, "Ann based sensor faults detection, isolation, and reading estimates–sfdire: applied in a nuclear process," *Annals of Nuclear Energy*, vol. 49, pp. 131–142, 2012.

[162] S. Toma, L. Capocchi, and G. Capolino, "Wound-rotor induction generator inter-turn short-circuits diagnosis using a new digital neural network," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4043–4052, 2013.

[163] D. F. Leite, M. B. Hell, P. Costa Jr, and F. Gomide, "Real-time fault diagnosis of nonlinear systems," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 12, pp. e2665–e2673, 2009.

[164] M. Valtierra-Rodriguez, R. de Jesus Romero-Troncoso, R. A. Osornio-Rios, and A. Garcia-Perez, "Detection and classification of single and combined power quality disturbances using neural networks," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2473–2482, 2014.

[165] C. Nan, F. Khan, and M. T. Iqbal, "Real-time fault diagnosis using knowledge-based expert system," *Process Safety and Environmental Protection*, vol. 86, no. 1, pp. 55 – 71, 2008.

[166] F. Zidani, D. Diallo, M. E. H. Benbouzid, and R. Nait-Said, "A fuzzy-based approach for the diagnosis of fault modes in a voltage-fed pwm inverter induction motor drive," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 2, pp. 586–593, 2008.

[167] H. C. Cho, J. Knowles, M. S. Fadali, and K. S. Lee, "Fault detection and isolation of induction motors using recurrent neural networks and dynamic bayesian modeling," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 430–437, 2010.

[168] O. Özgönenel and T. Yalcin, "A complete motor protection algorithm based on pca and ann: a real time study," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 19, no. 3, pp. 317–334, 2011.

[169] D. He, R. Li, and J. Zhu, "Plastic bearing fault diagnosis based on a two-step data mining approach," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3429–3440, 2013.

[170] A. Soualhi, G. Clerc, and H. Razik, "Detection and diagnosis of faults in induction motor using an improved artificial ant clustering technique," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4053–4062, 2013.

[171] J. Seshadrinath, B. Singh, and B. K. Panigrahi, "Vibration analysis based interturn fault diagnosis in induction machines," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 340–350, 2014.

[172] B. M. Ebrahimi, M. Javan Roshtkhari, J. Faiz, and S. V. Khatami, "Advanced eccentricity fault recognition in permanent magnet synchronous motors using stator current signature analysis," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 2041–2052, 2014.

[173] N. Sheibat-Othman, N. Laouti, J.-P. Valour, and S. Othman, "Support vector machines combined to observers for fault diagnosis in chemical reactors," *The Canadian Journal of Chemical Engineering*, vol. 92, no. 4, pp. 685–695, 2014.

[174] M. Šimandl and I. Punčochář, "Active fault detection and control: Unified formulation and optimal design," *Automatica*, vol. 45, no. 9, pp. 2052 – 2059, 2009.

[175] A. Esna Ashari, R. Nikoukhah, and S. L. Campbell, "Active robust fault detection in closed-loop systems: Quadratic optimization approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2532–2544, 2012.

[176] X. Pu, T. H. Nguyen, D. Lee, K. Lee, and J. Kim, "Fault diagnosis of dc-link capacitors in three-phase ac/dc pwm converters by online estimation of equivalent series resistance," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4118–4127, 2013.

[177] Z. Chen, F. Lin, C. Wang, Y. Le Wang, and M. Xu, "Active diagnosability of discrete event systems and its application to battery fault diagnosis," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1892–1898, 2014.

[178] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Transactions on Neural Networks*, vol. 22, no. 3, pp. 337–346, 2011.

[179] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, no. 1, pp. 195 – 220, 2001.

[180] A. Alwosheel, S. van Cranenburgh, and C. G. Chorus, "Is your dataset big enough? sample size requirements when using artificial neural networks for discrete choice analysis," *Journal of Choice Modelling*, vol. 28, pp. 167–182, 2018.

[181] F. S. Makri and Z. M. Psillakis, "On success runs of a fixed length in bernoulli sequences: Exact and asymptotic results," *Computers & Mathematics with Applications*, vol. 61, no. 4, pp. 761–772, 2011.

[182] P. C. Mahalanobis, "On the generalized distance in statistics," in *Proceedings of the National Institute of Sciences of India*, pp. 49–55, 1936.

[183] B. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, vol. 4, no. 3, pp. 419–420, 1962.

[184] C. Paleologu, J. Benesty, and S. Ciochina, "A robust variable forgetting factor recursive least-squares algorithm for system identification," *IEEE Signal Processing Letters*, vol. 15, pp. 597–600, 2008.

