# Development of Self-Learning Type-2 Fuzzy Systems for System Identification and Control of Autonomous Systems

Ayad Al-Mahturi

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Engineering and Information Technology

Faculty of Engineering

The University of New South Wales

December 2021

THE UNIVERSITY OF NEW SOUTH WALES
**Thesis/Dissertation Sheet**

Surname or Family name:

First name: Other name/s:

Abbreviation for degree as given in the University calendar: **PhD**

School: School of Engineering and Information Technology **School of Engineering and Information Technology** Faculty: University of New South Wales, Canberra **Faculty of Engineering**

Title: Development of Self-Learning Type-2 Fuzzy Systems for System Identification and Control of Autonomous Systems

**Abstract**

Modelling and control of dynamic systems are faced by multiple technical challenges, mainly due to the nature of uncertain complex, nonlinear, and time-varying systems. Traditional modelling techniques require a complete understanding of system dynamics and obtaining comprehensive mathematical models is not always achievable due to limited knowledge of the systems. As universal approximators, fuzzy logic systems (FLSs), neural networks (NNs) and neuro-fuzzy systems have proved to be successful computational tools for representing the behaviour of complex dynamical systems. Moreover, FLSs, NNs and learning-based techniques have been gaining popularity for controlling complex, ill-defined, nonlinear systems in the face of uncertainties. However, fuzzy rules derived by experts can be too ad-hoc, and the performance is less than optimum. Aiming to accommodate uncertainties, this thesis aims to address the shortcomings of existing fuzzy techniques by utilisation of type-2 FLCs with novel adaptive capabilities. The first contribution of this thesis is a novel online system identification technique by means of a recursive interval type-2 Takagi-Sugeno fuzzy C-means clustering technique (IT2-TS-FC) to accommodate the footprint-of-uncertainties (FoUs). This development is meant to specifically address the shortcomings of type-1 fuzzy systems in capturing the footprint-of-uncertainties such as mechanical wear, rotor damage, battery drain and sensor and actuator faults. The algorithm has the capability to model uncertainties. The second contribution of this thesis is the development of a novel self-adaptive interval type-2 fuzzy controller named the SAF2C for controlling multi-input multi-output (MIMO) nonlinear systems. The adaptation law is derived using sliding mode control (SMC) theory to reduce the computation time so that the learning process can be expedited by 80% compared to separate single-input single-output (SISO) controllers. Another contribution of this thesis is a novel stand-alone enhanced self-adaptive interval type-2 fuzzy controller named the ESAF2C algorithm. The proposed technique is applied on a quadcopter UAV (QUAV), where extensive simulations and real-time flight tests for a hovering QUAV under wind disturbances are also conducted to validate the efficacy of the ESAF2C. Yet another contribution of this thesis is the development of a type-2 evolving fuzzy control system (T2-EFCS) to facilitate self-learning (either from scratch or from a certain predefined rule). The proposed technique is applied to control an unmanned ground vehicle in the presence of multiple external disturbances demonstrating the robustness of the proposed control systems.

**Declaration relating to disposition of project thesis/dissertation**

Signature Witness Date

**FOR OFFICE USE ONLY** Date of completion of requirements for Award

## Originality Statement

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

**Ayad Al-Mahturi**
15 December, 2021

## Copyright Statement

**Ayad Al-Mahturi**
15 December, 2021

## Authenticity Statement

**Ayad Al-Mahturi**
15 December, 2021

UNSW SYDNEY | Australia's Global University

# INCLUSION OF PUBLICATIONS STATEMENT

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

**Publications can be used in their thesis in lieu of a Chapter if:**

- The candidate contributed greater than 50% of the content in the publication and is the "primary author", ie. the candidate was responsible primarily for the planning, execution and preparation of the work for publication
- The candidate has approval to include the publication in their thesis in lieu of a Chapter from their supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis

Please indicate whether this thesis contains published material or not:

□ This thesis contains no publications, either published or submitted for publication *(if this box is checked, you may delete all the material on page 2)*

☒ Some of the work described in this thesis has been published and it has been documented in the relevant Chapters with acknowledgement *(if this box is checked, you may delete all the material on page 2)*

□ This thesis has publications (either published or submitted for publication) incorporated into it in lieu of a chapter and the details are presented below

**CANDIDATE'S DECLARATION**

I declare that:

- I have complied with the UNSW Thesis Examination Procedure
- where I have used a publication in lieu of a Chapter, the listed publication(s) below meet(s) the requirements to be included in the thesis.

| Candidate's Name | Signature | Date (dd/mm/yy) |
|---|---|---|
| Ayad Al-Mahturi | | 15/12/2021 |

# Abstract

Modelling and control of dynamic systems are faced by multiple technical challenges, mainly due to the nature of uncertain complex, nonlinear, and time-varying systems. Traditional modelling techniques require a complete understanding of system dynamics and obtaining comprehensive mathematical models is not always achievable due to limited knowledge of the systems as well as the presence of multiple uncertainties in the environment. As universal approximators, fuzzy logic systems (FLSs), neural networks (NNs) and neuro-fuzzy systems have proved to be successful computational tools for representing the behaviour of complex dynamical systems. Moreover, FLSs, NNs and learning-based techniques have been gaining popularity for controlling complex, ill-defined, nonlinear, and time-varying systems in the face of uncertainties. However, fuzzy rules derived by experts can be too ad-hoc, and the performance is less than optimum. In other words, generating fuzzy rules and membership functions in fuzzy systems is a potential challenge especially for systems with many variables. Moreover, under the umbrella of FLSs, although type-1 fuzzy logic control systems (T1-FLCs) have been applied to control various complex nonlinear systems, they have limited capability to handle uncertainties. Aiming to accommodate uncertainties, type-2 fuzzy logic control systems (T2-FLCs) were established. This thesis aims to address the shortcomings of existing fuzzy techniques by utilisation of type-2 FLCs with novel adaptive capabilities.

The first contribution of this thesis is a novel online system identification technique by means of a recursive interval type-2 Takagi-Sugeno fuzzy C-means clustering technique (IT2-TS-FC) to accommodate the footprint-of-uncertainties (FoUs). This development is meant to specifically address the shortcomings of type-1 fuzzy systems in capturing the footprint-of-uncertainties such as mechanical wear, rotor damage, battery drain and sensor and actuator faults. Unlike previous type-2 TS fuzzy models, the proposed method constructs two fuzzifiers (upper and lower) and two regression coefficients in the consequent part to handle uncertainties. The weighted least square method is employed to compute the regression coefficients. The proposed method is validated using two benchmarks, namely, real flight test data of a quadcopter drone and Mackey-Glass time series data. The algorithm has the capability to model uncertainties (e.g., noisy dataset).

The second contribution of this thesis is the development of a novel self-adaptive interval type-2 fuzzy controller named the SAF2C for controlling multi-input multi-output (MIMO) nonlinear systems. The adaptation law is derived using sliding mode control

(SMC) theory to reduce the computation time so that the learning process can be expedited by 80% compared to separate single-input single-output (SISO) controllers. The system employs the ‘Enhanced Iterative Algorithm with Stop Condition’ (EIASC) type-reduction method, which is more computationally efficient than the ‘Karnik-Mendel’ type-reduction algorithm. The stability of the SAF2C is proven using the Lyapunov technique. To ensure the applicability of the proposed control scheme, SAF2C is implemented to control several dynamical systems, including a simulated MIMO hexacopter unmanned aerial vehicle (UAV) in the face of external disturbance and parameter variations. The ability of SAF2C to filter the measurement noise is demonstrated, where significant improvement is obtained using the proposed controller in the face of measurement noise. Also, the proposed closed-loop control system is applied to control other benchmark dynamic systems (e.g., a simulated autonomous underwater vehicle and inverted pendulum on a cart system) demonstrating high accuracy and robustness to variations in system parameters and external disturbance.

Another contribution of this thesis is a novel stand-alone enhanced self-adaptive interval type-2 fuzzy controller named the ESAF2C algorithm, whose type-2 fuzzy parameters are tuned online using the SMC theory. This way, we expect to design a computationally efficient adaptive Type-2 fuzzy system, suitable for real-time applications by introducing the EIASC type-reducer. The proposed technique is applied on a quadcopter UAV (QUAV), where extensive simulations and real-time flight tests for a hovering QUAV under wind disturbances are also conducted to validate the efficacy of the ESAF2C. Specifically, the control performance is investigated in the face of external wind gust disturbances, generated using an industrial fan. Stability analysis of the ESAF2C control system is investigated using the Lyapunov theory.

Yet another contribution of this thesis is the development of a type-2 evolving fuzzy control system (T2-EFCS) to facilitate self-learning (either from scratch or from a certain predefined rule). T2-EFCS has two phases, namely, the structure learning and the parameters learning. The structure of T2-EFCS does not require previous information about the fuzzy structure, and it can start the construction of its rules from scratch with only one rule. The rules are then added and pruned in an online fashion to achieve the desired set-point. The proposed technique is applied to control an unmanned ground vehicle (UGV) in the presence of multiple external disturbances demonstrating the robustness of the proposed control systems. The proposed approach turns out to be computationally efficient as the system employs fewer fuzzy parameters while maintaining superior control performance.

# Acknowledgement

Undertaking the PhD journey has been an absolutely life-changing experience for me and it would not have been possible without the guidance and encouragement of a number of people.

First and foremost, I would like to express my sincere gratitude to my advisor Professor Matthew A. Garratt for the perpetual support during my PhD journey, for his motivation, patience, spirit, tremendous knowledge, and for being an extraordinary mentor. My sincere thanks also go to my joint advisor Dr. Fendy Santoso for his continued support, encouragement, guidance, and for being an extraordinary advisor. Their continuous motivation and valuable input helped me to reach this stage. I am also grateful to my co-supervisor Dr. Sreenatha G. Anavatti for his untiring guidance, advice, valuable suggestions, and constant support. Without their guidance and constant feedback, this PhD would not have been achievable.

I gratefully acknowledge the funding received towards my PhD program from the Research Training Program (RTP) awarded by the University of New South Wales, Canberra.

My sincere thanks go to the academic and administrative staff at UNSW Canberra for assisting me in various ways. My deep appreciation goes out to my supportive colleagues and friends at UNSW Canberra. Thanks to Ahmad Jobran Al-Mahasneh, Md Meftahul Ferdaus, Phi vu, Huanneng Qiu, Sobers Francis, Deepak Rajamohan, Praveen Kumar, Tanmoy Dam, and Mohamed Tarek. I also take this opportunity to thank the Yemeni community living in Canberra for their support during my studies at UNSW Canberra.

Above all, I would like to express my heartfelt gratitude to my Mum, Dad, bothers, and sisters for their blessings, encouragement, and sacrifice.

# Publications

## List of Publications

### Book Chapter

1. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, and S. G. Anavatti. Chapter 2 - modeling and control of a quadrotor unmanned aerial vehicle using type-2 fuzzy systems. In Anis Koubaa and Ahmad Taher Azar, editors, Unmanned Aerial Systems, Advances in Nonlinear Dynamics and Chaos (ANDC), pages 25-46. Academic Press, *Elsevier*, 2021.

## Journal Papers

1. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, and S. G. Anavatti. A robust self-adaptive interval type-2 TS fuzzy logic for controlling multi-input-multi-output nonlinear uncertain dynamical systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.
2. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, and S. G. Anavatti. Self-learning in aerial robotics using type-2 fuzzy systems: Case study in hovering quadrotor fight control. *IEEE Access*, 2021.
3. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, and S. G. Anavatti. A Novel Learning-from-Scratch Evolving Type-2 Fuzzy System for Robotic Control. Preparing draft to be submitted to *IEEE Transactions on Intelligent Vehicles*, 2021.

## Conference Papers

1. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, and S. G. Anavatti. Online system identification for nonlinear uncertain dynamical systems using recursive interval type-2 TS fuzzy C-means clustering. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pages 1695-1701, 2020.

2. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, S. G. Anavatti, and M. M. Ferdaus. A simplified model-free self-evolving TS fuzzy controller for nonlinear systems with uncertainties. In 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), pages 1-6, May 2020.

3. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, and S. G. Anavatti, and M. M. Ferdaus. Online Takagi-Sugeno fuzzy identification of a quadcopter using experimental input-output data. In 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pages 527-533, Dec 2019c.

4. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, and S. G. Anavatti. An intelligent control of an inverted pendulum based on an adaptive interval type-2 fuzzy inference system. In 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pages 1-6, 2019b.

5. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, and S. G. Anavatti. A robust adaptive interval type-2 fuzzy control for autonomous underwater vehicles. In 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), pages 19-24. IEEE, 2019a.

6. **A. Al-Mahturi**, F. Santoso, M. A. Garratt, and S. G. Anavatti. Nonlinear altitude control of a quadcopter drone using interval type-2 fuzzy logic. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pages 236-241. IEEE, 2018.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| AUVs | Autonomous Underwater Vehicles |
| AGVs | Autonomous Ground Vehicles |
| CMAC | Cerebellar Model Articulation Controller |
| CoTS | Commercial-off-the-Shelf |
| DOF | Degrees of Freedom |
| EA | Evolutionary Algorithm |
| EFSs | Evolving Fuzzy Systems |
| EIASC | Enhance Iterative Algorithm with Stop Condition |
| EIS | Evolving Intelligent System |
| EKF | Extended Kalman Filter |
| ESAF2C | Enhanced Self-Adaptive Interval Type-2 Fuzzy Controller |
| ESCs | Electronics Speed Controllers |
| eTS | evolving Takagi Sugeno |
| FBL | Feedback Linearization |
| FCM | Fuzzy C-Means |
| FCRM | Fuzzy C-Regression Model |
| FIS | Fuzzy Inference System |
| FLCs | Fuzzy Logic Control System |
| FLSs | Fuzzy Logic Systems |
| FNN | Fuzzy Neural Network |
| FoU | Footprint of Uncertainty |
| FQL | Fuzzy Q-learning |
| GA | Genetic Algorithm |
| GCS | Ground Control Station |
| GRNN | Generalized Regression Neural Network |
| IMUs | Inertial Measurement Units |
| IP | Inverted Pendulum |
| IT2FLCs | Interval Type-2 Fuzzy Logic Control System |
| IT2-FLSs | Interval Type-2 Fuzzy Logic Systems |
| IT2-TS-FC | Interval Type-2 Takagi-Sugeno Fuzzy C-means Clustering |
| KM | Karnik Mendel |
| LQR | Linear Quadratic Regulator |
| MAVLink | Micro Air Vehicle Link |
| MF | Membership Function |

| | |
|---|---|
| MIMO | Multi-Input-Multi-Output |
| MISO | Multi-Input-Single-Output |
| MPC | Model Predictive Controller |
| MSE | Mean Squared Error |
| NARMAX | Nonlinear Autoregressive Moving Average Model with Exogenous |
| NDEI | Non-Dimensional Error Index |
| NFSs | Neuro-Fuzzy Systems |
| NNs | Neural Networks |
| NT | Nie-Tan Type-Reduction |
| OBC | OnBoard Computer |
| PID | Proportional Integral Derivative |
| RBFNN | Radial Basis Function Neural Network |
| RLS | Recursive Least Square |
| RMSE | Root Mean Squared Error |
| ROS | Robot Operating System |
| RT | Rise Time |
| SAF2C | Self-Adaptive Interval Type-2 Fuzzy Controller |
| SE-TS-FC | Self-Evolving Takagi-Sugeno Fuzzy Controller |
| SISO | Single-Input-Single-Output |
| SMC | Sliding Mode Control |
| ST | Settling Time |
| T1-EFCS | Type-1 Evolving Fuzzy Control System |
| T1-EFSs | Type-1 Evolving Fuzzy Systems |
| T1FLCs | Type-1 Fuzzy Logic Control System |
| T1-FLSs | Type-1 Fuzzy Logic Systems |
| T1-FSs | Type-1 Fuzzy Sets |
| T2-EFCS | Type-2 Evolving Fuzzy Control System |
| T2-EFSs | Type-2 Evolving Fuzzy Systems |
| T2FLCs | Type-2 Fuzzy Logic Control System |
| T2-FLSs | Type-2 Fuzzy Logic Systems |
| T2-FSs | Type-2 Fuzzy Sets |
| TR | Type Reduction |
| TS | Takagi-Sugeno |
| TSK | Takasi-Sugeno-Kang |
| UAS | Unmanned Autonomous Systems |
| UASs | Unmanned Aerial Systems |
| UAVs | Unmanned Aerial Vehicles |
| UGVs | unmanned ground vehicles |

# Nomenclature

| | |
|---|---|
| t | Time |
| $u,\ v,\ w$ | Velocity components along the $(x, y, z)$-axes. $(u,\ v,\ w) \in B$ |
| $p,\ q,\ r$ | Angular rates along the $(x, y, z)$-axes. $(p,\ q,\ r) \in B$ |
| $X,\ Y,\ Z$ | Location of drone donated in the inertial frame |
| $\overline{X},\ \overline{Y},\ \overline{Z}$ | Global velocities |
| $\theta$ | Euler Pitch angle |
| $\phi$ | Euler Roll angle |
| $\psi$ | Euler Yaw angle |
| $\psi$ | Dead-zone parameter for SAF2C |
| $S_s(t)$ | Sliding surface - SAF2C |
| $\gamma$ | Learning rate - SAF2C |
| $\Lambda$ | Adaptive parameter - SAF2C |
| $\psi$ | Boundary layer thickness - SAF2C |
| $\theta$ | Pendulum angle - IP control |
| $x$ | Cart position - IP control |
| $F$ | Force on the cart - IP control |
| $m_{pen}$ | Pendulum mass - IP control |
| $\eta$ | Learning rate - IP control |
| $\gamma$ | Sliding surface slope - IP control |
| $X_E,\ Y_E,\ Z_E$ | Location of AUV donated in the inertial frame |
| $X_B,\ Y_B,\ Z_B$ | Location of AUV donated in the body frame |
| $\eta$ | Position and orientation vector of the AUV in the fixed frame |
| $\nu$ | Linear and angular velocity vector |
| $M(\nu)$ | Inertia matrix |
| $C(\dot{\nu})$ | Coriolis and Centripetal forces matrix |
| $D(\dot{\nu})$ | Hydrodynamic damping matrix |
| $G(\eta)$ | Vehicle's buoyancy and gravitational forces matrix |
| $\tau_\nu$ | Control efforts- AUV control |
| $s_{AUV}$ | Sliding surface- AUV control |
| $\gamma$ | Sliding surface positive term- AUV control |
| $J(\eta)$ | Euler angle mapping matrix |
| $V_\infty$ | Free-stream velocity |
| $V_i$ | Rotor induced velocity |
| $V_n$ | Normal free-stream velocity |

| | |
|---|---|
| $V_t$ | Tangential free-stream velocity |
| $\theta_0$ | Blade Pitch |
| $\mu$ | Advance ratio |
| $\Omega$ | Rotational speed of the blade |
| $a$ | Blade lift curve slope |
| $B$ | Rotation matrix |
| $A_b$ | Total blade area |
| $N$ | Blades number |
| $T$ | Thrust of main rotor |
| $A$ | Disk area of rotor |
| $R$ | Rotation matrix |
| $D$ | Drag |
| $L$ | Lift |
| $\mathbf{g}$ | Gravity vector |
| $\omega$ | Blade rotational speed |
| $c$ | Blade chord |
| $P_{tot}$ | Main rotor power |
| $F_x$ | Force acting in $x$-axis direction |
| $F_y$ | Force acting in $y$-axis direction |
| $F_z$ | Force acting in $z$-axis direction |
| $V_c$ | Climbing Speed of the rotor |
| $I_{xy}$ | Cross product moments of inertia around $xy$-axes |
| $I_{yz}$ | Cross product moments of inertia around $yz$-axes |
| $I_{xz}$ | Cross product moments of inertia around $xz$-axes |
| $L\ M\ N$ | Net torque to the angular momentum rate of change |
| $I_x$ | Moment of inertia about $x$-axis |
| $I_y$ | Moment of inertia about $y$-axis |
| $I_z$ | Moment of inertia about $z$-axis |
| $m$ | Mass |
| $CG$ | centre of gravity |
| $q_0,\ q_1,\ q_2,\ q_3$ | Quaternion parameters |
| $k_p$ | PID control proportional gain |
| $k_d$ | PID control derivative gain |
| $k_i$ | PID control integral gain |

# Chapter 1

# Introduction

## 1.1 Background and Motivation

Most real-world physical systems are highly nonlinear, complex, and uncertain. While traditional control methods heavily rely on the accessibility of accurate mathematical models, for many nonlinear systems (e.g., aerial robots), building comprehensive mathematical models is not always possible due to limited knowledge of the systems as well as multiple uncertainties in the systems. On the other hand, data-driven (system identification) techniques provide more realistic solutions to model the dynamics of complex systems. As universal approximators, computational intelligence approaches such as fuzzy logic systems (FLSs), neural networks (NNs), and Neuro-Fuzzy systems (NFSs) have proved to be successful computational tools to describe the behavior of complex dynamical systems [2–14].

With the advancement of computing technology, a large amount of information can be simultaneously processed to analyze the behaviour of nonlinear systems under different operating conditions. Fuzzy systems provide an alternative to describe the dynamics of nonlinear uncertain dynamical systems. Although type-1 fuzzy logic systems (T1-FLSs) have been applied to model various complex nonlinear systems, they have limited capability to handle uncertainties (e.g., due to errors in measurement, change of actuator characteristics, uncertainties associated with any change of the operating conditions or due to unmodeled dynamics). Aiming to accommodate uncertainties, type-2 fuzzy logic systems (T2-FLSs) (i.e., with interval membership functions) were established which introduce the concept of a *footprint-of-uncertainties* (FoUs) [15–21]. FoUs introduces new degrees of freedom, allowing a type-2 fuzzy system to represent uncertainty [22]. FoUs can be expressed as the union of all primary membership functions as it will be explained in the coming chapters.

In the last few decades, adaptive control systems have been a very topical research area. Under the umbrella of adaptive control, intelligent control approaches have been gaining popularity [23, 24] and such systems (e.g., FLSs, NNs, NFSs) have the ability to control complex, ill-defined, nonlinear, and time-varying systems in the face of uncertainties. For instance, in aerial robots, system parameters often differ between platforms and over time due to issues such as environmental changes, mechanical wear, damage to rotors, battery drain or ground effect [25, 26]. Various adaptive FLS/NNs based control methods have been proposed to cope with large uncertainties. This thesis introduces new adaptive control approaches, where the efficacy of the proposed systems is highlighted through rigorous numerical simulations in the face of various uncertainties.

The the practical implementation of T2-FLSs based control systems in real-time is somewhat restricted due to high computational costs, such as type-reduction techniques needed to convert the T2-FSs into their type-1 counterparts [22]. Therefore, to ease the computational burden while maintaining the benefits of T2-FLSs, interval type-2 fuzzy logic systems (IT2-FLSs) were introduced in [27], where the secondary grade variables of IT2-FLSs can be simply set to be equal to 1. In other words, IT2-FLS were utilized to accelerate the response to uncertain input membership functions, allowing greater freedom in designing the desired control law while providing the ability to accommodate more uncertainties typically found in nonlinear systems [28]. This thesis examines different type-reduction methods aiming to achieve less computation cost that is suitable for real-time control applications.

The use of IT2-FLSs for control can improve closed-loop control performance compared to type-1 FLS and achieves greater robustness to accommodate uncertainties as demonstrated in various research papers [19, 20, 22, 29–34]. Hence, the development of the proposed control systems, in this research work, is based on IT2-FLSs.

However, implementation of knowledge-based systems is often challenging, time-consuming, require expert knowledge, and can be inefficient to design [11, 14]. Besides, fuzzy rules derived by experts can be too ad-hoc, and the performance is less than optimum. Also, generating fuzzy rules and membership functions in fuzzy systems is a potential challenge especially for systems with many variables [35, 36]. Employing evolutionary algorithms to find optimal fuzzy parameters may be less desirable due to their computational demand. Under the umbrella of intelligent and adaptive control, this research investigates the development of evolving fuzzy systems (EFSs) that have the ability to autonomously learn fuzzy structure and parameters in an online manner. As a proof of concept, the work in this thesis is deployed in various autonomous systems. Recently, evolving fuzzy systems have become popular in various engineering applications. They are implemented for system identification, regression, classification and control [28, 37–55].

## 1.2 Scope of Research

This thesis represents the development and implementation of IT2-FLSs for modeling and control of nonlinear dynamical systems in the presence of various uncertainties. This research starts with highlighting the limitation of existing system identification techniques. It also discusses the limitations of the existing nonlinear control methods used for multirotor unmanned aerial vehicles (UAVs) and the feasibility of applying IT2-FLSs to real-time control applications. Fig. 1.1 illustrates the structure of this research work.