[185] E. Schubert and M. Gertz, "Numerically stable parallel computation of (co-)variance," in *Proceedings of the 30th International Conference on Scientific and Statistical Database Management*, SSDBM '18, (New York, NY, USA), pp. 1–12, Association for Computing Machinery, 2018.

[186] J. K. Skipper, A. L. Guenther, and G. Nass, "The sacredness of. 05: A note concerning the uses of statistical levels of significance in social science," *The American Sociologist*, pp. 16–18, 1967.

[187] I. Škrjanc, "Cluster-volume-based merging approach for incrementally evolving fuzzy gaussian clustering-egauss+," *IEEE Transactions on Fuzzy Systems*, vol. 28, pp. 2222–2231, 9 2020.

[188] A. Qiu, A. W. Al-Dabbagh, and T. Chen, "A tradeoff approach for optimal event-triggered fault detection," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 2111–2121, 2019.

[189] S. Chen, S. A. Billings, and P. M. Grant, "Non-linear system identification using neural networks," *International Journal of Control*, vol. 51, no. 6, pp. 1191–1214, 1990.

[190] A. Nespoli, M. Mussetta, E. Ogliari, S. Leva, L. Fernández-Ramírez, and P. García-Triviño, "Robust 24 hours ahead forecast in a microgrid: A real case study," *Electronics*, vol. 8, no. 12, p. 1434, 2019.

[191] G. Andonovski, P. Angelov, S. Blažič, and I. Škrjanc, "A practical implementation of robust evolving cloud-based controller with normalized data space for heat-exchanger plant," *Applied Soft Computing*, vol. 48, pp. 29–38, 11 2016.

[192] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*. Springer Berlin Heidelberg, 2008.

[193] C. Alippi, M. Roveri, and F. Trovò, "A self-building and cluster-based cognitive fault diagnosis system for sensor networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1021–1032, 2014.

[194] M. Inacio, A. Lemos, and W. Caminhas, "Fault diagnosis with evolving fuzzy classifier based on clustering algorithm and drift detection," *Mathematical Problems in Engineering*, vol. 2015, 2015.

Annexes

Annex A

Updating formula for the interval width parameters

Taking into account that (3.12) depends on both $\underline{\boldsymbol{s}}_{\boldsymbol{i}}$ and $\overline{\boldsymbol{s}}_{\boldsymbol{i}}$ simultaneously, for obtaining a mathematical expression for the updating of each vector parameter, a separation of the interval width is proposed. Thus, the original interval width PINAW obtained by using (3.8) is divided into two different components:

$$\text{PINAW}^{up} = \frac{1}{NR}\sum_{k=1}^{N}\left(\overline{y}(k) - y(k)\right), \tag{A.1}$$

$$\text{PINAW}^{lw} = \frac{1}{NR}\sum_{k=1}^{N}\left(y(k) - \underline{y}(k)\right). \tag{A.2}$$

In equations (A.1)-(A.2) the values $\overline{y}(k), \underline{y}(k)$ are the bounds of the interval and $y(k)$ is the measurement of the output of the system at instant t. Because the update of the spread parameters has to be performed only for the closest cluster to the sample vector $\boldsymbol{x}(k)$, the equations (A.1)-(A.2) are raised differently for each cluster j. This is done by using the following values for the interval bounds and the system measurement, expressed in a vector form

$$\overline{y}_j(k) = \left(\boldsymbol{g^j}\right)^T \boldsymbol{x}(k) + \left(\overline{\boldsymbol{s}}^{\boldsymbol{j}}\right)^T |\boldsymbol{x}(k)|, \tag{A.3}$$

$$\underline{y}_j(k) = \left(\boldsymbol{g^j}\right)^T \boldsymbol{x}(k) - \left(\underline{\boldsymbol{s}}^{\boldsymbol{j}}\right)^T |\boldsymbol{x}(k)|, \tag{A.4}$$

$$y(k) = \left(\boldsymbol{g^j}\right)^T \boldsymbol{x}(k) + e(k), \tag{A.5}$$

where $\left(\boldsymbol{g^j}\right)^T$ is the vector of the lineal parameters associated with cluster j, $\left(\overline{\boldsymbol{s}}^{\boldsymbol{j}}\right)^T, \left(\underline{\boldsymbol{s}}^{\boldsymbol{j}}\right)^T$ are the vector of the corresponding spread parameters, and $e(k)$ is the model error that is initially

unknown at instant t. At this point, the two components of the interval width (A.1)-(A.2) can be rewritten for each cluster j, by using equations (A.3)-(A.5). Thus, both components can be expressed as follows

$$\text{PINAW}_j^{up} = \frac{1}{NR}\sum_{k=1}^{N}\left[\left(\overline{\boldsymbol{s}}^{\boldsymbol{j}}\right)^T|\boldsymbol{x}(k)| - e(k)\right], \tag{A.6}$$

$$\text{PINAW}_j^{lw} = \frac{1}{NR}\sum_{k=1}^{N}\left[e(k) + \left(\underline{\boldsymbol{s}}^{\boldsymbol{j}}\right)^T|\boldsymbol{x}(k)|\right]. \tag{A.7}$$

Based on these definitions, here the conditions to be considered for updating the values $\overline{\boldsymbol{s}}^{\boldsymbol{j}}$ and $\underline{\boldsymbol{s}}^{\boldsymbol{j}}$ must be explained. First, based on the values of PINAW_j^{up} and PINAW_j^{lw}, the following update guidelines must be considered:

- If PINAW_j^{up} or $\text{PINAW}_j^{lw} < 0$, that means the interval failed to contain the measurement $y(k)$. In that case, the spread parameter values associated with the interval width component with a negative value must be increased.

- If PINAW_j^{up} and $\text{PINAW}_j^{lw} \geq 0$, that means the interval contained the value of $y(k)$. In that case, the algorithm could consider reducing the values of the spread parameters to minimize the interval width.

On the other hand, another update guideline can be considered based on the coverage level obtained by the interval. Those additional guidelines are the following:

- If $\text{PICP} - \hat{\alpha} > 0$, that means the coverage level is higher than the desired value. In this case, the values of $\overline{\boldsymbol{s}}^{\boldsymbol{j}}$ and $\underline{\boldsymbol{s}}^{\boldsymbol{j}}$ could be reduced. This alternative is optional and may vary according to the behavior of the system. For applying this reduction of parameters, it is necessary to verify that this procedure does not significantly deteriorate the coverage level.

- If $\text{PICP} - \hat{\alpha} < 0$, that indicates the coverage level is lower than the desired value. In that case, the algorithm must consider an increment of the spread parameter values to reach the desired values of coverage level.

Taking as basis the optimization problem (3.12) used in the original interval identification, the following optimizations can be considered for the identification of each vector $\overline{\boldsymbol{s}}^{\boldsymbol{j}}$ and $\underline{\boldsymbol{s}}^{\boldsymbol{j}}$

$$\begin{aligned} \min_{\overline{\boldsymbol{s}}^{\boldsymbol{j}}} \quad & J_{up} = (\text{PINAW}_j^{up})^2 \\ s.t. \quad & \text{PICP} = \hat{\alpha}, \end{aligned} \tag{A.8}$$

$$\begin{aligned} \min_{\underline{\boldsymbol{s}}^{\boldsymbol{j}}} \quad & J_{lw} = (\text{PINAW}_j^{lw})^2 \\ s.t. \quad & \text{PICP} = \hat{\alpha}, \end{aligned} \tag{A.9}$$

where the objective functions are now $J_{up} = (\text{PINAW}_j^{up})^2$ and $J_{lw} = (\text{PINAW}_j^{lw})^2$, in order to guarantees the convexity of these optimization problems.

In this point of the algorithm, the gradients for the objective functions in (A.8)-(A.9) are considered as a preliminary value for the direction of the updating procedure of $\overline{\boldsymbol{s}}^{\boldsymbol{j}}$ and $\underline{\boldsymbol{s}}^{\boldsymbol{j}}$.

This is done in that way because (A.8)-(A.9) correspond to minimization problems. The respective gradients are given as follows

$$\nabla_{\overline{\boldsymbol{s}}^{\boldsymbol{j}}} J_{up} = \left(\frac{1}{NR} \sum_{k=1}^{N} \left(\overline{y}(k) - y(k) \right) \right) \cdot \left(\frac{1}{NR} \sum_{k=1}^{N} |\boldsymbol{x}(k)| \right), \tag{A.10}$$

$$\nabla_{\underline{\boldsymbol{s}}^{\boldsymbol{j}}} J_{lw} = - \left(\frac{1}{NR} \sum_{k=1}^{N} \left(y(k) - \underline{y}(k) \right) \right) \cdot \left(\frac{1}{NR} \sum_{k=1}^{N} |\boldsymbol{x}(k)| \right). \tag{A.11}$$

The computation of (A.10)-(A.11) may result with a high complexity for being solved for each instant t. Due to that, in this work the following approximation for the updating direction is considered

$$\Delta \overline{\boldsymbol{s}}^{\boldsymbol{j}} \approx -\nabla_{\overline{\boldsymbol{s}}^{\boldsymbol{j}}} J_{up} = - \left(\overline{y}(k) - y(k) \right) |\boldsymbol{x}(k)|, \tag{A.12}$$

$$\Delta \underline{\boldsymbol{s}}^{\boldsymbol{j}} \approx -\nabla_{\underline{\boldsymbol{s}}^{\boldsymbol{j}}} J_{lw} = \left(y(k) - \underline{y}(k) \right) |\boldsymbol{x}(k)|, \tag{A.13}$$

where both expressions only depend on the values measured at instant t. In (A.12)-(A.13) the orientation of the parameter updating is mainly given by the term $|\boldsymbol{x}(k)|$. The problem here lies in the fact that the sense of the parameter updating will only coincide with the requirement of minimizing the interval width.