At first, this thesis concentrates on developing an efficient online system identifier based on IT2-FLSs for multirotor UAVs. The proposed method is based on the $C$-means clustering technique. A built-from-scratch quadcopter UAV is developed to highlight the efficacy of

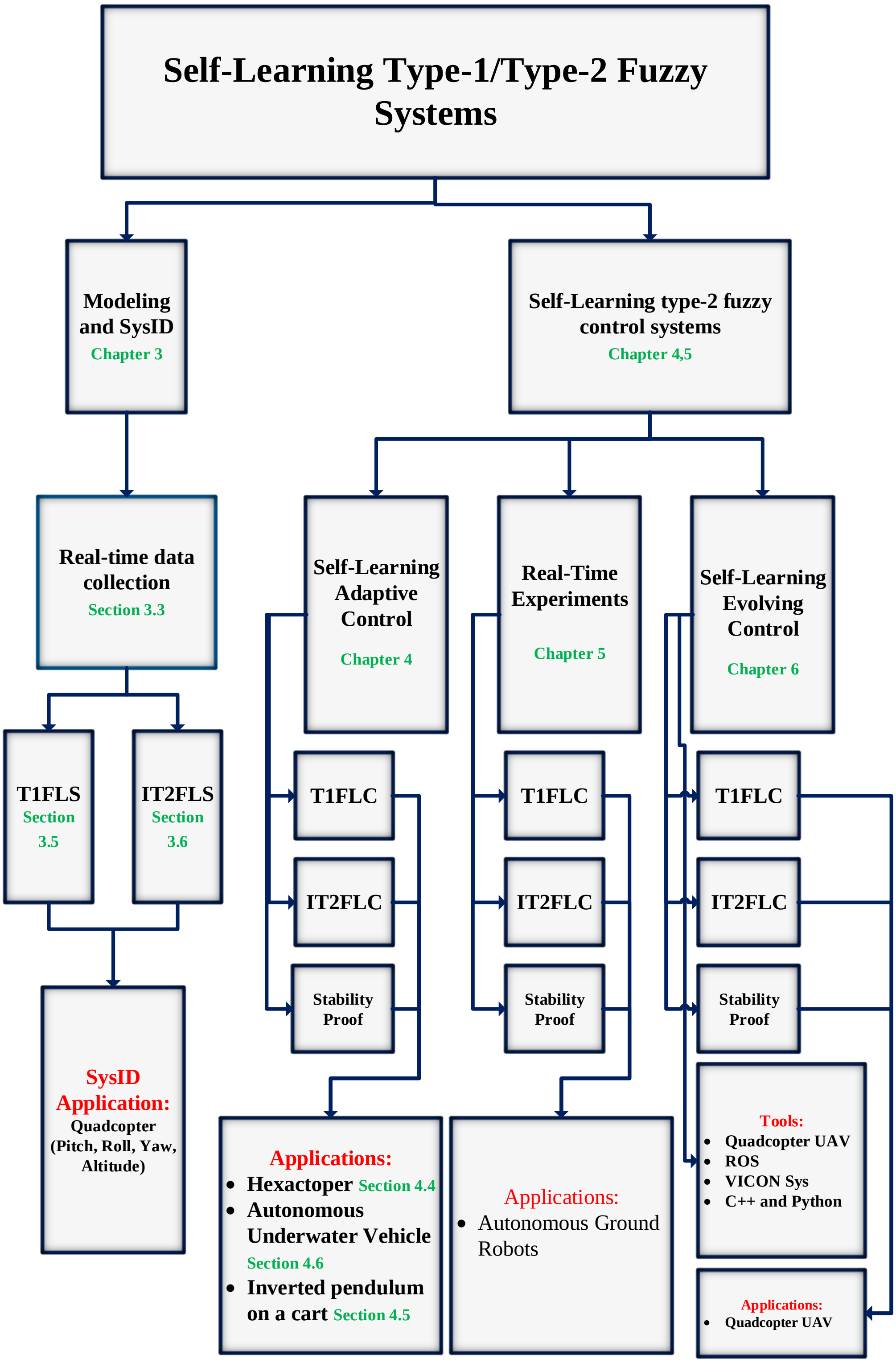


Figure 1.1: An Overview of this research.

the proposed system identification technique by conducting several flight tests to collect the necessary data using sensors on the UAV and an optical motion capture system.

Second, this thesis focuses on developing robust self-adaptive IT2-FLCs for MIMO nonlinear systems. The proposed technique is employed to control a MIMO multirotor drone. Furthermore, the robustness of the proposed approach is considered under different conditions, such as in the face of external disturbance (e.g. wind gusts) and parameter variations.

Third, this thesis focuses on the development of a novel stand-alone enhanced self-adaptive interval type-2 fuzzy control strategy for position control of a multirotor UAV, whose type-2 fuzzy parameters are tuned online using the sliding mode control theory. It also focuses on conducting real-time experiments under various disturbances to validate the efficacy of the proposed control system.

Finally, the thesis contributes to the development of an evolving fuzzy controller for nonlinear systems. The structure of the proposed method does not require previous information about the fuzzy structure and it starts its construction from scratch with only one rule. The rules can be added and deleted in an online manner to achieve the control objectives. The capability of the proposed technique is investigated in the face of uncertainties.

## 1.3 Contributions of the Thesis

The research contributions of this thesis include the following:

1. Development of a novel IT2-FLS online system identification model for a multirotor aircraft collected from real-time flight tests. The designed approach deploys the Lagrange approach to minimize the objective function. Unlike previous IT2-FLSs models, this method constructs two fuzzifiers (upper and lower) and two regression coefficients in the consequent part to handle uncertainties in multirotor UAVs. The algorithm has the capability to model uncertainties (e.g., noisy dataset).

2. Design of a new self-adaptive interval type-2 fuzzy controller named the *SAF2C* to control MIMO nonlinear dynamical systems. The algorithm deploys an efficient type-reduction method to reduce the computational burden of the Karnik-Mendel (KM) algorithm. The proposed SAF2C is employed to regulate the position and velocity of a simulated MIMO hexacopter UAV, where about 80% reduction of the execution time is achieved compared to the SISO SAF2C controller. The robustness of the proposed method is investigated under different conditions, such as in the face of external disturbances and parameter variations (e.g. wind gusts) for the case of the hexacopter UAV. The proposed approach has the ability to filter the measurement noise, where significant improvement is obtained using the SAF2C controller in the presence of measurement noise.

3. Implementation of real-time experiments using quadrotor UAVs to test the proposed adaptive fuzzy control systems. Specifically, the control performance is investigated in the face of external wind disturbance, using an industrial fan in the hover mode. The findings show that the proposed control technique has the capability to learn its parameters in an online manner and to handle external wind gust disturbance efficiently.

4. Development of a novel type-2 evolving fuzzy control system, named the T2-EFCS, which starts the construction of knowledge from scratch with only one rule. The fuzzy rules are then added and pruned in an online manner to achieve the desired set-point. Unlike most of the existing work, the proposed approach is model-free which does not require any information about the plant dynamics. Moreover, the adaptive law for parameter tuning is derived using sliding mode control (SMC) theory, making the system robust to variations in system parameters and external disturbance. The proposed control mechanism has been implemented to control an unmanned ground vehicle (UGV) in the presence of multiple external disturbances demonstrating promising tracking performance.

## 1.4 Organization of the Thesis

This thesis is divided into seven chapters. Chapter 2 describes the basis of T1-FLSs, T2-FLSs, and IT2-FLSs. It also reviews their application in modeling and control of nonlinear systems, followed by identifying the research gaps of the existing studies. Sequentially, Chapters 3 - 6 present the major technical contributions of this research work. In Chapter 3, the experimental setup for collecting real-time data is presented. It also proposes an online system identifier for modeling a quadrotor aircraft from real-time experimental flight data. Besides, the efficacy of the proposed approach is validated using a noisy dataset. Chapter 4 provides a new control technique based on IT2-FLSs with an efficient type-reduction technique that is suitable for real-time implementation and validated in various benchmark nonlinear systems. The robustness is investigated under different scenarios. Chapter 5 addresses the real-time implementation using a novel IT2-FLC. The efficiency against disturbances is also provided. Chapter 6 introduces a self-evolving fuzzy controller. The proposed method is model-free, meaning the control system does not require any information about the plant dynamics and its effectiveness is investigated in the presence of external disturbances. Finally, a summary of the findings of all technical contributions and some future research directions are identified in Chapter 7.

# Chapter 2

# Literature Review

Parts of this chapter have been taken from the following publications:

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). A Robust Self-Adaptive Interval Type-2 TS Fuzzy Logic for Controlling Multi-Input Multi-Output Nonlinear Uncertain Dynamical Systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems.*

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). Self-Learning in Aerial Robotics Using Type-2 Fuzzy Systems: Case Study in Hovering Quadrotor Flight Control. *IEEE Access.*

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). Modeling and control of a quadrotor unmanned aerial vehicle using type-2 fuzzy systems. *Unmanned Aerial Systems*, Advances in Nonlinear Dynamics and Chaos (ANDC), Elsevier.

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020).An Intelligent Control of an Inverted Pendulum Based on an Adaptive Interval Type-2 Fuzzy Inference System. *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).*

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020).A Simplified Model-Free Self-Evolving TS Fuzzy Controller for Nonlinear Systems with Uncertainties. *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS).*

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). Online System Identification for Nonlinear Autonomous Systems Using Recursive Interval Type-2 TS Fuzzy $C$-means Clustering. *2020 IEEE Symposium Series on Computational Intelligence (SSCI).*

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., Anavatti, S. G., and Ferdaus, M. M. (2019). Online Takagi-Sugeno Fuzzy Identification of a Quadcopter Using Experimental Input-Output Data. *2019 IEEE Symposium Series on Computational Intelligence (SSCI).*

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020).AA Robust Adaptive Interval Type-2 Fuzzy Control for Autonomous Underwater Vehicles. *2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT).*

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020).Nonlinear Altitude Control of a Quadcopter Drone Using Interval Type-2 Fuzzy Logic. *2018 IEEE Symposium Series on Computational Intelligence (SSCI).*

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). A Novel Learning-from-Scratch Evolving Type-2 Fuzzy System for Robotic Control. Preparing draft to be submitted to *IEEE Transactions on Intelligent Vehicles*, 2021.

## 2.1 Introduction

Inspired by the capability of fuzzy logic systems (FLSs) to learn complex nonlinear dynamics, they have been used to identify and model nonlinear dynamic systems. To address large uncertainties in dynamic systems, as well as to compensate for the systems' changing and uncertain environments such as external disturbances, the concept of interval type-2 fuzzy systems (IT2-FLSs) was developed, thanks to the footprint-of-uncertainties ($FOU$) in IT2-FLSs, to incorporate uncertainties efficiently.

Moreover, intelligent fuzzy logic controllers (FLCs) have been extensively employed to control nonlinear systems with complex dynamics. Inspired by the ability of FLCs to represent uncertainties in nonlinear systems by their continuous membership functions (MFs), they have been utilized to control various nonlinear systems. Nevertheless, since uncertainty information is not incorporated in the membership function of type-1 fuzzy sets (T1-FSs), controlling nonlinear systems subjected to uncertainties cannot be handled precisely. Therefore, interval type-2 fuzzy logic controllers (IT2-FLCs) were developed based on type-2 fuzzy sets (T2-FSs) to handle such uncertainties captured by their IT2 fuzzy membership functions.

To get a broad understanding of fuzzy systems, this chapter describes the basics of FLSs, IT2-FLSs, FLCs, IT2-FLCs, and their applications in the modeling and control of autonomous systems. To begin, a short review of type-1 fuzzy logic systems (T1-FLSs) is provided owing to the fact that IT2-FLSs build upon T1-FLSs. This chapter also provides a survey of their applications in the modeling and control of nonlinear dynamic systems. In addition, the concept of evolving fuzzy systems (EFSs) and their applications in modeling and control dynamic systems are discussed in this chapter. To provide context for applications of fuzzy logic control to unmanned autonomous systems (UASs), an overview of UASs is highlighted including unmanned aerial vehicles (UAVs), autonomous underwater vehicles (AUVs), and autonomous ground vehicles (AGVs). As unmanned autonomous systems are highly nonlinear, complex, and time-varying systems, they are convenient platforms for testing new algorithms. Finally, the research gaps of the existing methods are addressed, which motivates the research work pursued in this thesis.

The remainder of this chapter is structured as follows: the basic principles of T1-FLSs, T2-FLSs, IT2-FLSs and their associated concepts are presented in Sections 2.2. Subsequently, Section 2.3 provides the literature survey on modeling and system identification of dynamic systems. Section 2.4 presents methods of nonlinear control systems and a survey of the applications of IT2-FLCs in dynamic systems. The general design of EFSs is demonstrated in Section 2.5. This is followed by a brief review of Unmanned Autonomous Systems in Section 2.6. Also, the limitations of the existing studies are identified in Section 2.7. Finally, a summary of this chapter is provided in Section 2.8.

## 2.2 Basic Principles of Fuzzy Logic System

FLSs were firstly introduced by Zadeh in 1965 [56–58] and have been deployed in various engineering areas over the past few decades [55, 59–62]. Unlike conventional sets, a fuzzy set continuously defines the degree to which an object belongs to a certain set between '0' and '1' [14, 63, 64]. FLSs imitate human reasoning which derives a conclusion based on a set of expert fuzzy *If-Then* rules in a range between '0' and '1'.

Due to the necessity to accommodate uncertainties occurring in systems, Zadeh [57] introduced an enhanced version of the standard T1-FLSs which are nowadays referred to as T2-FLSs, where a new parameter called *footprint of uncertainty* was introduced. Although T1-FLSs reveal the denotation of uncertainties, they have a limited ability to model and to minimize the influence of uncertainties in nonlinear systems. Therefore, T2-FLSs can be an encouraging method to handle uncertainties associated with real-time applications as they incorporate uncertainties in their structure which are not incorporated in T1-FLSs [63, 65, 66]. In other words, the structure of T2-FLSs takes uncertainties into account and inherently encodes an uncertainty factor that is not addressed in T1-FLSs [64]. T2-FLSs appear to be more capable of handling problems with uncertainties and time-varying systems compared to their type-1 counterpart [30, 34, 64–67].

### 2.2.1 Type-1 Fuzzy Logic Systems (T1-FLSs)

#### 2.2.1.1 Type-1 Fuzzy Sets (T1-FSs)

In T1-FLSs, a fuzzy set $A$ can be represented as $(x, \mu_A(x))$, where $x \in X$, and $\mu_A(x) \in [0, 1]$. The $(MF)$ of $A$ is labeled as $\mu_A(x)$, while $X$ denotes the universe of discourse (which specifies the range of permissible values for a variable).

**Definition 2.2.1** *T1-FSs can be expressed as follows:*

$$A = \{(x, \mu_A(x)) | \forall x \in X, \mu_A(x) \in [0, 1]\}. \tag{2.1}$$

The fuzzy set $A$ can also be denoted in a continuous universe as:

$$A = \int_{x \in X} \mu_A(x)/x, \tag{2.2}$$

or for a discrete universe $X_{ds}$ as:

$$A = \sum_{x \in X_{ds}} (x, \mu_A(x), \tag{2.3}$$

where $\int$ and $\sum$ represents the collections of elements with a set $A$.

#### 2.2.1.2 T1-FSs - Theoretic Operations

T1-FSs can be aggregated using union (s-*norm*) operation, intersection (t-*norm*) operation and the complement operation. The (s-*norm*) operators are also known as (t-*conorm*). The maximum and the algebraic sum are fuzzy union, while the minimum and the algebraic product are fuzzy intersection. In the following equations, the minimum and the maximum representations are adopted.

**Definition 2.2.2** *Let $M_1$ and $M_2$ be two T1-FSs in A, which are represented by their MFs $\mu_{M1}(x)$ and $\mu_{M2}(x)$. The union of $M_1$ and $M_2$, $M_1 \cup M_2$, is expressed by its MF $\mu_{M_1 \cup M_2}(x)$, that is,*

$$\mu_{M_1 \cup M_2}(x) = max\Big[\mu_{M_1}(x),\ \mu_{M_2}(x)\Big], \quad \forall x \in X. \tag{2.4}$$

*The fuzzy intersection of $M_1$ and $M_2$, $M_1 \cap M_2$, is expressed by its MF $\mu_{M_1 \cap M_2}(x)$, that is,*

$$\mu_{M_1 \cap M_2}(x) = min\Big[\mu_{M_1}(x),\ \mu_{M_2}(x)\Big], \quad \forall x \in X. \tag{2.5}$$

*Additionally, the complement of a fuzzy set $A$ can be expressed by the MF $\mu_{\bar{A}}(x)$ as:*

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x), \quad \forall x \in X. \tag{2.6}$$

#### 2.2.1.3 Structure of T1-FLSs

The structure of T1-FLSs consists of four elements as follows (Fig. 2.1) [30]:

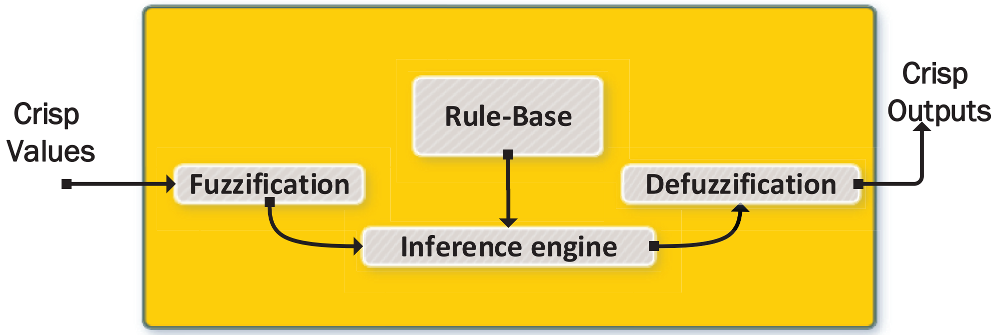


Figure 2.1: T1-FLSs structure.

- **Fuzzification**: this process transforms crisp values into fuzzy sets. The crisp values can be defined as the measured data from sensors (e.g. the altitude of an aircraft). In the fuzzification process, there are two types of fuzzifiers: singleton and non-singleton [22, 63].

- **Rule-base**: a set of rules, which consist of *antecedent* and *consequent* parts. These rules can be provided by experts or can be extracted from domain-specific data [22]. In this thesis, the first-order TSK rule structure is utilized, which can be presented as follows:

$$R^l:\ IF\ x_1\ is\ A_1^l\ and\ ...\ and\ x_j\ is\ A_j^l,\ THEN\ G^l = c_0^l + c_1^l x_1 + ... + c_j^l x_j,$$

where $x_i(t)$ denotes the $i$th input to fuzzy system; $A_i^l$ labels the T1-FSs of input state $i$ in rule $l$; $G^l$ is the output of the $l$th rule; $c_1^l, ..., c_j^l$ denote the coefficients of the output function, and $l = 1, ..., N$, where $N$ is the number of fuzzy rules. The linguistic term $A_i^l$ can be represented using several different shapes of *MFs* such as trapezoidal, triangular, Gaussian, Sigmoidal and bell-shaped [22,64] as shown in Fig. 2.2.

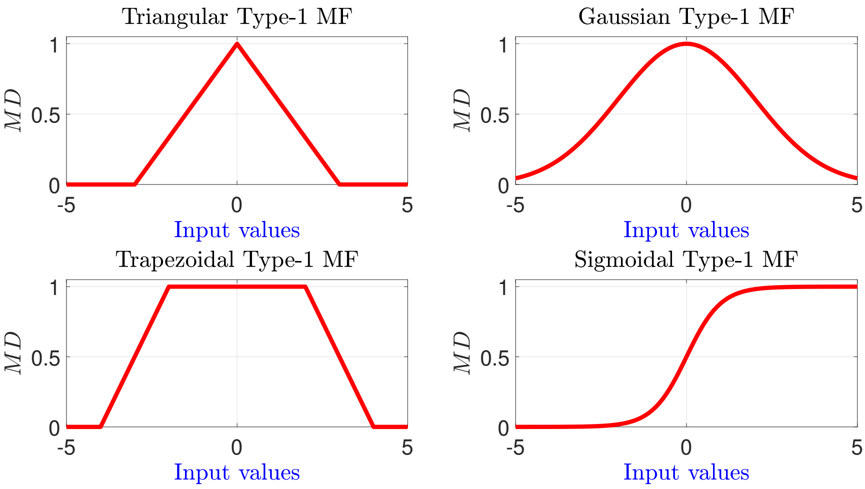


Figure 2.2: Different types of T1-FLSs membership functions, where *MD* denotes the membership degree.

- **Inference System**: this mechanism implements the required algebra to manipulate fuzzy sets. This manipulation can be performed using several algebraic operations as discussed in the theoretic operations subsection. The firing strength for each rule can be calculated using the product $t$-norm operator as follows:

$$f^l = \mu_{A_1^l}(x_1) \times ... \times \mu_{A_j^l}(x_j). \tag{2.7}$$

- **Defuzzification**: this step aggregates the fuzzy output sets into crisp values. There are various defuzzification methods including: centroid, center-of-sets, height and etc. The defuzzifier of a first-order TSK-type can be represented in two forms; the un-normalized TSK fuzzy system and the normalized TSK fuzzy system. The former can be represented using the following equation as described in [22]:

$$y = \sum_{l=1}^{N} f^l.G^l, \tag{2.8}$$

while the latter is described as follows:

$$y = \frac{\sum_{l=1}^{N} f^l.G^l}{\sum_{l=1}^{N} f^l}. \tag{2.9}$$

### 2.2.2 Type-2 Fuzzy Logic System (T2-FLSs)

T2-FLSs have been applied with great success in various engineering applications demonstrating better performance than T1-FLSs to handle uncertainties as reported in the literature. T2-FLSs have been implemented in numerous disciplines such as signal processing [68], pattern recognition [18], mathematical modeling [69] and control systems [70, 71]. The structure of T2-FLSs is depicted in Fig. 2.3.

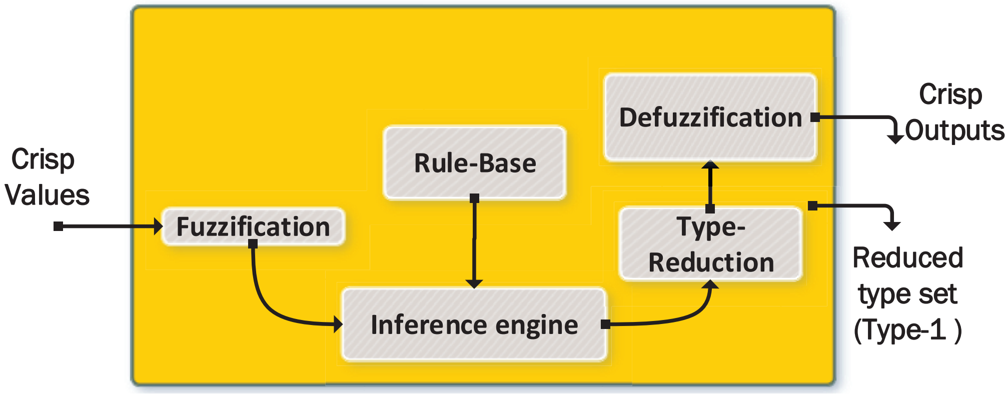


Figure 2.3: T2-FLSs structure.

In general, the scheme of T2-FLSs is similar to T1-FLSs, which includes a fuzzifier, a fuzzy inference engine, a rule-base, and an output processor. The main difference between them can be demonstrated in the defuzzification process, wherein T2-FLSs, a type-reduction is required before carrying out the defuzzification process [18–20, 72].

Uncertainties can affect decision-making in various ways, as the available data may be imprecise, incomplete, vague or fragmented [23, 31]. From a control perspective in FLCs, uncertainties can occur from input devices to the FLCs, which can be translated into uncertainties in the antecedents' membership functions. This could be caused due to input sensors, where their characteristics may be changed by the environmental conditions (e.g, wind, rain, humidity, sunshine, etc.). Also, a change of actuator characteristics (e.g, wear, tear of the actuators) can occur, causing various uncertainties in the overall control system. All the above-mentioned uncertainties can be translated into uncertainties in the overall FLCs structure [25, 73].