In order to improve this updating process, an additional parameter η and the values of $\underline{\boldsymbol{s}}^{\boldsymbol{j}}, \overline{\boldsymbol{s}}^{\boldsymbol{j}}$ can be included as multiplicative factors to avoid the risk of the interval width from being overfitted to high-frequency noises. With the same purpose, it is proposed to replace the terms $(\overline{y}(k) - y(k))$ and $\left(y(k) - \underline{y}(k)\right)$ by the term $(\text{PIPC} - \hat{\alpha})$, because with this change a similar behavior of the updating process is achieved, but with a reduction of the effect of high-frequency noises, while the requirement of reaching a coverage level close to its desired value $\hat{\alpha}$ is also included. Thus, the updating formulas for the spread parameters $\underline{S}^j, \overline{S}^j$ are given as

$$\overline{\boldsymbol{s}}^{\boldsymbol{j}}(k+1) = \overline{\boldsymbol{s}}^{\boldsymbol{j}}(k) + \eta_{up} \left(\text{PIPC} - \hat{\alpha} \right) |\boldsymbol{x}(k)| \cdot \overline{\boldsymbol{s}}^{\boldsymbol{j}}(k), \tag{A.14}$$

$$\underline{\boldsymbol{s}}^{\boldsymbol{j}}(k+1) = \underline{\boldsymbol{s}}^{\boldsymbol{j}}(k) + \eta_{lw} \left(\text{PIPC} - \hat{\alpha} \right) |\boldsymbol{x}(k)| \cdot \underline{\boldsymbol{s}}^{\boldsymbol{j}}(k), \tag{A.15}$$

where η_{up} and η_{lw} are new tuning parameters that can be obtained independently of each other.

Annex B

Robustness test of the self-evolving fuzzy prediction interval

An additional test is carried out for evaluating the robustness of the proposed SE-FPI algorithm in the hypothetical case when the plant is affected by an unexpected disturbance during the learning phase. This situation was emulated by multiplying the output measurements by 0.9 during a period of 2,000 seconds (between samples 4,000 and 4,500 in the original dataset used in the learning phase). The results of the final intervals evaluated in the same validation dataset used in the previous experiment are shown in Figure B.1.

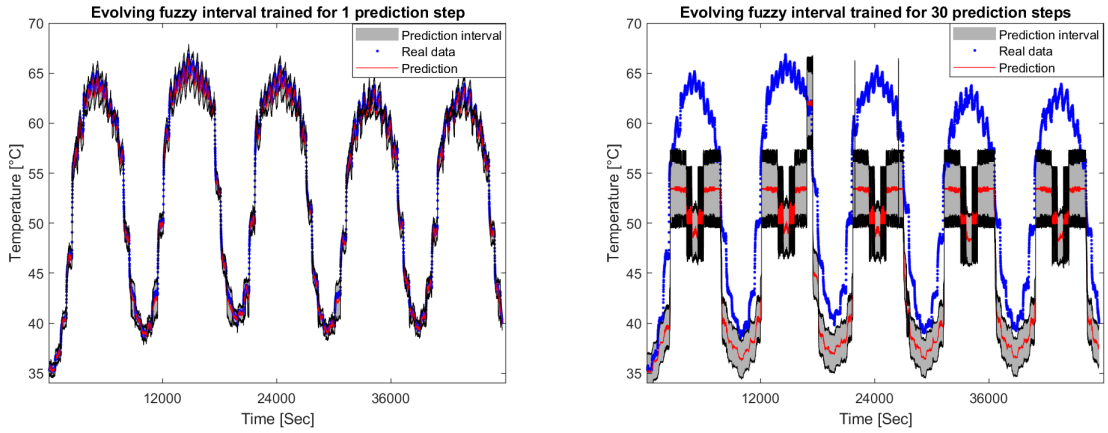


Figure B.1: The prediction interval output measured during the validation test.

From Figure B.1, it can be seen that, apparently, the evolving prediction interval can successfully handle the 1-step prediction despite the presence of an unexpected uncertainty during the learning phase. However, the 30-step prediction result shows that the quality of the model is considerably lower because it presents an important drop in performance in terms of the prediction error and the achieved coverage level. This result shows that the proposed algorithm is not robust enough to handle with unexpected disturbances in the learning phase if the user wants to make further predictions of more than one step ahead. Thus, here it is important to remark that the successfulness in the applicability of the proposed algorithm lies in the assumption that the data used in the learning phase does not present any kind of fault or substantial abnormal behavior of the system.