The utilization of T2-FLSs in real-time applications is moderately uncommon due to a high computational cost in the type-reduction operation needed to convert the type-2 fuzzy sets (T2-FSs) into their type-1 counterparts [22]. Therefore, to simplify the computational burden of the generalized T2-FLSs and to maintain their main advantages, IT2-FLSs were introduced [15, 27], where the secondary grade variables of an interval type-2 fuzzy sets (IT2-FSs) are set to unity = 1. In other words, IT2-FLSs were designed to accelerate the response to uncertain input membership functions, allowing greater freedom in designing the desired control law while providing the ability to accommodate more uncertainties typically found in nonlinear systems [16, 17, 28].

#### 2.2.2.1 Type-2 Fuzzy Sets (T2-FSs)

**Definition 2.2.3** *A typical type-2 fuzzy set, represented by $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x, u)$, where for every $x \in X$, and $u \in J_x \ \subseteq [0, 1]$ as follows [22]:*

$$\tilde{A} = \left\{((x, u), \mu_{\tilde{A}}(x, u)) | \forall x \in X, | \forall u \in J_x \subseteq [0, 1], \mu_{\tilde{A}}(x, u) \subseteq [0, 1]\right\}, \tag{2.10}$$

where $u$ is called the primary membership function and $\mu_{\tilde{A}}(x, u)$ is the secondary membership function. $\tilde{A}$ can be described in fuzzy set notation for a continuous universe as:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u)/(x, u) \quad J_x \subseteq [0, 1]. \tag{2.11}$$

or for a discrete universe as:

$$\tilde{A} = \sum_{x\in X}\sum_{u\in J_x}((x,u),\mu_{\tilde{A}}(x,u)) \quad J_x \subseteq [0,1]. \tag{2.12}$$

#### 2.2.2.2 Footprint-of-Uncertainties (FoUs)

Uncertainty in the primary membership function of a T2FS, $\tilde{A}$, is made up of the bounded region, as shown in Fig. 2.4, which is named the *footprint of uncertainty* (FoU). $FoU(\tilde{A})$ can be expressed as the union of all primary memberships, which can be defined as follows:

$$FoU(\tilde{A}) = \underset{x\in X}{\cup} J_x. \tag{2.13}$$

The upper bound of $FoU$ is called upper membership function ($UMF$), denoted by $\overline{\mu}_{\tilde{A}}(x)$ and the lower bound of $FoU$ is named lower membership function ($LMF$), which is denoted by $\underline{\mu}_{\tilde{A}}(x)$, that is,

$$\overline{\mu}_{\tilde{A}}(x) = UMF(\tilde{A}) \tag{2.14}$$

$$\underline{\mu}_{\tilde{A}}(x) = LMF(\tilde{A}). \tag{2.15}$$

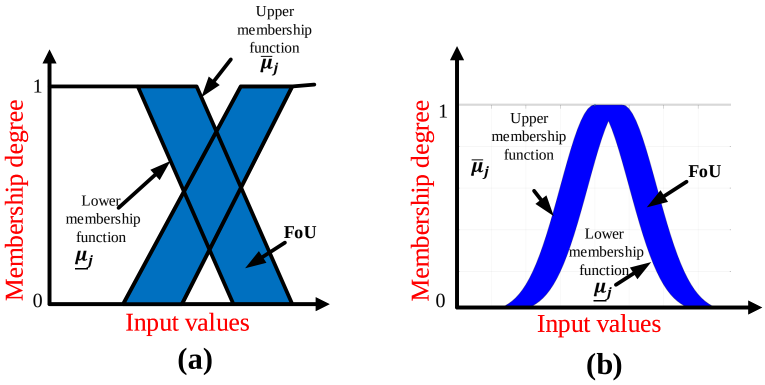

Upper membership function $\bar{\mu}_j$
1
0
Membership degree
Lower membership function $\underline{\mu}_j$
FoU
Input values
(a)
1
0
Membership degree
Upper membership function
$\bar{\mu}_j$
FoU
Lower membership function
$\underline{\mu}_j$
Input values
(b)

Figure 2.4: (a) Interval type-2 trapezoidal *MF*, (b) Interval type-2 Gaussian. *MF*

### 2.2.3 Interval Type-2 Fuzzy Logic Systems (IT2-FLSs)

IT2-FLSs were introduced as a method to reduce the computational cost while maintaining the main advantages of T2-FLSs [15]. The secondary grade values of IT2-FLSs are set to unity as shown in Fig. 2.4.

#### 2.2.3.1 Interval Type-2 Fuzzy Sets (IT2-FSs)

Mathematically, IT2-FSs can be expressed as:

$$\tilde{A} = \{((x,u),1) | \forall x \in X, | \forall u \in J_x \subseteq [0,1]\} \tag{2.16}$$

$\tilde{A}$ can be described in fuzzy set notation for a continuous universe as:

$$\tilde{A} = \int_{x\in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/1 \quad J_x \subseteq [0,1]. \tag{2.17}$$

or for a discrete universe as:

$$\tilde{A} = \sum_{x\in X} \sum_{u \in J_x} ((x,u),1) \quad J_x \subseteq [0,1]. \tag{2.18}$$

#### 2.2.3.2 IT2FSs - Theoretic Operations

IT2FSs can be aggregated using union, intersection and complement operations [20].

**Definition 2.2.4** *Let $\tilde{M}_1$ and $\tilde{M}_2$ be two IT2FSs in $\tilde{A}$, which are represented by their MFs $\mu_{\tilde{M}_1}(x)$ and $\mu_{\tilde{M}_2}(x)$. The union of $\tilde{M}_1$ and $\tilde{M}_2$, $\tilde{M}_1 \cup \tilde{M}_2$, is expressed as follows:*

$$\tilde{M}_1 \cup \tilde{M}_2 = 1/\left[\underline{\mu}_{\tilde{M}_1}(x) \vee \underline{\mu}_{\tilde{M}_2}(x), \quad \overline{\mu}_{\tilde{M}_1}(x) \vee \overline{\mu}_{\tilde{M}_2}(x)\right], \forall x \in X. \tag{2.19}$$

*The fuzzy intersection of $\tilde{M}_1$ and $\tilde{M}_2$, $\tilde{M}_1 \cap \tilde{M}_2$, is expressed as follows:*

$$\tilde{M}_1 \cap \tilde{M}_2 = 1/\left[\underline{\mu}_{\tilde{M}_1}(x) \wedge \underline{\mu}_{\tilde{M}_2}(x), \quad \overline{\mu}_{\tilde{M}_1}(x) \wedge \overline{\mu}_{\tilde{M}_2}(x)\right], \forall x \in X. \tag{2.20}$$

*and, the complement of IT2FSs $\tilde{A}$ can be expressed as:*

$$\overline{\tilde{A}} = 1/\left[1 - \overline{\mu}_{\tilde{A}}(x), \quad 1 - \underline{\mu}_{\tilde{A}}(x)\right], \forall x \in X. \tag{2.21}$$

#### 2.2.3.3 Structure of IT2-FLSs

The architecture of IT2-FLSs is similar to T2FLSs. It includes five different layers connected sequentially. The structural detail of each layer can be presented as follows:

1. **Layer 1 (the fuzzification layer)**: this process maps the input values into IT2-FSs. There are different types of membership functions to represent IT2-FSs such as trapezoidal and Gaussian membership functions as shown in Fig. 2.4.

2. **Layer 2 (Fuzzy Rules)**: the two main common rule of fuzzy rules for an IT2-FLSS: (1) Mamdani fuzzy rule, (2) Takagi-Sugeno (TS) type of fuzzy rule [74, 75].

   - Mamdani rule: in the Mamdani rule, the consequent part is represented by a fuzzy set. The structure of the Mamdani fuzzy rules can be described as follows:

     $$R^l : \ IF \ x_1 \ is \ \tilde{A}_1^l \ and \ ... \ and \ x_j \ is \ \tilde{A}_j^l, \ THEN \ y^l \ is \ \tilde{G}^l, l = 1, ..., N,$$

     where $R^l$ denotes $l^{th}$ fuzzy rule, $\tilde{A}^l$ labels linguistic terms of the $l^{th}$ rule and $j^{th}$ input attribute, while $\tilde{G}^l$ denotes the $l^{th}$ rule and $N^{th}$ output variables.

   - TS rule: in the TS fuzzy rules, the consequent part is a combination of linear and nonlinear functions of the inputs. The structure of a first-order TSK rule for an IT2-FLSS can be described as follows:

     $$R^l : \ IF \ x_1 \ is \ \tilde{A}_1^l \ and \ ... \ and \ x_j \ is \ \tilde{A}_j^l, \ THEN \ y^l = [c_0^l + c_1^l x_1 + ... + c_j^l x_j],$$

     where $R^l$ is the $l^{th}$ fuzzy rules, $\tilde{A}^l$ denotes linguistic terms, $c_j^l$ are the consequent polynomial parameters, $x_j$ denotes the fuzzy system inputs and $y^l$ is a linear combination of the $j$ states, which represent the fuzzy system output.

   There are advantages of both the Sugeno and the Mamdani methods. For the Sugeno method: (1) it is computationally efficient; (2) it can be represented as a linear function, resulting in working efficiently with linear control approaches (e.g., PID controller); (3) more suitable for mathematical analysis and optimization techniques. For the Mamdani method: (1) it is an intuitive approach; (2) more suitable for human interpretation [76].

3. **Layer 3 (Inference Engine)**: the function of the fuzzy inference engine is to combine all rules and gives the mapping from input fuzzy sets to output fuzzy sets. This layer is also called the firing layer, where the firing level is calculated. For T2-FLSs, the firing level is represented by a firing interval. This process can be achieved using the basic theoretic operations. The firing interval can be calculated as follows:

$$\begin{aligned} f^l\ (x_1,\ ...,\ x_n) &= [\underline{f}^l\ (x_1,\ ...,\ x_j),\ \overline{f}^l\ (x_1,\ ...,\ x_j)] \\ &= [\underline{f}^l,\ \overline{f}^l]. \end{aligned} \tag{2.22}$$

The expressions for $\underline{f}^l$ and $\overline{f}^l$ are given as follows:

$$\underline{f}_l = \prod_{l=1}^{j} \underline{\mu}_{\tilde{A}_j^l}, \qquad \overline{f}_l = \prod_{l=1}^{j} \overline{\mu}_{\tilde{A}_j^l}. \tag{2.23}$$

4. **Layer 4 (Type-Reduction)**: one major step in any rule-based fuzzy system is the process to convert a fuzzy set to a real number. In T1-FLSs, this process is done by the defuzzification process. For T2-FLSs, there are two possible avenues: (1) direct defuzzification, by a direct mapping of T2-FSs into a crisp value, or (2) by converting T2-FSs into T1-FSs (this process called *type-reduction*), then from T1-FSs into crisp numbers (defuzzification). There are many type-reduction approaches in the literature [17,77], such as centroid TR [78], the Karnik-Mendel (KM) TR [15], the enhanced KM TR [79], the enhanced iterative algorithm with stop condition (EIASC) [80] and the Nie and Tan closed-TR method [81]. In this research, we deploy several type-reduction methods including the EIASC and the NT closed-form TR algorithms to improve its efficiency for real-time implementation and also to reduce the computational burden of the KM algorithm.

   The Karnik-Mendel type-reducer is the most popular approach, which is an iterative method that calculates the left and right outputs $[y_l, y_r]$ for the centroids of T2-FSs (or Center-of-Sets (COS), Height, modified Height, etc). Hence, the $[y_l, y_r]$ can be computed as [78]:

$$y_l(x) = \frac{\sum_{i=1}^{L} c_l(\tilde{G})\overline{f}(x) + \sum_{i=L+1}^{M} c_l(\tilde{G})\underline{f}(x)}{\sum_{i=1}^{L} \overline{f}(x) + \sum_{i=L+1}^{M} \underline{f}(x)} \tag{2.24}$$

$$y_r(x) = \frac{\sum_{i=1}^{R} c_r(\tilde{G})\underline{f}(x) + \sum_{i=R+1}^{M} c_r(\tilde{G})\overline{f}(x)}{\sum_{i=1}^{R} \underline{f}(x) + \sum_{i=R+1}^{M} \overline{f}(x}, \tag{2.25}$$

where $y^l$ in layer 2 $\in [c_l(\tilde{G}),\ c_r(\tilde{G})]$, which replaces the IT2FS, $\tilde{G}$, using the COS type-reducer; $L$ and $R$ are called the switching points. These points have no closed-form solutions; however, they can be computed using the KM method or the EIASC algorithm. One interesting property of the EIASC algorithm is that no rule-reordering is required; unlike the KM-TR algorithm, where the consequent parameters need to be reordered in ascending order. Hence, EIASC has been adopted in our proposed designs.

5. **Layer 5 (Defuzzification)**: the final step is to calculate the crisp defuzzified output as follow:

$$Y(x) = (y_l(x) + y_r(x))/2. \tag{2.26}$$

#### 2.2.3.4 Several Considerations on Practical IT2-FLSs

There are several considerations when designing practical IT2-FLSs, which can be summarized as follows [19, 82]:

- Fuzzifier (singleton or non-singleton): the recommendation in [19] refers to the utilization of a non-singleton fuzzifier in the presence of measurement noise. Otherwise, the designer should start with singleton fuzzification due to its simplicity. For singleton fuzzification, the measurement is considered to be perfect and modeled as crisp values, while for non-singleton fuzzification, the measurements are considered to be corrupted and modeled as fuzzy sets [75],

- Membership function type: the two most widely used membership functions are Gaussian and piecewise linear functions. For Gaussian IT2 membership function, three parameters are required to define it $(m_1, m_2, \sigma)$ or $(m, \sigma_1, \sigma_2)$, where $m$ is the mean and $\sigma$ is the width. On the other hand, nine parameters are needed to construct a piecewise linear *MF*. The trapezoidal MF shape is the most widely used piecewise linear MF form in fuzzy systems [19]. Hence, one can conclude that it is simpler to use an IT2 Gaussian membership function as it requires a fewer number of parameters. However, the analytical structure of an IT2 piecewise function is easier to derive [19].

- Number of membership functions: although there is no restriction on the number of *MFs*, in practice some limitations may hinder the designer from using too many *MFs*. In [19], it is recommended to utilize $\leq 7$ *MFs* for each input domain to facilitate interpretation and to minimize computational cost.

- Optimization: For the sake of performance and accuracy, it is important to optimize fuzzy *MFs*. In the literature, the most widely used methods are the steepest descent algorithm (which is also referred to as the back-propagation technique), and evolutionary computation algorithms [82].

- Computational Cost: the use of traditional Karnik-Mendel (KM) type-reducer in the real-world application might be rare due to its computational cost. Hence, several algorithms are proposed to reduce the computational cost such as the Enhanced Iterative Algorithm with Stop Condition (EIASC) [83], Enhanced KM algorithm [22], optimized direct approach [84]. EIASC is recommended for its speed and simplicity [19, 74].

## 2.3 Modeling and System Identification of Dynamic Systems

Numerous researchers have studied techniques to model nonlinear dynamic systems. These techniques include: (1) first principle modeling approach, also known as, the direct computation modeling approach, which is based on physics laws and also known as white-box approach, and (2) experiment-based system identification approach, either black-box or grey-box techniques, also known as data-driven modeling approach [11, 60, 85]. The black-box approach requires observation of the input-output data in order to parametrize a function. The white-box approach is based on comprehensive first-principles derivations. Moreover, the grey-box, which is connected to semi-physical modelling and the steel-grey technique, is based on the concept of developing local modes (linearization) to handle nonlinearities [37]. While the first principle modeling technique has several merits and can provide an accurate understanding of systems' dynamics, this approach is impractical in many dynamic systems due to the associated complexity and the unknown dynamic behaviors [85]. On the other hand, experiment-based data-driven modeling techniques using input-output data may achieve more accurate and realistic solutions, especially in the case of aerial robotics, where the system is not only nonlinear but also uncertain. There are several ways to describe the relationship between input/output variables. These include the polynomial model using the linear/nonlinear Auto-Regressive with eXogenous input (ARX/NARX), and linear/nonlinear autoregressive moving-average (ARMA/NARMA) [86]. Although these techniques appear to be easy to implement, they have several constraints; particularly when it comes to highly nonlinear systems, in addition to constraints in the presence of other uncertainties [4, 85].

There exist some challenges in these data-driven techniques, such as the need to conduct several experiments before obtaining a robust model [87]. However, with the advancement of technology, sensing methods and fast processors, large amounts of information can be obtained which can be utilized to analyze the performance of a dynamic system in different operating conditions without the need to understand the mathematics of the system [11, 88].

To overcome the issue of the traditional data-driven modeling techniques, artificial intelligence based-methods such as fuzzy logic systems, neural networks, and neuro-fuzzy systems are used, where the exact analytical model can be avoided [13,14,89]. These methods have been employed successfully to model nonlinear systems, thanks to their universal approximation and their learning capabilities [2–10, 12, 36]. In other words, these systems can process vague data and produce acceptable outputs without the need for complex mathematical computations.

### 2.3.1 IT2-FLSs for System Identification of Nonlinear Systems

Fuzzy system identification techniques have been extensively used in various nonlinear systems to obtain accurate models using input-output data [3, 11, 14, 85, 90]. One of the most popular fuzzy modeling approaches is the Takagi-Sugeno (TS) fuzzy-based model. The main task of the TS fuzzy model is to build several local models that can approximate the dynamics of a nonlinear system [90]. The construction of the TS fuzzy model consists of three phases: (1) fuzzy structure identification; (2) fuzzy parameter identification; (3) model validation [91]. There are various automated methods for developing fuzzy models such as: (1) Batch least squares (BLS); (2) Recursive least squares (RLS); (3) Learning from example (LFE); (4) Modified learning from example (MLFE); (5) Gradient method (GM); and (6) Clustering method (CM) [92]. Fig. 2.5 demonstrates the architecture of FLSs for online system identification of dynamic systems.

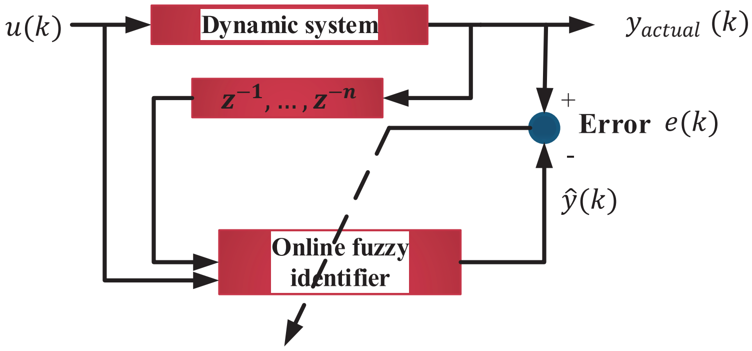


Figure 2.5: Architecture of online fuzzy system identification for dynamic systems.

Although T1FLSs have been applied to model various complex nonlinear systems, they have limited capability to handle uncertainties [30, 74, 90]. Therefore, T2-FLSs (i.e. with interval membership functions) were established [15–17].

The most significant process to establish a fuzzy model is the structure identification that is concerned with the selection of fuzzy inputs, the number of rules, and the membership functions [93]. Among the most common methods for fuzzy structure identification is the fuzzy clustering technique. There are various fuzzy clustering methods in the literature, such as the Fuzzy $C$-means (FCM) by Bezdek [94], modified Gath-Geva fuzzy clustering [95] and fuzzy C-regression model (FCRM) [96]. Nevertheless, type-1 fuzzy clustering techniques have limited ability to handle uncertain behaviors in real systems [97, 98]. Therefore, an interval type-2 fuzzy method based on the $C$-means clustering technique is proposed in [99] to handle uncertainties. A type-2 fuzzy clustering method using a differential evolution optimization approach was proposed in [100]. Moreover, a modified interval type-2 FCM approach was presented in [90, 101]. All of these methods were proposed to accommodate uncertainties in nonlinear systems and to improve the performance over the type-1 fuzzy counterpart.

## 2.4 Control of Dynamic Systems

There are several factors that involve designing closed-loop control systems (Fig. 2.6). Some of these factors can be summarized as follows [102]:

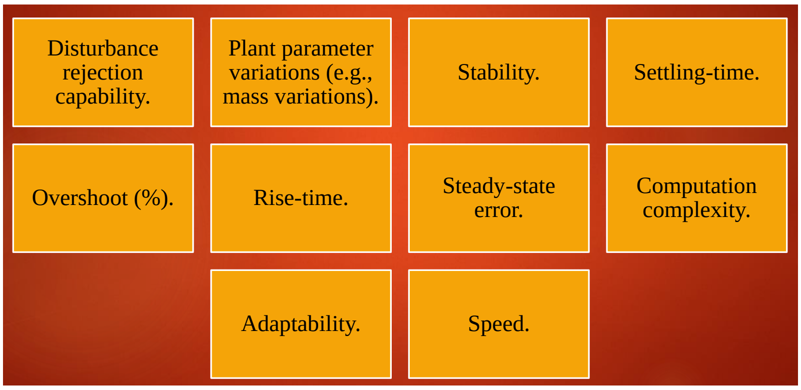


Figure 2.6: Some of performance objectives in closed-loop control design.

- **Disturbance rejection capability**: disturbance can affect the performance of the overall control system (e.g., wind gust, measurement noise) in aerial robotics [103].
- **Stability**: it is crucial to guarantee that the system output will converge to the desired set-point.
- **Ability to handle system parameter variations**: the control system should have the ability to compensate for variation of parameters (e.g., variation of mass,

external load in aerial robotics).

- **Rise-time**: defined as the time required for a unit-step response to rise from 10% of the steady-state value to 90% of the steady-state value [104].
- **Overshoot**: it can occur when the actual output exceeds its reference/desired input and can be described as [104]:

$$\%\ Overshoot = 100 \times \frac{Max\ value - steady\ state\ value}{steady\ state\ value}. \tag{2.27}$$

- **Settling-time**: defined as the time required for a step response of a system to reach within 2% of its set-point value [105].
- **Steady-state error**: defined as the error between the desired set-point and the actual output.
- Others: such as computation complexity, adaptability, and speed.

### 2.4.1 Control Methods

There are myriads of methods for constructing stable control systems. Some of these methods are listed as follows (Fig. 2.7) [102]:

- **Proportional-integral-derivative (PID) controller**: PID controllers are considered as reliable, simple and easy to understand and to tune. Control engineers often use intelligent controllers (e.g., fuzzy controllers, neural network controllers) to tune PID control parameters. The general continuous-time linear PID controller form can be described by the following expression:

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right), \tag{2.28}$$

where $e(t)$ represents the error signal, that is, $e(t) = r(t) - y(t)$, where $r(t)$ is the reference signal; $y(t)$ is the output of the system; $K$ is a gain; $T_i$ is the integration

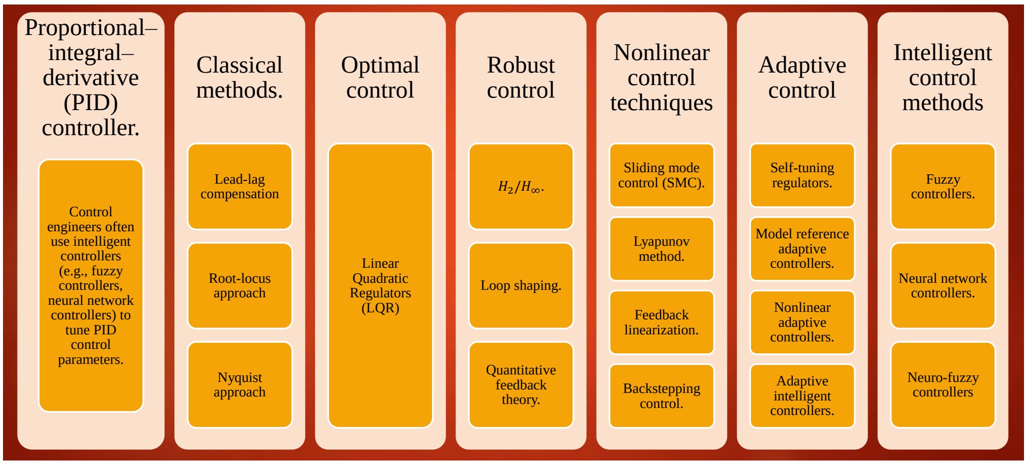

Proportional–integral–derivative (PID) controller.
Control engineers often use intelligent controllers (e.g., fuzzy controllers, neural network controllers) to tune PID control parameters.
Classical methods.
Lead-lag compensation
Root-locus approach
Nyquist approach
Optimal control
Linear Quadratic Regulators (LQR)
Robust control
$H_2/H_\infty$.
Loop shaping.
Quantitative feedback theory.
Nonlinear control techniques
Sliding mode control (SMC).
Lyapunov method.
Feedback linearization.
Backstepping control.
Adaptive control
Self-tuning regulators.
Model reference adaptive controllers.
Nonlinear adaptive controllers.
Adaptive intelligent controllers.
Intelligent control methods
Fuzzy controllers.
Neural network controllers.
Neuro-fuzzy controllers

Figure 2.7: Types of closed-loop control systems.

time; and $T_d$ is the derivative time. Nevertheless, for complex, nonlinear and time-variant systems, the effectiveness of conventional controllers becomes poor [106,107].

- **Classical methods**: which include lead-lag compensation, root-locus approach, Nyquist approach.
- **Optimal control**: such as Linear Quadratic Regulators (LQR).
- **Robust control**: (e.g., $H_2$ or $H_\infty$ techniques, loop shaping, quantitative feedback theory).
- **Nonlinear control techniques**: (e.g., sliding mode control (SMC), Lyapunov method, feedback linearization (FBL), backstepping control).
- **Adaptive control**: (e.g., self-tuning regulators, model reference adaptive controllers, nonlinear adaptive controllers, adaptive intelligent controllers).
- **Intelligent control methods**: (e.g., fuzzy controllers, neural network controllers, neuro-fuzzy controllers). A more detailed explanation of intelligent controllers is discussed in the following section.

### 2.4.2 Intelligent Control - Fuzzy Controllers

While some conventional control strategies intensely rely upon the exact model of controlled process/system, FLCs can be designed instinctively based on the knowledge gained about a systems' behaviors. This knowledge is often acquired through experience and common sense. FLCs rely on the reasoning approximation that derives a decision based on a set of fuzzy *IF-THEN* rules that mimic human-like reasoning [14]. FLCs are a practical option for a variety of complex control applications for their suitability to construct nonlinear controllers via the use of heuristic knowledge, where such heuristic knowledge can be acquired from experts who act as *human-in-the-loop* controllers for a certain process [102]. The general structure of the closed-loop FLCs structure can be depicted in Fig. 2.8. In this thesis, several type-1/type-2 fuzzy control strategies are proposed and discussed in detail.

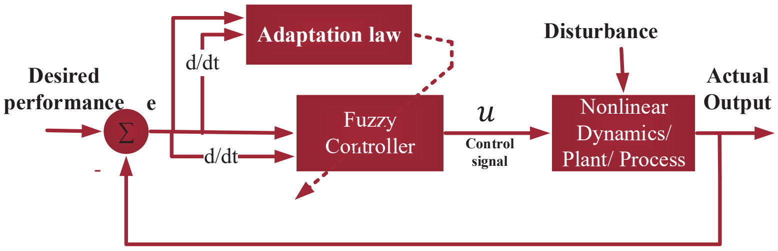


Figure 2.8: Closed-loop FLCs structure with disturbance signal.

### 2.4.3 Intelligent Control - Neural Networks Controllers

In recent years, neural networks (NNs) have attracted great attention and have been used in many different applications. NNs-based controllers can be regarded as one of the adaptive nonlinear controllers consisting of interconnected neurons which exchange information with each other [14]. Inspired by the structure of neurons, NNs were developed to emulate the learning capability of biological neural systems [48, 108, 109].

The basic mathematical model of shallow NNs comprise of three layers: (1) the input

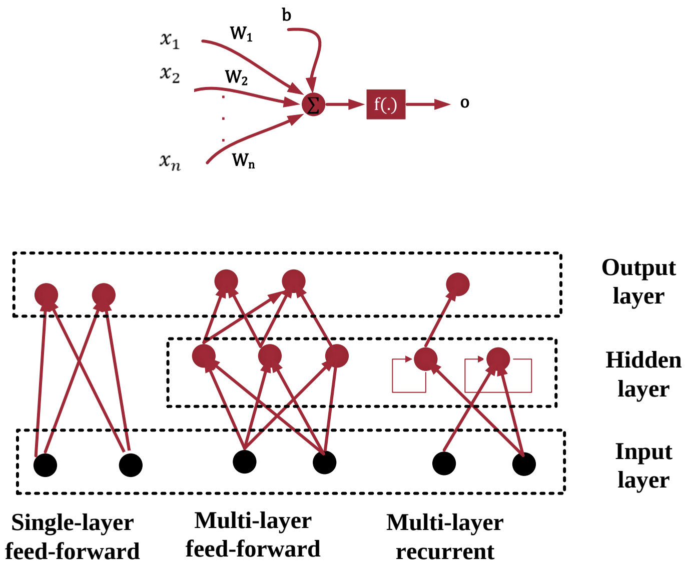


Figure 2.9: Neural networks structure. (top) perceptron neuron structure, (bottom) Different network architectures.

layer; (2) the hidden layer; (3) and lastly the output layer. Unlike FLSs, NNs have the capability of learning and the ability of parallel processing [14, 110]. In the first stage, the inputs $[x_1, x_2, ..., x_n]$ are multiplied by their respective weights $[w_1, w_2, .., w_n]$, so that the resulting summation, $net = (w_1.x_1 + w_2.x_2 + ... + w_n.x_n)$. This can be written as [111]:

$$net = \left(\sum_{i=1}^{n} w_i.x_i\right) = W^T X. \tag{2.29}$$

Additionally, a threshold value $b$, named *bias*, plays an important role in some NNs models, so that Eq. (2.29) can be expanded as:

$$net = \left(\sum_{i=1}^{n} w_i.x_i\right) + b. \tag{2.30}$$

Lastly, the final output can be described as:

$$y = f(net) = f(W.x + b) \tag{2.31}$$

where $f(.)$ is a nonlinear activation function in the neuron arrangement. There are variety of activation functions. Among them: (1) the *sign* function, (2) the *step* function, (3) the *sigmoid* (logistic) function, (4) the *exponential* activation function, (5) the *reciprocal* activation function, (6) the *Gaussian* activation function, (7) the *saturation* activation function [112]. Various NN architectures are represented in Fig. 2.9, where the top figure represents the architecture of a simple perceptron neuron model, while the bottom figure elaborates various NN architectures, which include single layer feed-forward network, multi-layer feed-forward network, and multi-layer recurrent network [113].

### 2.4.4 Intelligent Control - NeuroFuzzy Controllers

FLCs and NNs have shown the ability to deal with complex, nonlinear and time-varying systems in the presence of uncertainties due to their universal approximation and learning capabilities. Manual tuning of FLCs can be a tedious task and time-consuming for human operators to examine all the possible input-output data scenarios to find an optimal solution of dynamic systems [114]. By merging the human-like reasoning style of FLCs with the learning structure of NNs, neuro-fuzzy hybridization produces a hybrid intelligent system that synergizes these two methodologies. The lack of interpretability of NNs, on the one hand, and the poor learning power of fuzzy systems, on the other, are both issues that limit their use. Neuro-fuzzy systems are hybrid systems that attempt to tackle this challenge by merging connectionist models' learning capabilities with fuzzy systems' interpretability attribute [14, 42, 113, 115]. Several studies have been investigated to automate knowledge acquisition for FLSs by the means of automatic learning using NNs. There can be several ways to combine FLCs with NNs. Among them [111], NNs can be utilized to fine-tune FLSs parameters, to learn fuzzy rules and to determine the number of *MFs*.

### 2.4.5 IT2FLCs in Nonlinear Control Systems

Most robotic systems are dynamically unstable, nonlinear, and are multi-input multi-output (MIMO) systems, and therefore, robust control systems are required to stabilize them [98]. The classical model-based control approaches such as PID controllers [116], LQR [117], and model predictive controllers (MPC) [118] can provide optimal control performance when the model is well-defined, precise and there are no external uncertainties.

Nevertheless, there are inevitable uncertainties, (e.g. lack of modeling, mechanical wear, rotor damage, battery drain, and sensor and actuator faults [25, 119]). In the face of the aforementioned uncertainties, the control performance of conventional methods becomes poor [30]. In other words, designing a reliable closed-loop control system is a challenging task in the face of uncertainties [120].

In general, adaptive control methods have proven to be more effective than fixed gain controllers [30, 98]. Intelligent control systems such as FLCs, NNs-based controllers, and their combinations have been extensively employed to control nonlinear systems with complex dynamics [23, 121–124]. Under the umbrella of FLCs, Takagi-Sugeno (TS) fuzzy control systems have several advantages. Among them, TS fuzzy systems allow a description of a nonlinear system via a set of local linear system domains with corresponding well-designed *MFs* [125]. Moreover, FLCs can represent uncertainties in nonlinear systems by their continuous *MFs* [22].

Nevertheless, since uncertainty information is not incorporated in the membership function of type-1 fuzzy logic control systems (T1FLCs), controlling nonlinear systems subjected to uncertainties cannot be handled precisely. In other words, parameter uncertainties in nonlinear systems may lead to uncertain membership degree, and hence, the scheme of T1FLCs become more conservative [126]. To overcome this drawback, type-2 fuzzy logic control systems (T2FLCs) and interval type-2 fuzzy logic control systems (IT2FLCs) can be a good solution to achieve higher accuracy and greater robustness to system uncertainties as demonstrated in various research papers [19, 20, 22, 29–31, 34, 127–129].

Several studies combine T2FLCs and the sliding mode control (SMC) theory. Combining fuzzy systems with SMC theory has the advantages of achieving higher computational efficiency and improving the performance of the system by eliminating the chattering effect of the SMC controllers [128, 130, 131]. IT2FLCs based on SMC were implemented to control nonlinear systems such as unmanned aerial vehicles (UAVs) [29, 132], and also for obstacle avoidance and control for mobile robots [114] achieving reasonably good results. Nevertheless, the control law is designed as a combination with other conventional control approaches (e.g, PID controller) demanding additional tuning for its parameters. Similarly, in [133], a hybrid IT2FLC-based PID controller was proposed for a power system, where the fuzzy parameters were tuned using the firefly algorithm-particle swarm optimization technique. Nevertheless, their proposed representation requires extra tuning for the PID parameters.

In the work by [107], a fault-tolerant control based on interval type-2 fuzzy neural networks and sliding mode controller was developed for a 6-DOF octocopter UAV. Although their proposed controller can guarantee the stability of the proposed control system, it lacks experimental validation.

In [134–136], the SMC technique was also implemented with fuzzy systems to improve the robustness of dynamic systems. Although their studies achieved good tracking performance, the closed-loop control design was based on multiple SISO systems, which is less efficient than having a single MIMO controller in the loop. In [136, 137], IT2FLCs based on SMC were designed for an inverted pendulum, where simulation results demonstrated the robustness of the system. However, their proposed control design was based on the KM type-reduction method, which is computationally intensive.

In recent studies, IT2FLCs have been combined with other control techniques to handle parameter uncertainties, disturbance rejection and robust stabilization. In [138], a $H_2/H_\infty$ based IT2FLCs was proposed for nonlinear systems with minimal control effort to handle uncertainties such as disturbance and measurement noise, where better tracking performance was obtained compared to a T1FLCs. An IT2FLCs-based MPC was considered for nonlinear networked control systems in the presence of parameter uncertainties and defective communication links with good results in [139]. Moreover, a robust nonlinear control system based on the FBL technique, assisted with IT2FLCs was proposed in [140], for controlling the dynamics of a flapping-wing vehicle.

## 2.5 Evolving Fuzzy Systems (EFSs)

FLSs can provide a large avenue for classifying, modeling and controlling various applications. However, traditional approaches of deriving fuzzy rules are often tedious, time-consuming, require knowledge from experts and inefficient [14]. Moreover, the fuzzy rules derived by an expert are too heuristic which means they cannot be verified analytically due to their nontrivial nature. Therefore, the development of automatic fuzzy evolving systems is required. Generating fuzzy rules and membership functions in fuzzy systems is a potential challenge especially for systems with many variables [35]. Moreover, employing evolutionary algorithms to find optimal fuzzy parameters is not desirable due to the large population space which results in slow performance. Hence, researchers have proposed several evolving fuzzy systems (EFSs) for the structure and parameter learning of FLSs. EFSs can be defined as adaptive intelligent systems, which can learn their fuzzy structure and parameters simultaneously in an online manner [55]. Table 2.1 shows the main differences between adaptive and evolving fuzzy systems. In [141], EFSs were referred to as *Smart Adaptive Systems*, where they differ from 'simple' adaptive systems and have the following features: (1) autonomy, where they are able to evolve on their own; (2) flexibility to change, where they are able to simultaneously evolve both their structure and parameters; (3) ability to respond to a surprise (e.g., unexpected inputs); (4) ability to accumulate experience (e.g., able to build-up their architecture during the routine process); (5) they are smart (able to make decisions). The basic architecture of smart adaptive systems/ EFSs is illustrated in Fig. 2.10 [141].

Table 2.1: Adaptive vs. evolving fuzzy systems

| Method | Fuzzy parameters | Fuzzy structure | Number of rules | Location of *MFs* | Consequences |
|---|---|---|---|---|---|
| Adaptive | Adjusted | Fixed | Fixed | Adjusted | Adjusted |
| Evolving | Adjusted | Adjusted | Adjusted | Adjusted | Adjusted |

The field of evolving intelligent systems (EIS) was first conceptualized in [142], for single-pass incremental learning, with self-constructing neural fuzzy inference network (SONFIN) capabilities. SONFIN can learn both structure and parameters in an online manner.

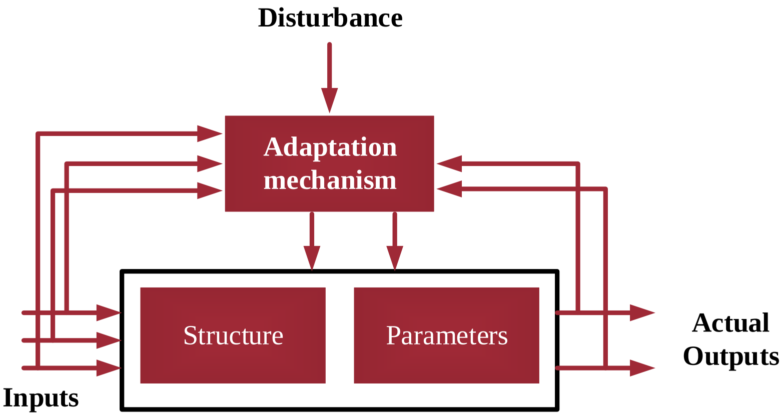


Figure 2.10: Basic architecture of smart adaptive systems/ EFSs

Nevertheless, it does not have the ability to remove ineffective rules. Besides, Kasabov and Song [143] proposed a new technique named a dynamic evolving neural fuzzy inference system (DENFIS) based on the evolving clustering method (ECM), where ECM requires a threshold value for defining the maximum distance between each date sample and cluster centers. Angelov and Filev [144] have developed an evolving Takagi-Sugeno (eTS) fuzzy system that can evolve its structure from scratch without any prior knowledge of the system. However, their proposed algorithm is not capable to eliminate useless rules. In [145], a sequential adaptive fuzzy inference system (SAFIS) was proposed, where the Kalman filter is utilized for updating the parameters. Angelov [146] developed a new version of eTS called eTS+, where the online dimensionality was reduced compared to the original eTS. In [47], a parsimonious network based on fuzzy inference system (PANFIS) was developed as an improved version of SAFIS, where the extended recursive least square method was utilized to guarantee the stability of the learning algorithms.

The integration of EFSs and IT2-FLSs was first introduced by Juang and Tsao [147], where an improvement over their proposed SONFIN is made using T2-FSs. In [147], a self-evolving FLS is developed to generate fuzzy rules from scratch for modeling nonlinear systems. Their proposed algorithm utilized the Kalman filter for parameter learning of the consequent part while the antecedent part was learnt by the gradient descent method. The same algorithm was implemented in [148] for system identification of nonlinear systems. In [149, 150], fuzzy rules were generated from input-output data and an entropy criterion was used to determine when to extract and to generate a new fuzzy rule. However, one drawback of this framework is that once the rule is created, it cannot be pruned. Also, two rules should be combined into a single fuzzy rule if they are similar to each other. This combination reduces the complexity of rule base design and improves the readability and interpretability of rule semantics [151, 152].

Recently, EFSs have become popular in various engineering applications. They have been used for system identification, regression, classification and control [28, 37–54, 153]. In general, EFSs are made up of three major elements: (1) rule addition; (2) rules pruning; and (3) learning of fuzzy parameters [50]. Besides, some new concepts such as rules merging and rules splitting were introduced [55].

### 2.5.1 Structure Learning (Rule Addition in EFSs)

Rule addition is the most essential element in EFSs [55]. A new rule is added when new samples contain important novelty content. There are various techniques to determine when and in which place to add a fuzzy rule in the data space. In other words, new fuzzy rules are generated when satisfying certain thresholds. Among these thresholds [5, 50, 55, 154]: (1) based on the current error of the system (the difference between the measured and the estimated outputs); (2) measures of distance (e.g., Euclidean distance); (3) density-based approaches (e.g., to measure how close the current sample to the recent estimation of the density); (4) based on the guarantee of $\epsilon - completeness$ , that is related to the membership degree/ firing strength of the current sample. Besides, several other methods such as the entropy [37], and mutual information [47] are utilized for adding new rules in EFSs.

### 2.5.2 Structure Learning (Rule Pruning in EFSs)

Mechanisms of pruning rules in EFSs are convenient to remove inactive rules, which are no longer useful [155]. This reduces the redundancy of fuzzy models and also results in faster computation time during the adaptation of parameters. Various pruning techniques were studied in the literature to guarantee a successful pruning of unnecessary rules. One of the most widespread approaches for pruning existing rules is based on the contribution of membership grade, so when it is smaller/bigger than a prior threshold value, the rule is deleted [156]. In [157], the mechanism of removing rules was based on the age of the rule. Besides, the deletion of rules was based on the contribution of a fuzzy rule to the model output error in [145, 158, 159].

### 2.5.3 T1/T2 EFSs in System Identification of Nonlinear Systems

In recent years, evolving systems have been successfully employed for nonlinear system identification. Most real-world systems are nonlinear, which may vary over time. In other words, the behavior of these systems changes over time. Nevertheless, a large amount of data are generated from multiple sources, which may result in high dimensional data [55]. Hence, the use of evolving systems is essential to handle and to process big data in real-time [160, 161].

There are various intelligent evolving methods in the literature, which are based on FLSs, NNs or neuro-fuzzy-based concepts. Among the EFSs models [55]: evolving Takagi-Sugeno (eTS) [141,144,160], eTS+ [146], flexible fuzzy inference systems (FLEXFIS) [162], FLEXFIS+ [163], generalized smart evolving fuzzy systems (GS-EFSs) [164], fuzzy set based evolving modeling (FBeM) [165], type-2 evolutionary Takagi-Sugeno fuzzy inference systems with information entropy-based pruning technique (T2-ETS-IE) [37] and evolving fuzzy model (eFuMo) [166].

In [167], an evolving neuro-fuzzy system was developed for modeling a hexacopter plant, where the learning mechanism was built from scratch with very limited knowledge about the system. In another study by [168], a self-adaptive Takagi-Sugeno fuzzy system was developed for online identification of a UAV quadcopter using $C$-means clustering technique. However, their proposed technique was based on T1-FLSs which have a limited capacity to accommodate the $FoU$.

To accommodate more uncertainties in capturing the dynamics of nonlinear systems, several EFSs based on T2-FLSs were proposed in the literature. In [42], self-organizing IT2-FLSs combined with a neural network was proposed for system identification of nonlinear systems. Their proposed approach was based on the intensity of the transmitting information algorithm to determine the firing neurons. In [37], learning from scratch evolving T2-FLSs was proposed for system identification of nonlinear systems using the concept of information entropy. Their proposed work showed better accuracy compared to evolving T1-FLSs, with a fewer number of membership functions. A self-evolving recurrent T2-FLSs based on radial basis function network (RBFN) named T2FRBFN was developed by [51] for dynamic system identification. Their proposed technique illustrated better accuracy compared to a conventional RBFN. A reduced-interval type-2 neuro-fuzzy system was proposed in [169] using weighted bound-set boundaries, with online tuning capability. Their proposed algorithm was based on the Takagi-Sugeno-Kang (TSK) inference system and was verified through simulations of sequence prediction and process modeling. In [156], a function-link self-evolving type-2 neuro-fuzzy-based system was proposed for nonlinear system identification and control, where their proposed technique has the ability to construct the rule base from scratch. Their proposed technique was verified using computer simulation for timing-varying systems. A TSK type-2 self-evolving neuro-fuzzy system was proposed in [43], named as the TSCIT2FNN for system identification and noise cancellation using a compensatory operator. Their proposed TSCIT2FNN can adapt the T2-FLSs parameters and learn their structure online.

### 2.5.4 T1/T2 EFSs for Nonlinear Control Systems

This section presents a survey of type-1 evolving fuzzy systems (T1-EFSs) and type-2 evolving fuzzy systems (T2-EFSs) in control systems. One major issue of the conventional fuzzy control techniques is their fixed structure which is not able to adapt to any changes in the dynamical systems. Hence, EFSs come into the picture [144,170]. In [41], a generic self-evolving neuro-FLCs was designed for trajectory tracking of a bio-inspired UAV and also to achieve a robust control performance of a voice coil motor in [171]. These recent controllers learnt their structure from scratch and utilized the sliding mode control theory to adapt fuzzy parameters. In another study by [13], a self-evolving FLC was designed for a hypersonic vehicle with online structure and parameter learning. Their simulation results demonstrated superior performance over a fixed structure FLC. However, their proposed controller is model-based, requiring an accurate mathematical model of the system. Furthermore, most of the real-world applications are nonlinear, making it difficult to establish an accurate mathematical model, especially in the presence of external disturbances and other types of uncertainties.

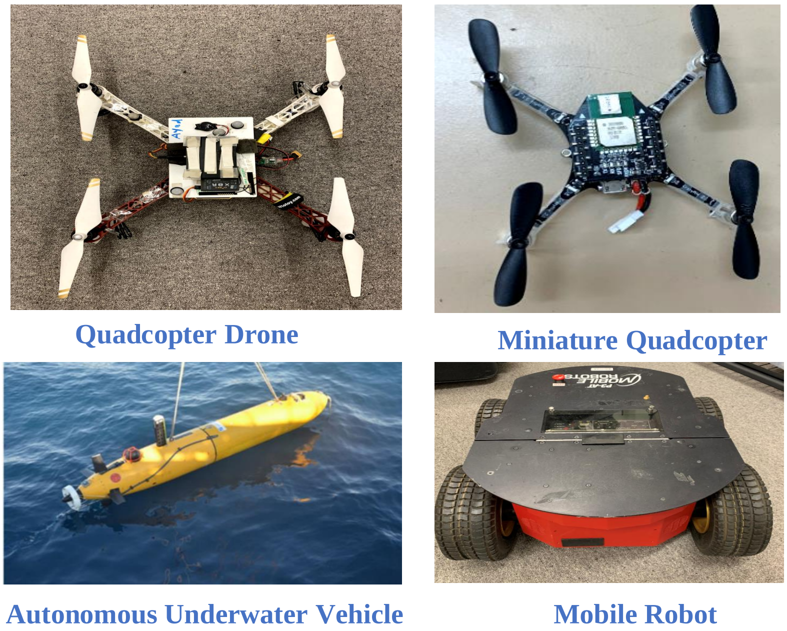
EFSs were also used for fault-tolerant control in [172], where a self-constructing FLC was proposed for aircraft fault-tolerant control. Their simulation results depicted that the trajectory of the faulty system can reach the equilibrium point within a finite time.

Nevertheless, EFSs-based T1-FLSs have limited ability to handle uncertainties. Therefore, EFSs based IT2-FLSs were proposed to capture more uncertainties in nonlinear systems, thanks to the FoU in IT2-FLSs. In the work by [106], an evolving IT2FLCs was developed to control a redundant robotic manipulator, where genetic algorithms optimization technique was utilized to optimize the scaling factors of IT2FLCs. Their proposed technique was tested in the presence of external disturbances, random noise and parameter variation resulting in robust performance. In [130, 156, 173], self-learning T2FLCs were proposed, resulting in superior performance to handle uncertainties in dynamic systems. The development of evolving IT2FLCs helps to cope with sudden variations and disturbances in nonlinear systems.

A self-organizing cerebellar model articulation controller (CMAC) for MIMO nonlinear systems was proposed in [174]. The gradient-descent method was utilized for online tuning of fuzzy parameters. The stability of their technique was guaranteed using the Lyapunov theory. An improved version of self-organizing CMAC was proposed in [175] using mixed Gaussian membership functions. Their proposed technique was deployed to control a double inverted pendulum system and a magnetic levitation plant. Nevertheless, these proposed methods were based on T1-FLCs. An improved version of the CMAC algorithm based on ITFLCs was proposed in [176], where the particle swarm optimization technique was used to optimize the control parameters. PSO-based optimization technique was also used to find the optimal learning rates in [177] to construct their proposed interval type-2 fuzzy brain emotional learning control system (IT2FBELC). However, the efficacy of their proposed approaches in the face of various uncertainties has not been investigated adequately.

## 2.6 Unmanned Autonomous Systems (UAS)

Recently, Unmanned autonomous systems (UAS) have been attracting a large amount of consideration in the last few decades. UAS are advanced, intelligent machines that have the capability to travel by air, sea or land without the need for a human personnel on-board [178]. They have been utilized in both military and civil sectors [179]. Fig. 2.11 demonstrates different types of UAS. In this section, UAS are classified into three different categories: UAVs, AUVs, and AGVs.


Quadcopter Drone

Miniature Quadcopter

Autonomous Underwater Vehicle

Mobile Robot

Figure 2.11: Nonlinear systems including aerial robots, ground robots, underwater robots (AUV image is taken from [1]).

### 2.6.1 Unmanned Aerial Vehicles (UAVs)

In recent years, Unmanned Aerial Vehicles (UAVs), known as drones, have brought several benefits in modern societies. UAVs, compared to manned aircraft, have been used in a myriad of applications (e.g. transportation [180], inspection of power transmission lines [181], search and rescue missions [182], and collecting traffic information [183]). Aerial vehicles can be categorized into four subdivisions as follows: (1) rotary-wing vehicles which include quadcopters, helicopters, octocopters, hexacopters, etc; (2) fixed-wing vehicles; (3) flapping-wing UAVs; and (4) blimps [60,184]. Among different types of UAVs, multirotor UAVs are commonly used due to their high maneuverability [185]. They are also convenient platforms due to their compact size, ease of maintenance, safety for human interaction and suitability for hazardous situations due to their high maneuverability [186], which makes them suitable platforms for research purposes. Moreover, they have the capability to hover, perform vertical take-off and landing (VTOL), and fly in confined spaces [14,60,187–191]. However, this type of aircraft is hard to model and control across its full flight envelope due to various forms of uncertainties in addition to its complex and nonlinear nature [11,192].

Although modeling multirotor UAVs from first principles has several advantages and can provide an accurate understanding of their behaviors, this approach results in computational load for physical and mathematical computations and is also time-consuming [193, 194]. Furthermore, these models usually neglect the impact of various factors such as motor dynamics, relative airspeed, motor breakdown, variations in payload and ground effect, which are not trivial to model.

One way to address this drawback is to build models from data using input-output system identification methods. Instead of using a physically relevant state vector, the state of the system can be represented by a finite number of historical inputs and outputs of the system [195]. For instance, FLSs [3, 4, 11, 196], and NNs [7, 9, 197] are two common approaches that have been successfully implemented to approximate the dynamics of multirotor UAVs. In other words, data-driven modeling approaches provide more accurate models and are computationally simpler.

From a control perspective, various techniques have been proposed for UAV flight control systems including the PID controller [198], $H_\infty$ controller [194], LQR controller [117], and other nonlinear controllers such as the SMC [199], backstepping control [200] as well as the MPC control techniques [118]. However, one shortcoming of model-based controllers is their limited ability to handle uncertainties that occur in the dynamics of many real-life systems.

On the other hand, intelligent controllers have demonstrated better performance and have been applied successfully in various nonlinear systems. In the case of multirotor UAVs, FLCs (T1FLCs and T2FLCs) and NNs have been successfully applied in multirotor control [3, 74, 201–205], with good results being achieved. Although T1FLCs have been applied to control multirotor UAVs, they have limited capability to handle uncertainties [30, 74, 90]. Therefore, T2FLCs (i.e. with interval membership functions) can be a solution to T1FLCs to achieve higher accuracy [15–17]. The use of T2FLCs in aerial robotics is useful to the FLCs community due to various reasons. Among these reasons, they can provide better control performance compared to their type-1 counterparts and are more robust to system uncertainties as demonstrated in various research papers, where better system accuracy is achieved for many real-time applications [20].

### 2.6.2 Autonomous Underwater Vehicles (AUVs)

Autonomous Underwater Vehicles (AUVs) have attracted a lot of interest in recent years as a tool to perform various underwater tasks in both civilian and military sectors. One of the main advantages of AUVs is that no human operator is required. Hence, they are capable of carrying operations in an autonomous manner [206–209]. AUVs can be utilized to perform various tasks such as pipeline surveying, underwater resource assessment, port safety defenses, and many other applications [210]. Controlling AUVs is a challenging task due to their nonlinear and time-varying dynamics. Over the past few years, several control techniques have been proposed. These include linear controls [211, 212], sliding mode control [213], LQR control [214], [215], and model predictive control [216–218]. In chapter 4, a new intelligent control system is proposed for AUVs.

### 2.6.3 Autonomous Ground Vehicles (AGVs)

The development of Autonomous Ground Vehicles (AGVs) has become one of the most fruitful research areas in the field of autonomous systems. Currently, AGVs have been used in numerous applications such as military operations [219, 220], agriculture [221], and urban logistics [222]. Various nonlinear control systems for mobile robots have been proposed. As such, numerical approaches have been developed for trajectory tracking [223, 224], navigation [225], parking [226], and obstacle avoidance [114]. In chapter 6, a novel intelligent evolving type-2 fuzzy controller is proposed for AGVs.

## 2.7 Research Gaps

From studies reviewed in this chapter, the main limitations are identified as follows:

- Most of the data-driven fuzzy system identification techniques for multirotor UAVs are offline and based on T1-FLSs, which have limited ability to model and to minimize the influence of uncertainties in nonlinear systems. To solve this limitation, IT2-FLSs based on the $C$-means clustering technique is proposed for online system identification of multirotor UAVs in the presence of disturbances in Chapter 3.
- Most of the existing IT2FLCs approaches are SISO-based controllers and commonly utilize the 'Karnik-Mendel' type-reduction algorithm, which is computationally expensive. To fill this gap, in Chapter 4, a new MIMO intelligent control system named SAF2C is developed to control nonlinear dynamic systems and to reduce the computation time of SISO controllers in the face of various uncertainties (e.g., wind gust, parameters variation, measurement noise). Also, the Enhanced Iterative Algorithm with Stop Condition type-reducer is accommodated to reduce the computation cost of the KM type-reducer. The proposed method is validated using various benchmark

dynamic systems. Moreover, the stability analysis of the proposed method is studied using the Lyapunov theory.

- The development of IT2FLCs in real-world applications as stand-alone controllers has not been addressed adequately; mainly for multirotor UAVs. To fill this gap, in chapter 5, a novel ESAF2C is developed for real-time control of a quadcopter UAV, where several flight tests were conducted to prove the efficacy of the proposed method in the face of external disturbances. Also, the stability analysis of the proposed method is investigated using the Lyapunov theory.
- Most of the T1-EFSs and T2-EFSs focused on modeling/regression problems, which are well-established in the literature. Nevertheless, there is a lack of research on using T1-EFSs and T2-EFSs in nonlinear control systems; especially in the face of various uncertainties. Therefore, in Chapter 6, to address this gap, T1-EFCs and T2-EFCs are developed to control nonlinear dynamic systems, having the ability to evolve both their structure and parameters in an online manner, and to minimize the complexity of designing FLCs.

## 2.8 Summary

In this chapter, the basic concepts of T1-FLSs, T2-FLSs, and IT2-FLSs were reviewed. Sequentially, we reviewed their applications on modeling and control of autonomous systems. Besides, a survey of the fundamental concept of EFSs and their applications on modeling and control was highlighted. Lastly, the research gaps of the existing studies were unveiled, which motivates the research work pursued in this thesis.

To address the limitation of system identification of dynamic systems, a novel online system identification technique using IT2-FLSs based on $C$-means clustering technique is proposed in Chapter 3. Besides, Chapter 4 addresses the limitation of adaptive control systems using MIMO IT2FLCs for various nonlinear dynamic systems including multirotor UAVs, AUVs, and an inverted pendulum system in the face of various uncertainties such as external disturbances, and variation of plant parameters. The limitation of real-time implementation of IT2-FLCs is addressed in Chapter 5. Lastly, Chapter 6 addresses the advantages of EFSs to control nonlinear systems using T1-FLSs, and builds an improved version of EFSs for controlling nonlinear systems using T2-FLSs.

# Chapter 3

# System Identification of Dynamic Systems Using IT2-FLSs

Parts of this chapter have been taken from the following publications:

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). Online System Identification for Nonlinear Autonomous Systems Using Recursive Interval Type-2 TS Fuzzy $C$-means Clustering. *2020 IEEE Symposium Series on Computational Intelligence (SSCI)* [91].
- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., Anavatti, S. G., and Ferdaus, M. M. (2019). Online Takagi-Sugeno Fuzzy Identification of a Quadcopter Using Experimental Input-Output Data. *2019 IEEE Symposium Series on Computational Intelligence (SSCI)* [11].

## 3.1 Introduction

Most real-world systems are highly nonlinear in nature and they cannot be fully described using linear differential equations. For example, modeling multirotor aircraft is a challenging research task due to the inherent nonlinear, complex, over-actuated or under-actuated system dynamics. As discussed in the previous chapter, data-driven modelling techniques are desirable since they have no dependency on the physical laws of nature. Rather, data-driven methods are based on the input-output data obtained from experiments. Therefore, the first part of this chapter discusses a sequential learning machine based on the Takagi-Sugeno (TS) fuzzy inference system to model the dynamics of a multicopter drone using experimental data. This is followed by proposing a novel IT2-FLS system identifier by means of the C-means clustering technique to improve the performance of a T1-FLS system identifier. This chapter leverages the advantages of nonlinear system identification, which can incorporate various uncertainties such as noise and wind gusts that occur in nonlinear dynamics of multirotor systems.

The remainder of this chapter is structured as follows. The contributions of this chapter are listed in Section 3.2. Section 3.3 discusses the experimental setup. Next, Section 3.5 presents the proposed T1-FLS system identification method to model the dynamics of multirotor aircraft. This is followed by a novel IT2-FLS system identification technique to model the dynamics of nonlinear systems in Section 3.6. Finally, a summary of this chapter is provided in Section 3.7.

## 3.2 Contributions of this chapter

Inspired by the previous work in fuzzy system-based data-driven modeling, a novel online system identification approach by means of a recursive interval type-2 TS fuzzy C-means clustering (IT2-TS-FC) method is developed for modeling of a multirotor UAV is described in Section 3.3. The data is collected from real-time flight tests. The designed approach deploys the Lagrange method to minimize the objective function (the modeling error and the membership functions). Unlike previous IT2-FLSs models, this method constructs two fuzzifiers (upper and lower) and two regression coefficients in the consequent part to handle uncertainties. Besides, the weighted least square method to compute the regression coefficients is deployed. Moreover, the proposed approach is validated using two set of data, namely, the real flight test of a quadcopter drone and the Mackey-Glass time series data

[227]. Lastly, to investigate the robustness of this approach, an online system identification using a noisy dataset is conducted, providing results that demonstrate the capability of this method to capture uncertainties in nonlinear systems.

## 3.3 Design of Experimental Setup

This section describes the experimental setup, including the commercial-off-the-shelf (CoTS) autopilot and air-frame. A brief description of the robotic operating system (ROS) implementation used on the UAV is provided. Finally, an explanation of integration of the drone with the VICON optical motion capture system and ground control station (GCS) is presented.

### 3.3.1 Pixhawk Autopilot

Autopilots can be defined as systems that combine both hardware and software, making them capable to guide autonomous systems during operations [228]. Autopilots can perform fully autonomous missions replacing a human-in-the-loop, or provide partial autonomy via remote control commands. To achieve these missions, autopilot boards usually consist of power management modules, inertial measurement units (IMUs), peripheral sensors, communication links, on-board computers etc. The on-board software includes mission planning, control, and trajectory generation [228–230]. The typical avionics diagram implemented for multirotor aircraft, a quadcopter in this research study, can be illustrated in Fig. 3.1. There are different types of autopilots in the literature. In this study, the Pixhawk autopilot hardware is accommodated.

There are several flight controllers including Ardupilot [231] and PX4 [232]. They are open-source and all-in-one autopilot systems developed by engineers that have the capability to control aircraft, unmanned ground vehicles and many other autonomous platforms [233]. In this study, the PX4 autopilot system is utilized.

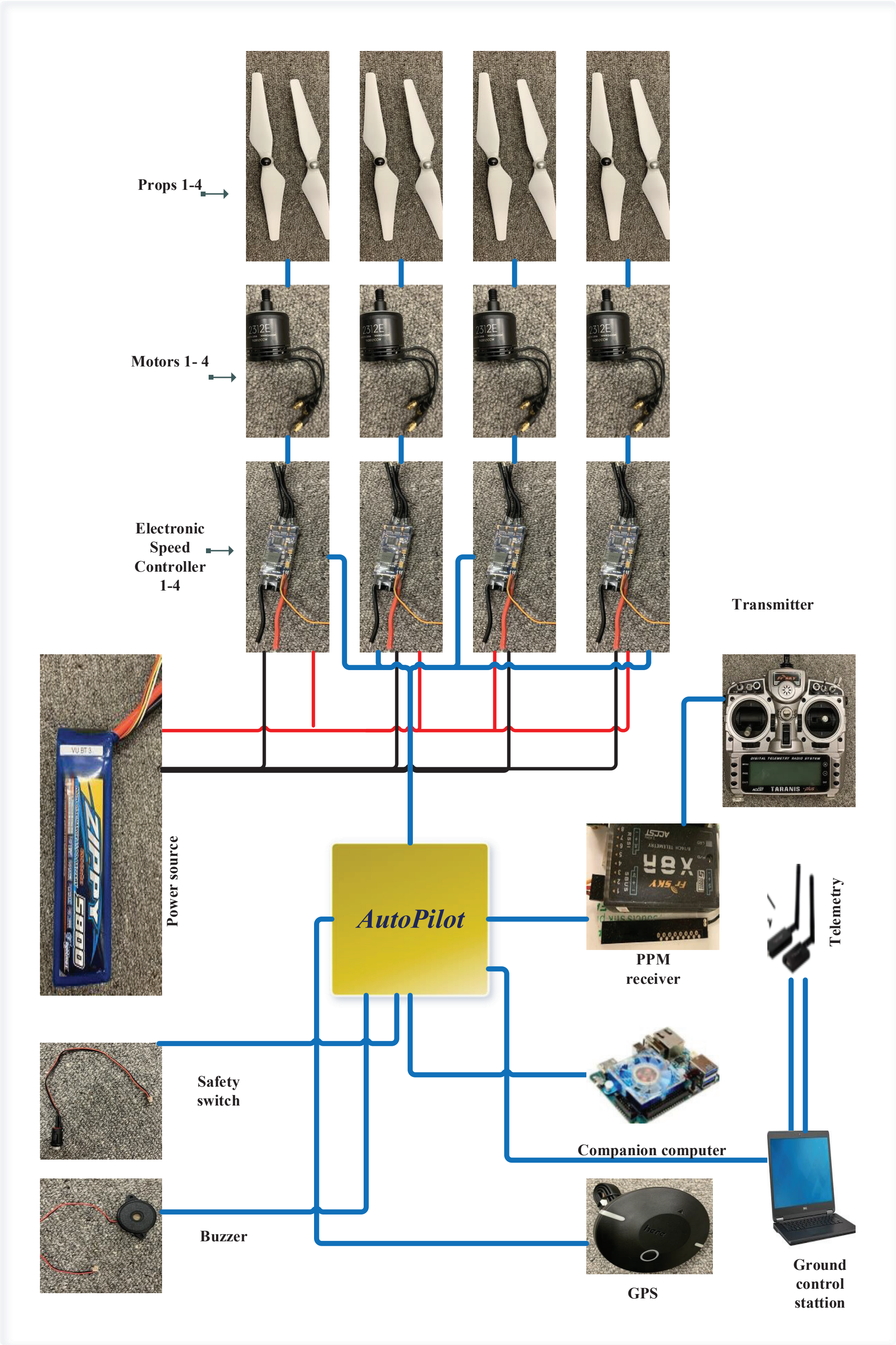
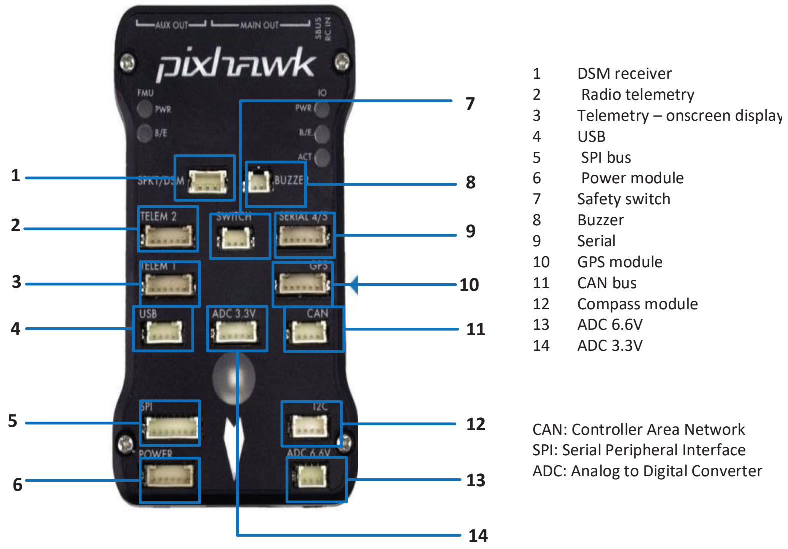


Figure 3.1: Common on-board avionics system on multirotor UAVs (in this case: a quadrotor).

Fig. 3.2 shows the top view and interfaces available on the Pixhawk autopilot hardware. The key features of the Pixhawk autopilot hardware are tabulated in Table. 3.1. For more details, readers can refer to the documentations in [228, 233].



| | |
|---|---|
| 1 | DSM receiver |
| 2 | Radio telemetry |
| 3 | Telemetry – onscreen display |
| 4 | USB |
| 5 | SPI bus |
| 6 | Power module |
| 7 | Safety switch |
| 8 | Buzzer |
| 9 | Serial |
| 10 | GPS module |
| 11 | CAN bus |
| 12 | Compass module |
| 13 | ADC 6.6V |
| 14 | ADC 3.3V |

CAN: Controller Area Network
SPI: Serial Peripheral Interface
ADC: Analog to Digital Converter

Figure 3.2: Pixhawk Autopilot top view.

| **Features** | **Pixhawk** |
|---|---|
| **Support of various UAVs** | Helicopter, Multirotor and Airplanes |
| **Memory** | 256KB RAM - 2 MB Flash |
| **Processors** | 32bit - STEM32F427-<br>CortexM4-168MHz |
| **Control modes** | Manual, Auto, Assistant |
| **OS** | Nuttx |
| **Interfaces** | CAN, 12C, PWM, ADC, SPI |
| **Redundancy** | IMU |
| **IMU** | 2x 6DOF inertials, 1x Magnetometer |
| **Sensors** | ST Micro L3GD20H 16 bit gyro<br>LSM303D 14 bit<br>accelerometer/magnetometer<br>Invensense MPU 6000 3-axis<br>accelerometer/gyroscope<br>MEAS MS5611 barometer |

Table 3.1: Pixhawk features

### 3.3.2 VICON Motion Capture System (MCS)

The Vicon *Tracker* software is a robust tool for tracking objects, that provide unrivaled data accuracy to be integrated into 3-D applications. It provides users with the capability to utilize Vicon camera hardware to track rigid bodies resulting in accurate streaming of 6-DOF data in real-time with minimal latency [234].

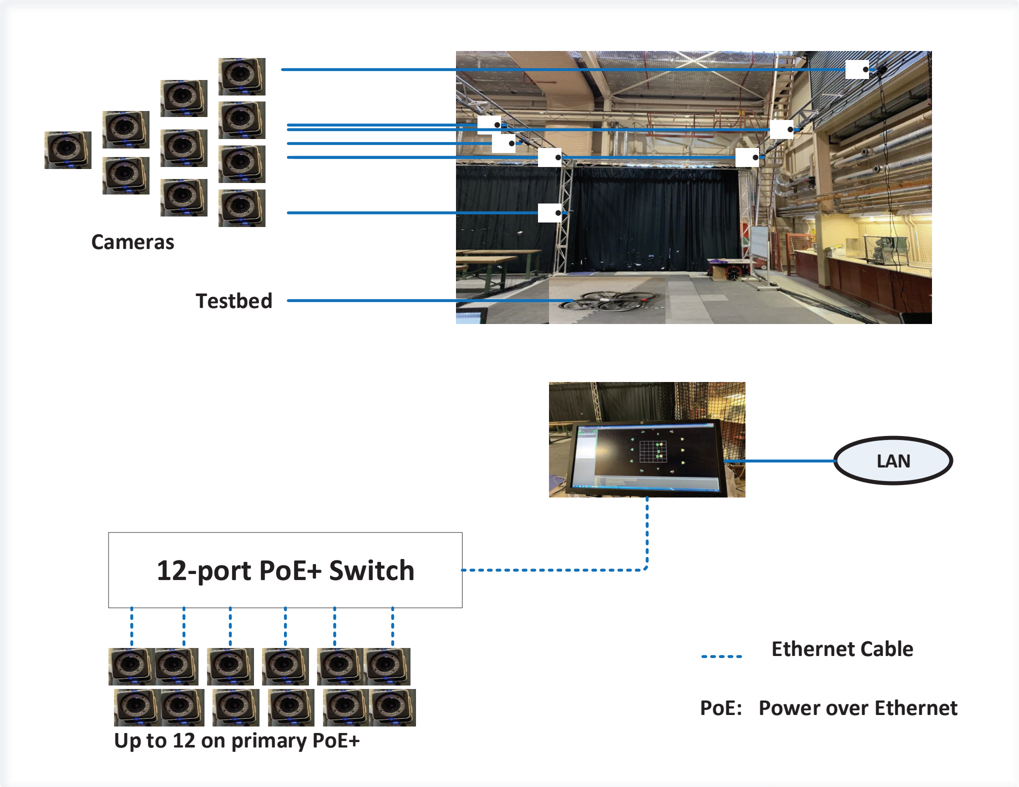
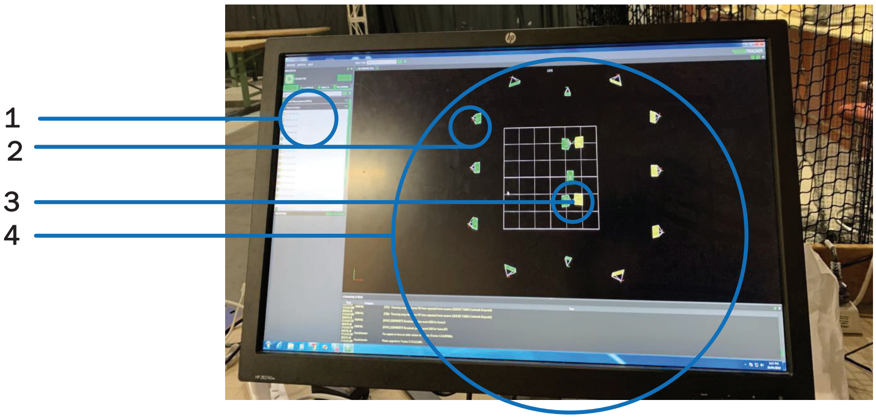


Figure 3.3: Indoor test area using VICON optical motion capture system.

Fig. 3.3 illustrates the indoor test area equipped with 19 Vicon cameras at the UNSW Canberra Autonomous Systems laboratory. This facility is used for experiments to capture motions of various autonomous vehicles for closed loop control and data logging purposes. The Vicon optical capture system utilizes a series of Near Infrared (NIR) strobe units attached to the front of each Vicon camera for test area illumination. Also, markers are used and placed onto objects being tracked in a unique pattern. The markers are captured by the Tracker software based on their reflections of the NIR strobe. The interface with objects in 3D using the Vicon tracker software is shown in Fig. 3.4.



| | |
|---|---|
| 1 | Name of objects in VICON tracker |
| 2 | Cameras in real-time VICON tracker |
| 3 | The interface with objects in 3D view |
| 4 | GUI of test area |

Figure 3.4: Interface with objects in 3D plane using VICON tracker software.

### 3.3.3 Companion Computers

In this work, the ODroid-XU4 companion computer is utilized to interface and communicate with the PX4-autopilot during operation via the micro air vehicle link (MAVLink) protocol over a serial connection as illustrated in Fig. 3.5. MAVLink is a serial protocol that is widely to communicate between unmanned systems and GCS [235].

Moreover, the GCS used in this work is the QGroundcontrol, which provides complete flight control and mission planning capabilities for any drone equipped with MAVLink messaging protocol. This enables users to update firmware, calibrate flight sensors and the radio control system, set and tune flight control parameters, as well as plan and track flight trajectories [236].

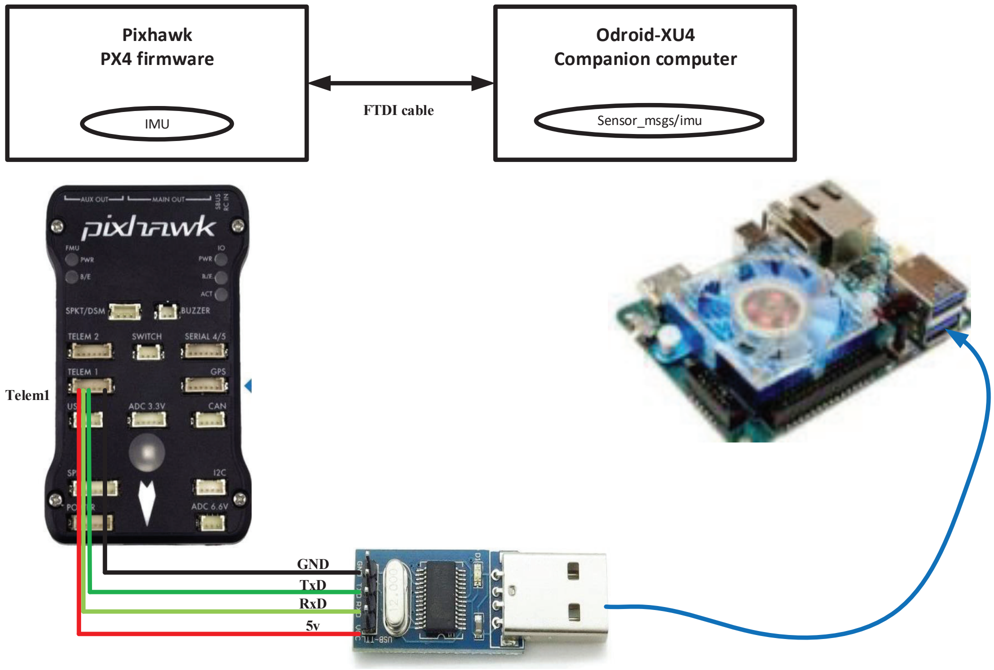


Figure 3.5: Communication between Onboard computer ODroid-XU4 and Pixhawak autopilot via MAVLink.

### 3.3.4 Robotic Operating System (ROS)

The Robot Operating System (ROS) is an open-source framework, that has been widely used by researchers in the robotics community [237]. It was initially developed by the Stanford Artificial Intelligence Laboratory in 2007. ROS provides services as a middleware system (a collection of software frameworks for robot software development) and is packed with various useful functionalities that include package management, hardware abstraction, distributed computing, message passing and low-level device control [237,238]. Interested readers may refer to [237,238] for detailed information about ROS architecture.

The communication between the companion computers running ROS, GCS or peripherals is implemented using the MAVROS package for any MAVLink enabled autopilots. In other words, MAVROS is a ROS package that facilitates communication between the companion computers (e.g., ODROID-XU4) and the PX4 flight software using the MAVLink protocol. The detailed installation guide of MAVROS can be found at [232].

### 3.3.5 Airframe

The aircraft used for this work was the DJI F450 quadrotor. As with rotorcraft vehicles, the system has the ability to hover and to perform vertical take-off and landing tasks. Quadcopters have various configurations namely: quad X, quad +, quad H and quad V configurations. The most common configurations are the plus (+) and cross (X) configurations. Fig. 3.6 represents an X-configuration used in this research study. Quadrotors have six degrees of freedom (DOF), where six variables are considered. As quadcopters have four control variables, the system is under-actuated as there are only four variables to be directly controlled, that is, the pitch, roll, and yaw angles as well as the altitude to deal with their 6-DoF. Meanwhile, the $x$ and $y$ coordinate positions can only be indirectly controlled through the pitch and roll control loops. The schematic diagram of a quadcopter is shown in Fig. 3.6, where rotors 1 & 3 rotate clockwise, and rotors 2 & 4 rotate anticlockwise. The control input is achieved by changing the speed of each rotor using electronics speed controllers (ESCs) coupled to the DC motors. By applying the

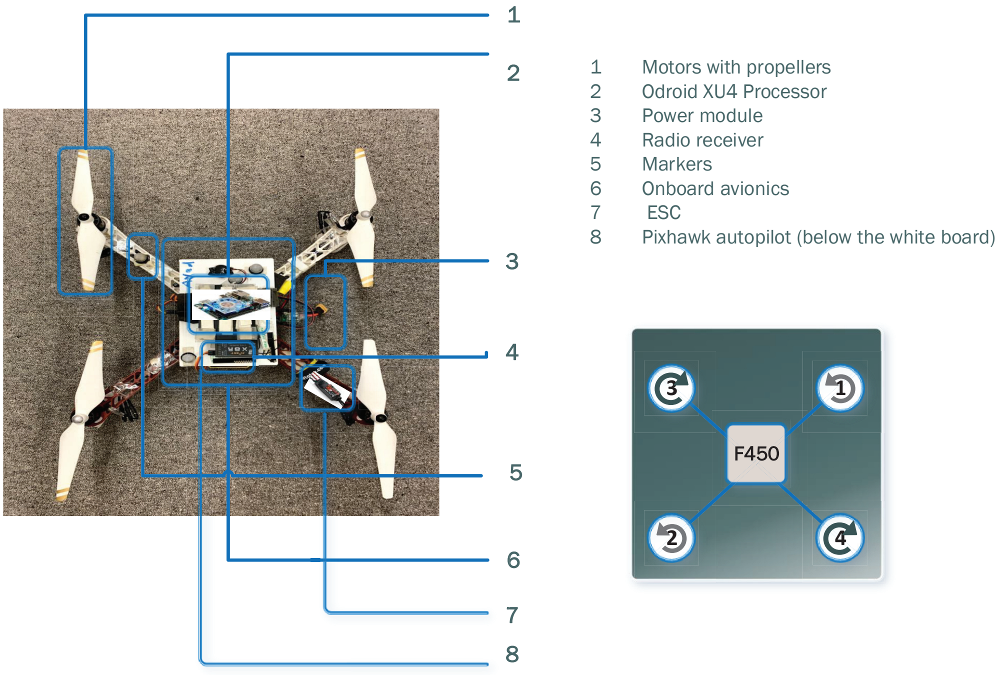


Figure 3.6: The configuration of DJI F450 quadrotor (X)-configuration.

same amount of thrust to the four individual motors, the drone is set to hover. Pitching and rolling are achieved by adjusting the amount of thrust given to one rotor and less to the diametrically opposite rotor.

An industrial fan with three different speed modes was utilized for wind gust generation during flight tests as shown in Fig. 3.7.

## 3.4 Performance Indices

The performance indices used in this research are the Root Mean Square Error (RMSE), Mean Square Error (MSE) and the Non-Dimensional Error Index (NDEI), which are described in (3.1), (3.2) and (3.3).

$$\mathbf{RMSE} = \sqrt{\frac{\sum_{k=1}^{N} \left(y_{actual} - \hat{y}_k\right)^2}{N}} \tag{3.1}$$

Figure 3.7: Industrial fan to generate wind gust disturbance.

$$\textbf{MSE} = \frac{1}{N}\sum_{k=1}^{N}\left(y_{actual} - \hat{y}_k\right)^2 \tag{3.2}$$

$$\textbf{NDEI} = \frac{\text{RMSE}}{STD(T)} \tag{3.3}$$

where $y_{actual}$ is the actual output of the system; $\hat{y}_k$ represents the estimated fuzzy output; $N$ represents the number of observations; and $STD(T)$ denotes a standard deviation function.

## 3.5 Sequential Learning Machine Based on TS Fuzzy System Identification

To model a nonlinear system like a quadrotor, there are several existing data-driven techniques. Among these data-driven techniques, FLSs and NNs have been employed successfully in recent times due to their universal approximation and learning capability. Besides, the use of artificial intelligence such as FLSs provides an advantage, where it is not possible to derive comprehensive mathematical models of the system. [13, 14]. In this section, a sequential learning machine based on the TS-fuzzy inference system to model the dynamics of a MIMO nonlinear quadcopter using experimental data are presented. Unlike conventional knowledge-based TS-fuzzy systems, all the antecedent and consequent parameters of the proposed TS-fuzzy model are updated using the gradient descent-based algorithm.

### 3.5.1 TS Fuzzy Online System Identifier

#### 3.5.1.1 Problem Statement

In the following, a nonlinear system identification is described using the relation between the inputs and outputs as follows:

$$\hat{y}(k) = f\left(u(k), ..., u(k-n+1), \quad y(k), y(k-1), ..., y(k-m+1)\right), \tag{3.4}$$

where $\hat{y}(.)$ indicates the identified model output; $f(.)$ represents the unknown nonlinear function; $n$ and $m$ denote the previous input $u$ and output $y$ data, respectively.

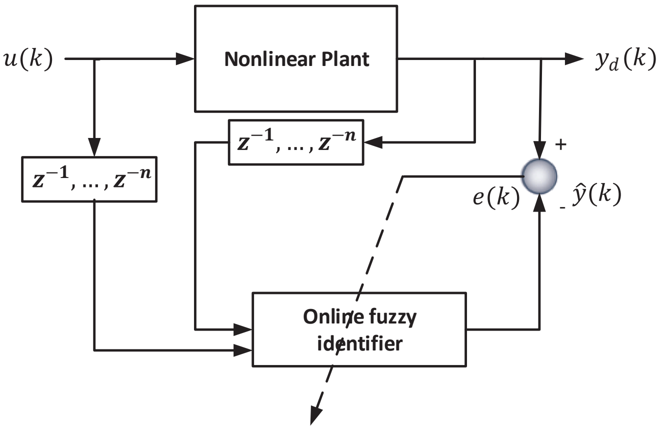


Figure 3.8: Block diagram using series-parallel architecture. The error is calculated as the difference between the plant outputs and the output of the identified fuzzy model, that is, $e = \hat{y} - y$.

Following the work in [239], identification of dynamic systems can be obtained using online and offline techniques to describe the relationship between the input and output data. If the identified model is entirely trained before its use, the model becomes offline. However, if the model is progressively trained during its use, it becomes online. Online system identification approaches are described as adaptive models, where the parameters are adapted at every sample time [239]. Moreover, this technique is suitable when the system parameters keep changing continuously with time. In this work, the online system identification technique is adopted.

The proposed dynamic fuzzy model for the multi-copter UAV is constructed by adopting the MIMO structure in Eq. (3.5) to Eq. (3.8).

$$y_1(k) = f_1\left(u_1(k), u_2(k), u_3(k), u_4(k)\right) \tag{3.5}$$

$$y_2(k) = f_2\left(u_1(k), u_2(k), u_3(k), u_4(k)\right) \tag{3.6}$$

$$y_3(k) = f_3\left(u_1(k), u_2(k), u_3(k), u_4(k)\right) \tag{3.7}$$

$$y_4(k) = f_4\left(u_1(k), u_2(k), u_3(k), u_4(k)\right) \tag{3.8}$$

where $u_1$, $u_2$, $u_3$, and $u_4$ represent the four inputs; and $y_1$, $y_2$, $y_3$, and $y_4$ are the outputs of the MIMO fuzzy system.

There are two main techniques of fuzzy system identification, namely, the series-parallel and the parallel architectures. The difference between the two architectures is that the parallel structure applies the history of outputs from the identified model, while the series-parallel structure, as in Fig. 3.8, utilizes the plant true output as the input to the fuzzy model resulting in a better accuracy since the prediction error is not accumulated or propagated by the system [240].

### 3.5.2 System Structure

As a recap from Chapter 2, the structure of T1-FLSs consists of four steps: (1) fuzzification; (2) knowledge base; (3) inference system; (4) defuzzification. Knowledge base layer is made of a set of rules, which consists of *antecedents* and *consequents*. The Gaussian MFs are utilized in the fuzzification process as shown in Fig. 3.9. In this work, all the

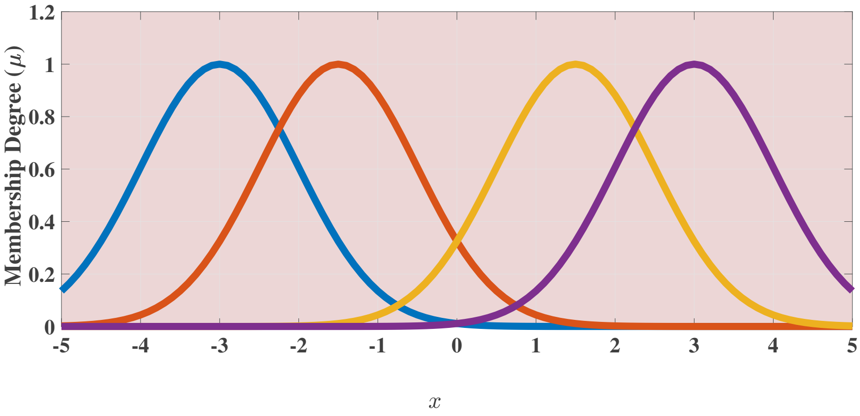


Figure 3.9: Four T1-FLSs Gaussian MFs.

*antecedents* and *consequents* parameters of the FLS are updated using the descent-based back-propagation method. A T1 Gaussian MF is used to perform the fuzzification process for the *antecedents*, which can be represented as follows:

$$\mu_{A_i}(x_j, c_{ij}, \sigma_{ij}) = exp\left(-\frac{((x_j - c_{ij})^2}{2\sigma_{ij}^2}\right) \tag{3.9}$$

where $c_{ij}$ and $\sigma_{ij}$ denote the centre and the width of the membership function and $x_j$is the $i^{th}$input variable. This work uses the Gaussian membership function to map the inputs, while using the delta function, which is an impulse function of zero width that has only one value with full membership located at $q_i$ to map the outputs. Following the work in [85, 92], the final estimated output of a first-order TS fuzzy model can be represented as follows:

$$\hat{y} = \sum_{i=1}^{R} \gamma_i \left(q_i^0 + q_i^1 x_1 + ... + q_i^k x_k\right) \tag{3.10}$$

where

$$\gamma_i = \frac{\prod_{j=1}^{P} exp\left(-\frac{((x_j - c_{ij})^2}{2\sigma_{ij}^2}\right)}{\sum_{i=1}^{R}\prod_{j=1}^{P}} \tag{3.11}$$

For Eq. (3.11), $R$ represents the number of rules in the rule-base; and $p$ represents the number of inputs per data-tuple.

### 3.5.3 Parameters Learning

The performance index function of the training process to be minimized can be written as follows:

$$J = \frac{1}{2}\left(y(t) - \hat{y}(t)\right)^2 \tag{3.12}$$

The error between the estimated and the actual value can be written as follows:

$$\mathbf{e} = y - \hat{y}, \tag{3.13}$$

where $y$ and $\hat{y}$ illustrate the actual and the estimated signals, respectively.

The update law for consequent weights $q_i$ is given as follows [85, 92]:

$$q_i(t+1) = q_i + \eta\gamma_i e, i = 1, 2, .., R. \tag{3.14}$$

Similarly, the antecedents update law is given as follows:

$$c_{ij}(t+1) = c_{ij}(t) + 2\eta\gamma_i e\,(q_i - \hat{y})\,\frac{x_j - c_{ij}(t)}{\sigma_{ij}^2} \tag{3.15}$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + 2\eta\gamma_i e\,(q_i - \hat{y})\,\frac{(x_j - c_{ij})^2}{\sigma_{ij}^3}, \tag{3.16}$$

where $\eta \in [0,1]$ is the learning rate.

### 3.5.4 Data normalization

Utilizing the Min-Max normalization method, the data are normalized between '0' and '1' as follows:

$$z_j = \left(\frac{x_j - x_{min}}{x_{max} - x_{min}}\right) \tag{3.17}$$

where $z_j$ is the $j^{th}$ new normalized values; $x_j$ represents the current values.

## 3.6 A Novel System Identification Based on the Recursive IT2-FLS TS C-means Clustering Technique

As a recap, fuzzy system identification techniques have been extensively used in various nonlinear systems to obtain accurate models using input-output data [3,11,85,90]. One of the most popular fuzzy modeling approaches is the Takagi-Sugeno (TS) fuzzy model. The main task of the TS fuzzy model is to build several local models that can approximate the dynamics of a nonlinear system [90]. The construction of the TS fuzzy model consists of two phases: fuzzy structure identification and fuzzy parameter identification. The most significant process to establish a fuzzy model is the structure identification that is concerned with the selection of fuzzy inputs, the number of rules, and the membership functions [93].

Modeling of autonomous systems such as multi-rotor drones has several challenges due to their complex, nonlinear, and under-actuated system dynamics [11, 85]. As universal approximators, artificial intelligent systems such as fuzzy logic [11,85,241,242] have proved to be successful computational tools to model nonlinear systems [2, 241].

Although type-1 fuzzy systems have been applied to model various complex nonlinear systems, they have limited capability to handle uncertainties [30,74,90]. Therefore, type-2 fuzzy systems (T2FS) (i.e. with interval membership functions) were established [15–17, 243]. Among the most common methods for fuzzy structure identification is the fuzzy clustering technique. There are various fuzzy clustering methods in the literature, such as the Fuzzy C-means (FCM) by Bezdek [94], modified Gath-Geva fuzzy clustering [95] and fuzzy C-regression model (FCRM) [96]. Since T1-FLSs based clustering techniques have limited ability to handle uncertainty in real systems [97], several T2-FLSs/IT2-FLSs based C-means clustering techniques were proposed [90, 99–101].

In this section, a novel online system identification technique based on a recursive interval type-2 Takagi-Sugeno fuzzy C-means clustering technique (IT2-TS-FC) is presented. The construction of the IT2-TS-FC antecedent parameters is based on the interval type-2 fuzzy C-means clustering (IT2FCM) technique, while the weighted least square (WLS) algorithm is utilized to determine the upper and lower fuzzy consequent parameters. Moreover, a scaling factor to represent the $FoU$ is introduced to convert T1-FSs and T2-FSs. The efficacy of the proposed algorithm is validated using two benchmark system datasets, namely, the flight test data from a quadcopter and the Mackey-Glass time series data. Besides, a comparison with a type-1 fuzzy C-means technique is conducted. The robustness of the proposed method is investigated by means of a noisy dataset.

### 3.6.1 IT2-TS-FC Structure Identification

Fuzzy clustering is a well-known technique for fuzzy space partition. It is also a popular method for the identification of fuzzy antecedent parameters [90, 101]. Assume a data set $D = \{(\mathrm{x}_1, y_1), ..., (\mathrm{x}_k, y_k)\}$, with a $k$ number of input-output patterns, in which an $i^{th}$

fuzzy regression model is extracted. The data pairs $(k = 1, ..., n)$ are sorted in $c$ cluster, whereas the data samples in the $i^{th}$ cluster can be represented as a linear regression model as follows:

$$\begin{aligned} y_k^i = f^i(x_k, \theta^i) &= {\theta_0}^i + \theta_1^i x_{k1} + ... + \theta_M^i x_{kM} \\ &= [x_k \ 1]{\theta^i}^T, \end{aligned} \tag{3.18}$$

where $\mathrm{x}_k = [x_{k1}, ..., x_{kn}]$ represents the $k^{th}$ input variables; $y_k$ describes the $i^{th}$ regression model of the fuzzy output; and $\theta^{\mathrm{i}} = [\theta_1^i, ..., \theta_n^i]$ are the coefficients vector of the linear regression model. The modeling error can be computed by taking the difference between the actual signal $y_k$ and the prediction $y_k^i$, that is,

$$E^2(\theta^i) = \left(y_k - f^i(x_k, \theta^i)\right)^2. \tag{3.19}$$

The objective function of IT2-TS-FC is to minimize the modeling errors. In type-2 fuzzy modeling, two fuzzifiers (or can be called weighting exponents of IT2-FSs) are required, which brings up two objective functions, which can be defined as follows:

$$\begin{cases} J_{m_1}(U, \theta) & = & min \sum_{k=1}^{n} \sum_{i=}^{C} (\mu_{ik})^{m_1} E_{ik}(\theta^i) \\ J_{m_2}(U, \theta) & = & min \sum_{k=1}^{n} \sum_{i=}^{C} (\mu_{ik})^{m_2} E_{ik}(\theta^i), \end{cases} \tag{3.20}$$

where $\mu_{ik} = [\overline{\mu}_{ik}, \ \underline{\mu}_{ik}] \in [0, 1]$ is the fuzzy membership degree of $k^{th}$ data belong to the $i^{th}$ cluster [244]. This is subjected to the following constraint:

$$\sum_{i=1}^{C} \mu_{ik} = 1, \forall k = 1, ..., n. \tag{3.21}$$

To minimize the objective function, the Lagrange method is implemented for obtaining the upper and the lower membership degrees, $\overline{\mu}_{ik}$ and $\underline{\mu}_{ik}$, as follows [245]:

$$\overline{\mu}_{ik} = \begin{cases} \dfrac{1}{\sum_{m=1}^{C} \left[\frac{E_{ik}(\theta^i)}{E_{mk}(\theta^i)}\right]^{\frac{1}{m_1 - 1}}} & for \ \dfrac{1}{\sum_{m=1}^{C} \frac{E_{ik}(\theta^i)}{E_{mk}(\theta^i)}} < \frac{1}{C} \\ \dfrac{1}{\sum_{m=1}^{C} \left[\frac{E_{ik}(\theta^i)}{E_{mk}(\theta^i)}\right]^{\frac{1}{m_2 - 1}}} & otherwise \end{cases} \tag{3.22}$$

$$\underline{\mu}_{ik} = \begin{cases} \dfrac{1}{\sum_{m=1}^{C}\left[\frac{E_{ik}(\theta^i)}{E_{mk}(\theta^i)}\right]^{\frac{1}{m_1-1}}} & for \ \dfrac{1}{\sum_{m=1}^{C}\frac{E_{ik}(\theta^i)}{E_{mk}(\theta^i)}} \geq \frac{1}{C} \\ \dfrac{1}{\sum_{m=1}^{C}\left[\frac{E_{ik}(\theta^i)}{E_{mk}(\theta^i)}\right]^{\frac{1}{m_2-1}}} & otherwise \end{cases} \tag{3.23}$$

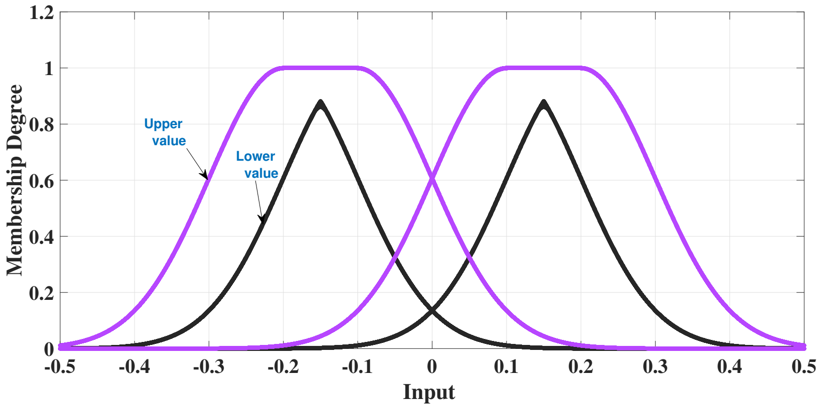


Figure 3.10: IT2-FLSs Gaussian MF used for the proposed design

The WLS approach is utilized to calculate the regression coefficients of the upper and lower $\theta^i$, where $X = [x_k \ 1]_{n\times(M+1)}$, $y = [y_k]_{n\times 1}$, as follows:

$$\overline{\theta}^i = \left[X^T\overline{P}_iX\right]^{-1}X^T\overline{P}_iy \tag{3.24}$$

$$\underline{\theta}^i = \left[X^T\underline{P}_iX\right]^{-1}X^T\underline{P}_iy \tag{3.25}$$

The membership degree of the partition matrix $\overline{P}_i$ and $\underline{P}_i$ are utilized as weights and can be represented as follows:

$$\overline{P}_i = \begin{bmatrix} \overline{\mu_{i1}} & 0 & \cdots & 0 \\ 0 & \overline{\mu_{i2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \overline{\mu_{in}} \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{3.26}$$

$$\underline{P}_i = \begin{bmatrix} \underline{\mu_{i1}} & 0 & \cdots & 0 \\ 0 & \underline{\mu_{i2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \underline{\mu_{in}} \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{3.27}$$

In order to reduce the complexity of the framework, the regression coefficient $\theta_i$ can be obtained as follows:

$$\theta^i = \frac{\left(\overline{\theta}^i + \underline{\theta}^i\right)}{2}. \tag{3.28}$$

### 3.6.2 IT2-TS-FC Parameters Identification

#### 3.6.2.1 Antecedent Parameters Identification

Type-2 Gaussian membership functions are adopted in this research study as shown in Fig. 3.10. The upper and lower centers and the widths of the Gaussian functions can be computed as follows:

$$\overline{c}_{ij} = \frac{\sum_{k=1}^{n} \overline{\mu}_{ik} x_{kj}}{\sum_{k=1}^{n} \overline{\mu}_{ik}} \tag{3.29}$$

$$\underline{c}_{ij} = \delta_1 \overline{c}_{ij} \tag{3.30}$$

where $i = 1, ..., c$ represents the number of clusters, $j = 1, ..., M$ denotes the number of variables, $k = 1, ..., n$ represents the number of data samples and $\delta_1$ is a scaling factor.

$$\overline{\sigma}_{ij} = \sqrt{\frac{2 \sum_{k=1}^{n} \overline{\mu}_{ik} (x_{kj} - \overline{c}_{ij})^2}{\sum_{k=1}^{n} \overline{\mu}_{ik}}}. \tag{3.31}$$

$$\underline{\sigma}_{ij} = \delta_2 \overline{\sigma}_{ij}, \tag{3.32}$$

where $\delta_2$ is a scaling factor to represent *footprint of uncertainties* (FoUs). The lower and upper membership functions are calculated based on Chapter 2.

#### 3.6.2.2 Consequent Parameters Identification

After identifying the antecedent parameters, the consequent parameters can be obtained using the following equation:

$$\hat{y} = \varXi\theta, \tag{3.33}$$

where $\theta = [\theta_0^1, \theta_M^1, ..., \theta_0^C, ..., \theta_M^C]$ is the consequent parameters, $y = [y_1, y_2, ..., y_m]$ is the target output and $\varXi \in \mathbb{R}^{n\times(c\times(M+1))}$ denotes the coefficient matrix, as follows:

$$\Xi = \left[\begin{array}{cccccccc} \chi_1^1 & \chi_1^1 x_{11} & \cdots & \chi_1^1 x_{1N} & \cdots & \chi_1^c & \cdots & \chi_1^c x_{1N} \\ \chi_2^1 & \chi_2^1 x_{21} & \cdots & \chi_2^1 x_{2N} & \cdots & \chi_2^c & \cdots & \chi_2^c x_{2N} \\ & & & & \vdots & & & \\ \chi_M^1 & \chi_M^1 x_{M1} & \cdots & \chi_M^1 x_{MN} & \chi_M^c & \cdots & \chi_M^c & \chi_M^c x_{MN} \end{array}\right] \tag{3.34}$$

The $\chi_{ik}$ is the normalized weight of MFs, which has the following expression,

$$\chi_{ik} = \frac{w^{ik}}{\sum\limits_{i=1}^{C} w^{ik}}, \tag{3.35}$$

where $w^{ik}$ is the firing strength average of the upper and the lower MF, which can be calculated based on Chapter 2. If an upper Gaussian MF is represented by $\overline{A}_j^i$, and a lower Gaussian MF is represented by $\underline{A}_j^i$, the upper and the lower firing strengths can be represented by $\overline{w}^{ik} = min\{\overline{A}_j^i\}$, $\underline{w}^{ik} = min\{\underline{A}_j^i\}$. Hence, $w^{ik} = \frac{\overline{w}^{ik}+\underline{w}^{ik}}{2}$.

### 3.6.3 Results and Discussion

The proposed IT2-TS-FC is implemented for input-output data partition $[x(t),\ y(t)]$ into $c$ hyper planed clusters. After generating the upper and lower membership functions, IT2-FSs are transformed into T1-FSs. The consequent parameters are obtained using the WLS method. Moreover, different levels of *FOUs* are performed to study the robustness of the identified model against uncertainties. The performance of the proposed system was evaluated under three different scaling factor that represents the *FoUs* at 25%, 50% and 75% as demonstrated in Table 3.2. The identified model is validated using two benchmark systems. First, a Macky-Glass chaotic system is modeled using the proposed method. Second, the proposed approach is deployed to model the aircraft dynamics.

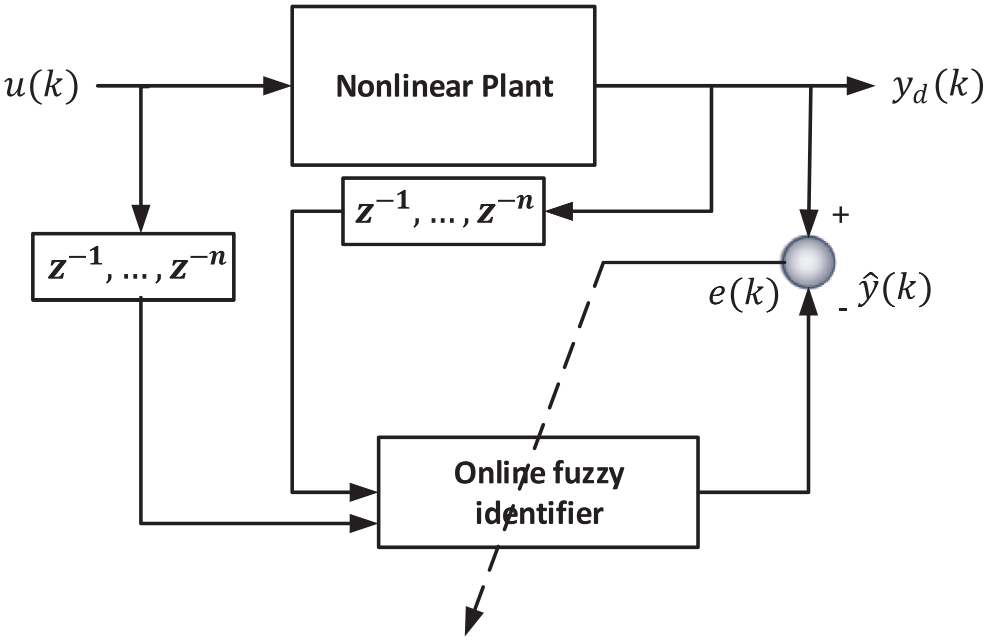


Figure 3.11: Online recursive IT2-TS-FC system identification block diagram. The system employs the prediction error to constantly learn the dynamics of the systems.

#### 3.6.3.1 Online Identification of Mackey-Glass (MG) Chaotic System

The Mackey-Glass (MG) time series is labeled by the delayed differential equation [227]. It is recognized as a benchmark model for comparing the learning ability of various algorithms. The MG chaotic time series can be represented as follows:

$$\dot{x}(t) = \frac{0.2(t-20)}{1+x^{10}(t-20)} - 0.1x(t). \tag{3.36}$$

In this work, the initial condition is set as $x(0) = 1.2$. The sampling time $t_s = 1\ sec$ and $x(t-\tau) = 0\ for\ t < \tau$, where $\tau = 20$. In general, the standard prediction technique is to create a mapping from sample data points space $D$ as follows:

$$\begin{cases} D &= [x(t) \quad y(t)],\ t = 1, ..., 1200 \\ x(t) &= [x(t-3)x(t-2)x(t-1)x(t)] \\ y(t) &= x(t+1) \end{cases} \tag{3.37}$$

In other words, four consecutive known values were utilized to predict the next time-series value. The performance of the IT2-TS-FC model is shown in Fig. 3.12. The NDEI and

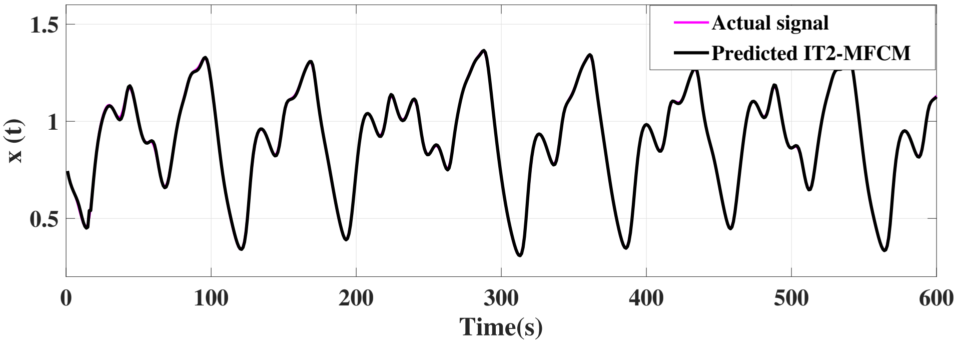


Figure 3.12: Mackey-Glass Chaotic Time Series identified model

RMSE values are listed in Tables 3.2 & 3.3, where better accuracy is obtained compared to a T1-FLSs based C-means clustering model.

#### 3.6.3.2 Identification of the Attitude and Altitude Dynamics

Data are collected during the flight using an indoor VICON motion capture system (MCS). The MCS transmits the aircraft position and orientation in real-time to a GCS using WIFI, where all data are being recorded in the drone. Control inputs are also transmitted to the GCS from the aircraft. To collect data, the drone was flown in manual mode, where the system was excited sufficiently in order to capture drone dynamics. Six reflective markers were attached to the DJI F450 quadrotor to enable the VICON MCS to track the quadrotor's position and orientation. The data was collected with a sampling rate of 100 $Hz$. The experimental setup is shown in Fig. 3.13.

In this section, 9000 samples for online identification of the pitch, roll, yaw, and altitude motions are utilized. To study the robustness of the proposed technique, an artificial white Gaussian noise was added to the measurement data. Moreover, the dataset was normalized between the range $[0, 1]$ as illustrated in the graphs. The performance of the identified type-2 fuzzy model is compared with a type-1 fuzzy model that was designed in the previous section. Also, the model was bench-marked against a type-1 fuzzy based c-means clustering model. In the proposed model, five clusters were used to perform the system identification. The centres, $[m_1,\ m_2]$, were selected as $[1.8,\ 2.2]$, respectively. These

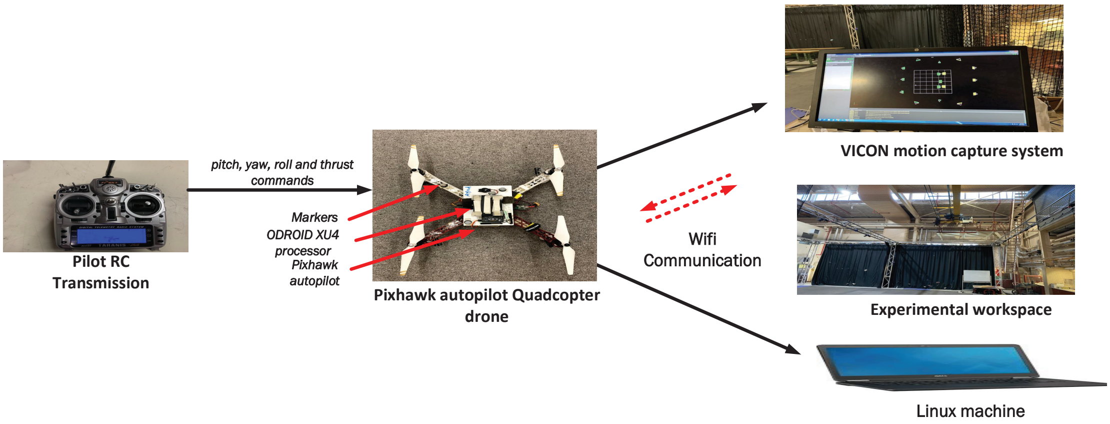


Figure 3.13: Experimental setup for data acquisition of a quadcopter drone. Our indoor flight test facility is equipped with 20 VICON motion capture cameras with millimetre accuracy.

parameters were selected by performing several experiments on the NDEI and RMSE values. The performance of the system was assessed at different scaling factor as shown in Table 3.2.

The NDEI and RMSE values were recorded as shown in Tables 3.2 & 3.3 for the normalized dataset. Also, Figs 3.19-3.23 provides the original experimental dataset without the normalization process. Moreover, the mean absolute error (MAE) values were measured for the proposed technique. Table 3.4 illustrated the MAE values for the proposed IT2-TS-FC.

- Identification of the pitch dynamics: the pitch angle was regulated by the operator to set the desired reference, which is manually set. The oscillations were attained by applying different inputs to front actuators with respect to the rear actuators. As shown in Fig. 3.14, the accuracy of the IT2-FLS model is better than its T1-FLS counterpart to represent the dynamics of the pitch loop.
- Identification of the roll dynamics: similar to the pitch dynamics, the roll angle was controlled manually by the operator to give the desired reference signal within the specific range of the indoor flight test area. Whilst collecting the roll data, the system was excited with an oscillating roll motion [240]. Similarly, the performance

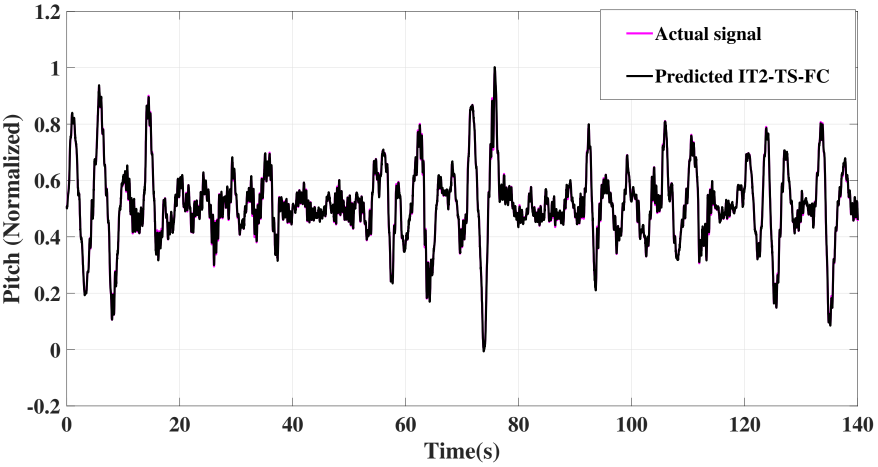


Figure 3.14: Predicted vs. actual signals of the pitch loop of the quadcopter

of the proposed IT2-FLS identification model was compared with a T1-FLS model, where better accuracy was attained using the proposed method, thanks to the $FOU$ in IT2-TS-FC.

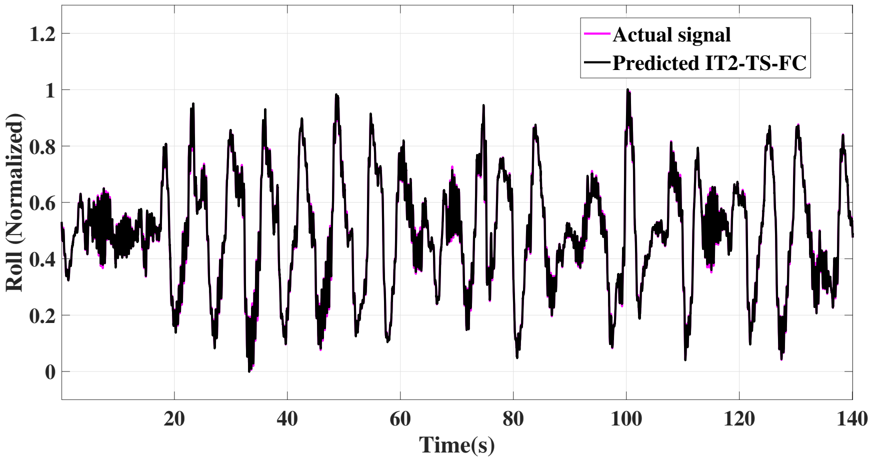


Figure 3.15: Predicted vs. actual signals of the roll loop of the quadcopter

- Identification of the yaw dynamics: the pilot applied yaw control inputs during flight. Likewise, higher accuracy was achieved with lower RMSE values between the actual and predicted model using the proposed IT2-FLS model as demonstrated in Fig. 3.16.

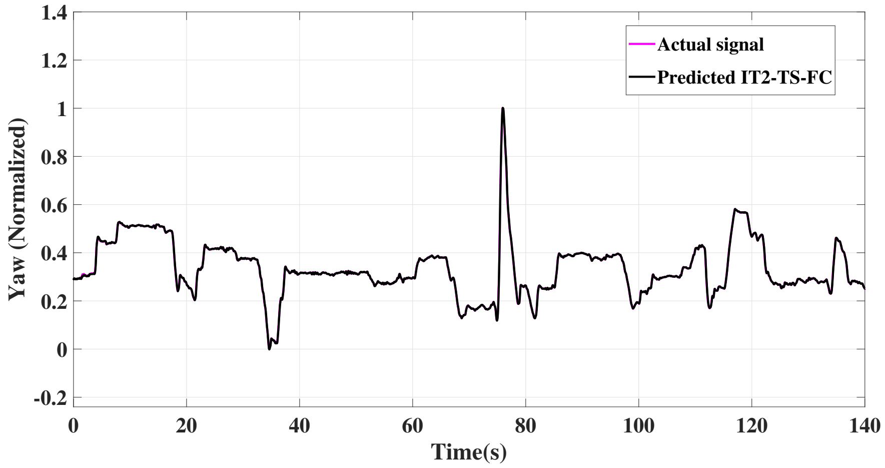


Figure 3.16: Predicted vs. actual signals of the yaw loop of the quadcopter.

- Identification of the altitude dynamics: to model the vertical dynamics, the pilot applies the same thrust to each actuator. Fig. 3.17 demonstrated the identified IT2-FLS model for the altitude dynamics. Lower RMSE values were obtained as shown in Table 3.3.

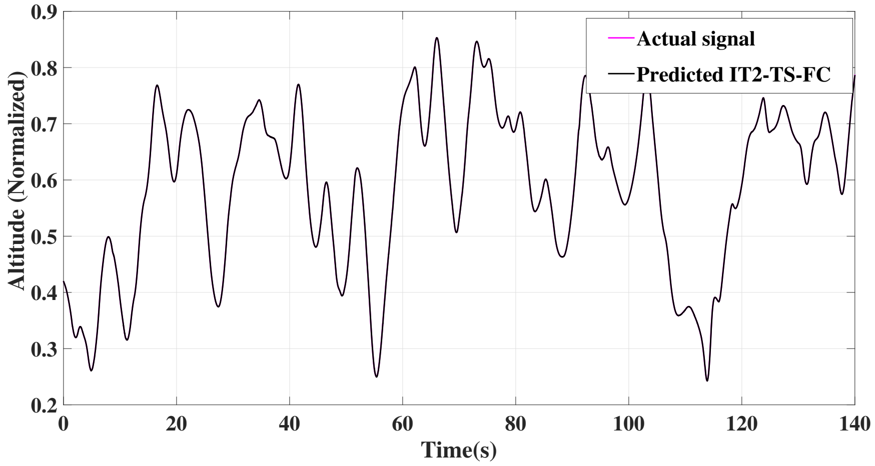


Figure 3.17: Predicted vs. actual signals of the vertical loop of the quadcopter.

- Identification of the altitude dynamics (with noisy samples): To investigate the robustness against uncertainties, artificial white Gaussian noise was added to the samples to corrupt measurement data in the vertical loop. Fig. 3.18 shows the adaptation power of the proposed IT2-FLS method with low RMSE values compared to its type-1 counterpart.

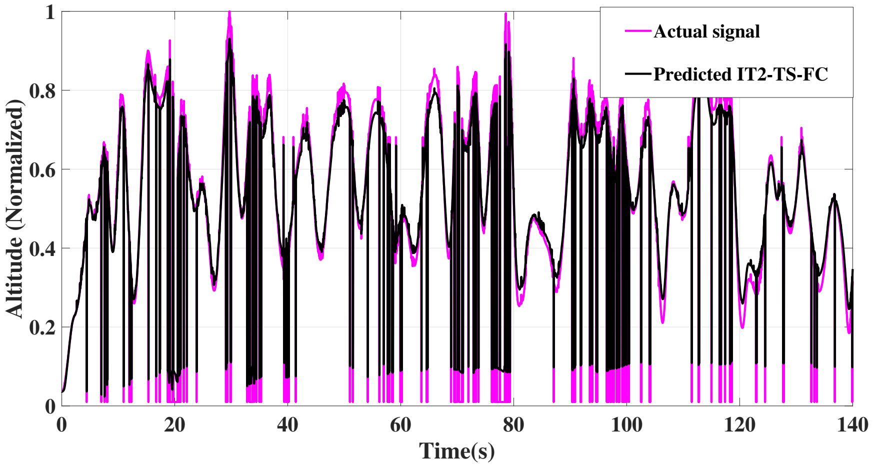


Figure 3.18: Predicted vs. actual signals of the vertical loop of the quadcopter with noisy flight data (different dataset).

Table 3.2: NDEI values for different scaling factor - FoU

| **Properties** | **FoU**=0.25 | **FoU**=0.5 | **FoU**=0.75 |
|---|---|---|---|
| Pitch | 0.0461 | 0.0460 | 0.0455 |
| Roll | 0.0436 | 0.0433 | 0.0435 |
| Yaw | 0.0234 | 0.0234 | 0.0233 |
| Z-Altitude | 0.053 | 0.055 | 0.047 |
| Z-noisy | 0.5633 | 0.5633 | 0.5569 |
| MG | 0.0361 | 0.0278 | 0.0263 |

Table 3.3: Normalized RMSE values of different modeling methods

| **Properties** | **TS Fuzzy** | **TS Fuzzy C-means Clustering** [11] | **IT2-TS-FC** |
|---|---|---|---|
| Pitch | 0.00734 | 0.0853 | 0.0063 |
| Roll | 0.00855 | 0.0833 | 0.0080 |
| Yaw | 0.00856 | 0.03848 | 0.0027 |
| Z-Altitude | 0.00906 | 0.07434 | 0.0078 |
| Z-noisy | 0.2274 | 0.5045 | 0.1274 |
| MG | 0.0082 | 0.04532 | 0.0069 |

Table 3.4: Normalized MAE values for the proposed IT2-TS-FC

| **Properties** | **MAE (Normalized)** |
|---|---|
| | **IT2-TS-FC Clustering** |
| **Pitch** | 0.0047 |
| **Roll** | 0.0054 |
| **Yaw** | 0.0013 |
| **Z-Altitude** | 0.0060 |
| **Z-noisy** | 0.0508 |
| **MG** | 0.0052 |

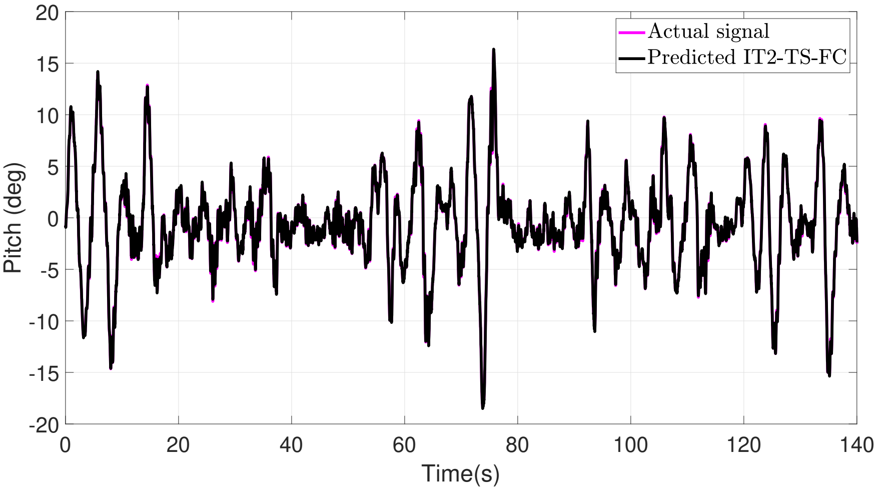


Figure 3.19: Predicted vs. actual signals of the pitch loop of the quadcopter

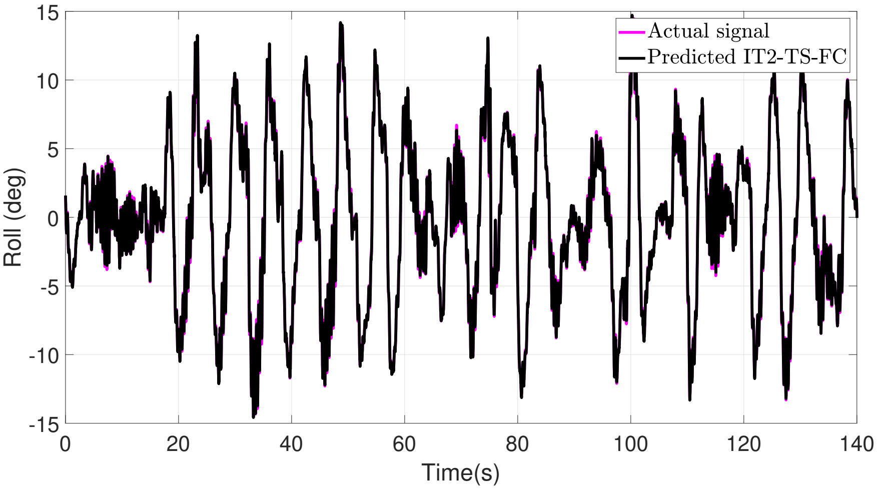


Figure 3.20: Predicted vs. actual signals of the roll loop of the quadcopter

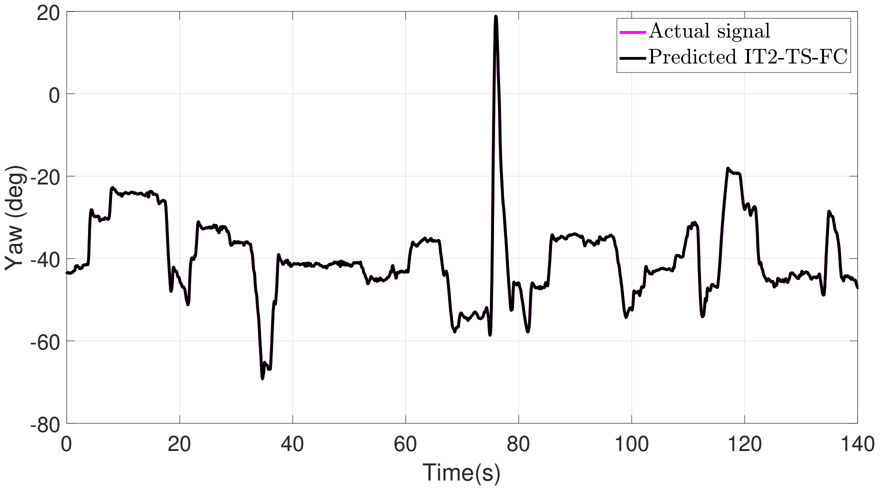


Figure 3.21: Predicted vs. actual signals of the yaw loop of the quadcopter

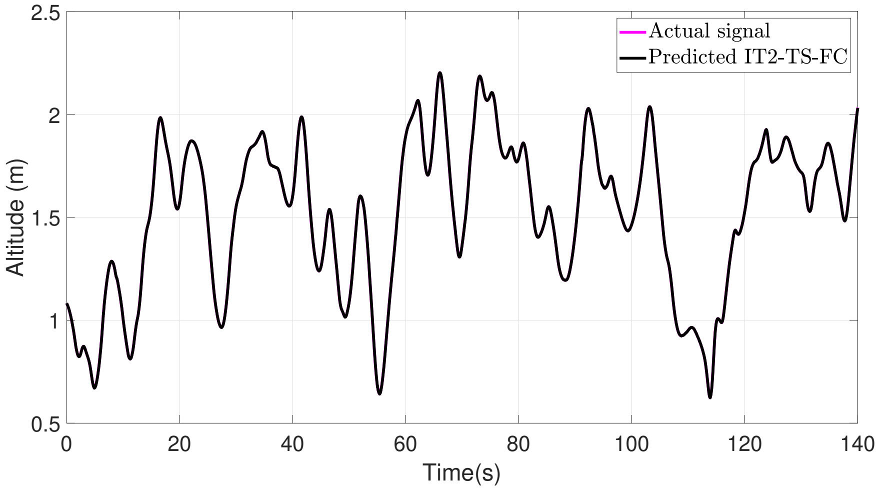


Figure 3.22: Predicted vs. actual signals of the vertical loop of the quadcopter.

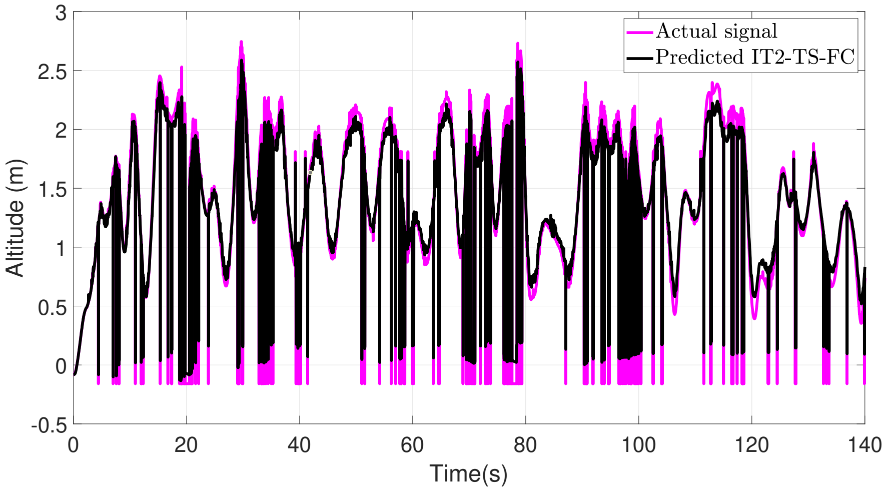


Figure 3.23: Predicted vs. actual signals of the vertical loop of the quadcopter with noisy flight data (different dataset).

## 3.7 Summary

This chapter provided an overview of the experimental setup used in this thesis. This was followed by designing an efficient online system identification technique to model a quadcopter dynamics employing a recursive interval type-2 fuzzy C-means clustering technique. Simulation results illustrated that the accuracy of the acquired IT2-TS-FC models was better than the accuracy of its type-1 fuzzy counterpart as shown by their moderately smaller RMSE and NDEI values. This approach leverages the superior performance of IT2-FLSs for modeling dynamic systems, especially in the presence of noisy datasets. The system was computationally efficient and is suitable for online modeling and control of robotic systems.

# Chapter 4

# Self-Adaptive IT2FLCs for MIMO Nonlinear Systems

The content of this chapter has been taken from the following publications:

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). A Robust Self-Adaptive Interval Type-2 TS Fuzzy Logic for Controlling Multi-Input-Multi-output Nonlinear Uncertain Dynamical Systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, [246].

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). An Intelligent Control of an Inverted Pendulum Based on an Adaptive Interval Type-2 Fuzzy Inference System. 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), [30].

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). A Robust Adaptive Interval Type-2 Fuzzy Control for Autonomous Underwater Vehicles. 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), [36].

## 4.1 Introduction

To date, adaptive closed-loop control systems have received much attention from researchers. Developing adaptive controllers to obtain high accuracy and robustness is usually difficult due to multiple disturbances and uncertainties in the systems. Therefore, the development of intelligent controllers that have the capability of learning to accommodate the footprint-of-uncertainties is necessary. Recently, there has been a growing interest in using intelligent control techniques for various applications such as in autopilot control systems. Smart control techniques such as FLCs, ANNs and learning-based techniques have been gaining popularity [3, 23, 202, 247–251]. The efficacy of FLCs to accommodate uncertainties and nonlinearities has been established beyond doubt. They have demonstrated the ability to handle complex, ill-defined, nonlinear, and time-varying systems. For instance, T1FLCs were utilized for controlling nonlinear systems in [206, 252, 253]. However, to date, most of the existing fuzzy controllers are type-1 FLCs leading to the limited ability of the closed-loop control systems to accommodate the uncertainties.

In contrast to T1FLCs, IT2FLCs provide further flexibility to handle uncertainties as they are specifically designed to adapt with the $FoU$ in the system [129]. Several studies combine IT2FLCs and sliding mode control (SMC) theory. Combining fuzzy systems with SMC theory has the advantages of achieving higher computational efficiency and improving the performance of the system by eliminating the chattering effect of the SMC controllers [128, 131, 254]. Type-2 fuzzy-based sliding mode controllers have been implemented to control nonlinear systems such as UAVs [29, 132], and also for obstacle avoidance and control for mobile robots [114, 254]. Nevertheless, the control laws used in these works also require augmentation by other control approaches such as PID control, which requires additional tuning for its parameters.

In [134–136], the SMC technique was also implemented with fuzzy systems to improve the robustness of some nonlinear control systems. Although their studies achieved good tracking performance, the closed-loop control design was based on multiple SISO systems, which is less efficient than having a single MIMO controller in the loop. In [136, 137], a type-2 fuzzy sliding mode controller was designed for an inverted pendulum, where simulation results demonstrated the robustness of the system. However, their proposed control design was based on the Karnik-Mendel (KM) type-reduction (TR) method, which is computationally intensive.

In recent studies, IT2FLCs have been combined with other control techniques to handle parameter uncertainties, disturbance rejection and robust stabilization. In [138], a $H_2/H_\infty$ based interval type-2 fuzzy controller was proposed for nonlinear systems with minimal control effort to handle uncertainties such as disturbance and measurement noise, where better tracking performance was obtained compared to a type-1 fuzzy controller. An IT2FLCs-based model predictive controller (MPC) was considered for nonlinear networked control systems in the presence of parameter uncertainties and defective communication links with good results in [139]. Moreover, a robust nonlinear control system based on the feedback linearization technique, assisted with IT2FLCs was proposed in [140], for controlling the dynamics of a flapping-wing vehicle.

The remainder of this chapter is structured as follows. The contributions are stated in Section 4.2. Section 4.3 provides the design of the proposed SAF2C, including the stability analysis, followed by the simulation results showing the effectiveness of the proposed control system in dealing with a highly nonlinear system in Section 4.4. Besides, a similar control design is implemented to regulate both the cart position and pendulum angle of an inverted pendulum system in 4.5. Moreover, a similar approach is also proposed for the position and the attitude control of an autonomous underwater vehicle in Section 4.6. Finally, section 4.7 provides a summary of this chapter.

## 4.2 Contribution of this Chapter

Addressing the aforementioned, the contributions of this chapter can be summarized as follows:

1. A new self-adaptive interval type-2 fuzzy controller, named the SAF2C controller,

is introduced whose parameters are tuned automatically using the sliding surface theory. Unlike much of the existing work in T2-FLCs, where the KM type-reduction is utilized, this study employs several type-reduction methods, including the the enhanced iterative algorithm with stop condition (EIASC) and the Nie-Tan (NT) closed-form TR algorithms to improve its efficiency for real-time implementation and also to reduce the computational burden of the KM algorithm. This study also compares the execution time of the three different TR methods.

2. The proposed control system is employed to regulate the position and velocity of a simulated MIMO hexacopter UAV, where the proposed algorithm could achieve about 80% reduction of the execution time compared to the SISO SAF2C controller.

3. The robustness of the proposed controller is investigated in the face of different conditions, such as in the face of external disturbance and parameter variations (e.g. wind gusts) for the case of the hexacopter UAV. Besides, a rigorous comparative study of this controller is performed with respect to the T1FLCs and a conventional PID controller. The outcomes of this research indicate that the proposed SAF2C controller demonstrates an improvement of 20% and 50% in the RMSE values with respect to its type-1 counterpart and the conventional PID controller.

4. Another important consideration in the proposed control design is its ability to filter the measurement noise, where significant improvement is obtained using the proposed controller in the face of measurement noise. The average standard deviation and the mean of the tracking error values are computed for 10 different iterations.

5. The stability analysis of the proposed closed-loop control system is conducted using the Lyapunov theorem.

6. Lastly, we implement our proposed self-adaptive closed-loop control systems in typical uncertain non-linear dynamical systems, namely, an inverted pendulum system and an autonomous underwater vehicle.

## 4.3 SAF2C Design

The architecture of the interval type-2 fuzzy controller is shown in Fig. 4.1. The antecedent part of the proposed controller utilizes interval type-2 fuzzy sets with trapezoidal membership functions, while the consequent part is of the Takagi-Sugeno (TS) type with interval weights. In this work, the rules used are of the TS-type consequent part as follows:

$$IF \;\; x_1(t) \;\; is \;\; \tilde{A}_1^l \; and \; .... \;\; x_m(t) \;\; is \;\; \tilde{A}_m^l$$

$$THEN \; \mathbf{u}_{fuzz} = c_0^l + \sum_{i=1}^{m} c_i^l x_i(t),$$

where $[x_1(t), ..., x_m(t)]$ are the premise variables, $\mathbf{u}_{fuzz}(t)$ is the output fuzzy variables, which form the control signal, $\tilde{A}_i^l$ is the interval type-2 membership functions for the $i$th input variable and $c_i^l$ are the consequent parameters.

The architecture of the proposed control system is made up of five different layers connected sequentially. The structural detail of each layer can be presented as follows:

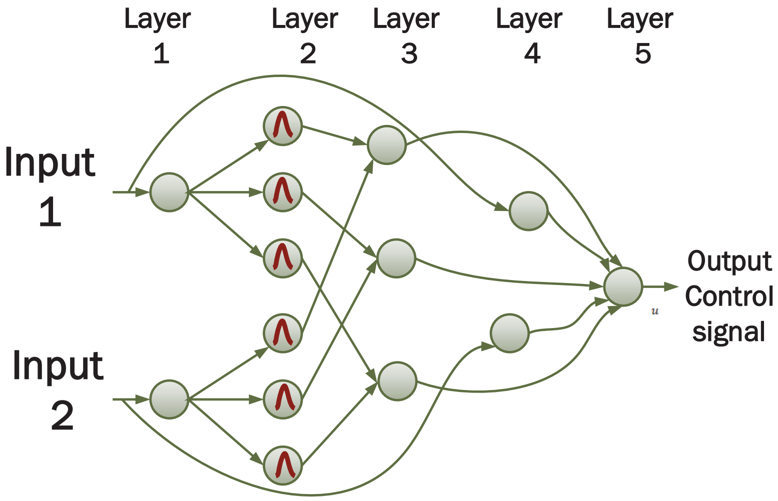


Figure 4.1: Interval type-2 fuzzy controller structure: layer 1 is the input, layer 2 is the fuzzification layer, layer 3 is the firing layer, layer 4 is the consequent layer, and layer 5 is the output control signal.

1. **Layer 1 (the input layer)**: this layer is the input signals which are crisp values. In this study, the inputs are considered to be the error $e(t)$ and its derivative $\dot{e}(t)$.

This layer has no weights to be adjusted.

2. **Layer 2 (the fuzzification layer)**: this layer performs the fuzzification task and assigns a membership degree for each input variable. There are various forms of membership functions. In this work, a IT2 trapezoidal fuzzy set was utilized. Generally, nine parameters are required to represent IT2 FSs [82]. To simplify the design, a symmetric trapezoidal IT2 FS was utilized, which has six parameters, ($a$,$b$,$c$,$d$,$m$,$h$) as demonstrated in Fig. 4.2. The output of the interval type-2 membership functions, $\tilde{A}$, can be described by the upper and lower membership functions within the interval [$\overline{\mu}_{\tilde{A}}$, $\underline{\mu}_{\tilde{A}}$].

   Meanwhile, the trapezoidal type-l fuzzy set is determined by four parameters (if we consider $p = 1$ in T1-FSs) as shown in the blue dotted line in Fig. 4.2 and its membership functions can be represented as follows:

$$\mu_A(x') = \begin{cases} \frac{x'-\grave{a}}{\grave{b}-\grave{a}}, & \grave{a} < x' < \grave{b} \\ p, & \grave{b} \leq x' \leq \grave{c} \\ \frac{\grave{d}-x'}{\grave{d}-\grave{c}}, & \grave{c} < x' < \grave{d} \\ 0 & otherwise. \end{cases} \tag{4.1}$$

   To get the degree of membership of a given input using the trapezoidal FSs, a linear interpolation is implemented. For two values $x_1 < x_2$, and $y_1 = f(x_1)$ and $y_2 = f(x_2)$, the linear interpolation of an input $x'$ is given as follows:

$$f(x') = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1} \cdot (x' - x_1). \tag{4.2}$$

3. **Layer 3 (the firing layer)**: the firing strength $F_i(x')$ is computed in this layer to perform the aggregation operation. Each node in layer 3 represents the antecedent part of the fuzzy rule and the firing strength is computed. To calculate the firing strength $F_i(x')$, a fuzzy *meet* operation is performed in each node for the received inputs from layer 3 using a product (t-*norm*) operation as discussed in Chapter 2 [255].

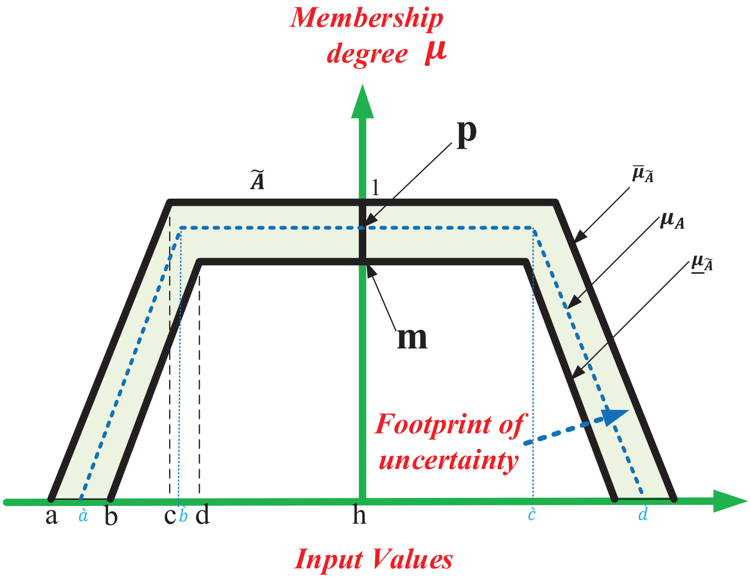


Figure 4.2: An interval type-2 trapezoidal membership function, where the blue dotted line, $A$, represents the type-1 fuzzy membership function and the blurry area between $\bar{A}$ and $\underline{A}$ is called the *Footprint-of-Uncertainty* (FoU).

4. **Layer 4 (the consequent layer)**: the TS fuzzy structure is utilized in this design since it has some structural advantages over the Mamdani-type fuzzy counterpart, where the consequent fuzzy part can be represented in the form of linear equations as in (4.3). Each node in the 4th layer receives inputs from the previous layer and also from the input data in the first layer. The consequent part can be written as follows:

$$W^l(x') = C_0^l + C_1^l e + C_2^l \dot{e}, \tag{4.3}$$

where $C_i^l = [l_i^l, r_i^l]$ $i = 1, .., p$ and $l$ and $r$ denote the left and right bound of an interval; $e(t)$ and its derivative $\dot{e}(t)$ are the inputs to fuzzy control system. The output $W^l(x')$ is a type-1 interval fuzzy number [256].

**Definition 4.3.1** *[256] A type-1 interval fuzzy number $M$ is a type-1 fuzzy number so that $\mu_M(x') = 1, x \in [l, r]$.*

To simplify the notation, the output of this node can be expressed by [256]:

$$w_l^l = \sum_{i=1}^{p} l_0^l + l_i^l x_i', \tag{4.4}$$

$$w_r^l = \sum_{i=1}^{p} r_0^l + r_i^l x_i', \tag{4.5}$$

where $p$ is the number of inputs and $x_0 = 1$.

By using the EIASC algorithm, the left $L$ and right $R$ end points can be computed. One interesting property of the EIASC algorithm is that no rule-reordering is required; unlike the KM-TR algorithm, where the consequent parameters need to be reordered in ascending order. As a recap from chapter 2, the node output of this layer is in the form of the interval type-1 set $[y_{tsk,l}, y_{tsk,r}]$ as follows:

$$y_{tsk,l} = \frac{\sum_{i=1}^{L} \overline{f}_i w_l^l + \sum_{i=L+1}^{M} \underline{f}_i w_l^l}{\sum_{i=1}^{L} \overline{f}_i + \sum_{i=L+1}^{M} \underline{f}_i} \tag{4.6}$$

$$y_{tsk,r} = \frac{\sum_{i=1}^{R} \underline{f}_i w_r^l + \sum_{i=R+1}^{M} \overline{f}_i w_r^l}{\sum_{i=1}^{R} \underline{f}_i + \sum_{i=R+1}^{M} \overline{f}_i} \tag{4.7}$$

In (4.6) and (4.7), $\overline{f}_i$ and $\underline{f}_i$ represent the end points of the firing strength interval for $i^t h$ rule, which can be computed using (2.21) and (2.22).

5. **Layer 5 (the output layer)**: this layer computes the final defuzzified value $y_{fuzz}(x')$ by taking the average of $y_{tsk,l}$ and $y_{tsk,r}$, using the following equation:

$$y_{fuzz}(x') = \frac{y_{tsk,l} + y_{tsk,r}}{2}. \tag{4.8}$$

The proposed IT2 fuzzy controller has the following four fuzzy rules:

$$R^1 : \ IF\ e\ is\ F_1^e\ and\ \dot{e}\ is\ F_1^{\dot{e}},\ THEN\ u_1 = C_{01} + C_{11}e + C_{21}\dot{e}$$

$$R^2 : \ IF\ e\ is\ F_1^e\ and\ \dot{e}\ is\ F_2^{\dot{e}},\ THEN\ u_2 = C_{02} + C_{12}e + C_{22}\dot{e}$$

$$R^3 : \ IF\ e\ is\ F_2^e\ and\ \dot{e}\ is\ F_1^{\dot{e}},\ THEN\ u_3 = C_{03} + C_{13}e + C_{23}\dot{e}$$

$$R^4 : \ IF\ e\ is\ F_2^e\ and\ \dot{e}\ is\ F_2^{\dot{e}},\ THEN\ u_4 = C_{04} + C_{14}e + C_{24}\dot{e},$$

where $e$ represents the closed-loop feedback error, $\dot{e}$ indicates the change of error, $u_i$ is the control output, and $F_{1,2}^e$ and $F_{1,2}^{\dot{e}}$ denote the upper and lower membership degrees of $e$ and $\dot{e}$ respectively. Finally $C_i = [l_i, r_i] \quad i, = 1, .., 3$ are the adaptive consequent fuzzy values, where $l$ and $r$ denote the left and right limits as discussed in (4.3).

### 4.3.1 Sliding Surface Design

Consider a nonlinear dynamic system of $n^{th}$ order as:

$$\mathbf{x}^{(n)} = \mathbf{F}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x}, t)\mathbf{u}(t) + \mathbf{d}(t), \tag{4.9}$$

where $\mathbf{x} = [x, \dot{x}, ..., x^{(n-1)}]^T \in \mathbb{R}^n$ represents the state vector, $\mathbf{F}(x, t)$ and $\mathbf{g}(x, t)$ are nonlinear functions of the state vector, $\mathbf{u}(t)$ is the control input and $\mathbf{d}(t)$ is the external disturbance. For the theoretical study, it is assumed that the nonlinear terms in Eq. (4.9) are known, bounded and $\mathbf{g}^{-1}(x, t)$ exists for all $x$. The tracking error of the state variables can be defined as:

$$\mathbf{e}(t) = \mathbf{x}_d(t) - \mathbf{x}(t) = \left[e(t),\ \dot{e}(t), ...,\ e(t)^{(n-1)}\right]^T, \tag{4.10}$$

where $\mathbf{x}_d = [x_d, \dot{x}_d, ..., x_d^{(n-1)}]^T$ is the desired tracking vector. Hence, (4.10) can be extended as:

$$\begin{aligned} \dot{e}_1(t) &= e_2(t), \\ \dot{e}_2(t) &= e_3(t), \\ &\vdots \\ \dot{e}_n(t) &= \dot{x}_{dn} - \mathbf{F}(\mathbf{x}, t) - \mathbf{g}(\mathbf{x}, t)\mathbf{u}(t). \end{aligned} \tag{4.11}$$

The sliding surface $s(e, t) = 0$ can be defined as:

$$s_s(t) = \gamma e(t) + \dot{e}(t), \tag{4.12}$$

where $\gamma$ is a strictly positive real constant. If all the nonlinear functions of (4.9) are known, a control law can be written as [257]:

$$\mathbf{u}_{tot}(t) = \frac{1}{\mathbf{g}(\mathbf{x}, t)}\left[\dot{x}_{dn}(t) - \mathbf{F}(x, t) - \dot{e}(t) - \mathbf{d}(t) + \dot{s}_s(t) + \gamma s_s(t)\right]. \tag{4.13}$$

The error $e(t)$ and its derivative $\dot{e}(t)$ can be considered as the input variables to the proposed fuzzy controller. To produce the control signal $u(t)$, fuzzy operations are employed so that the equivalent control law $\mathbf{u}_{tot}(t)$ can be approximated. The purpose of SMC is to derive the system dynamics to the sliding surface $s_s(t) = 0$, so that $\dot{e}(t) + \gamma e(t) = 0$. Similarly, if we consider the sliding surface and its derivative as the input variables to the fuzzy system and also by following the control law $\mathbf{u}_{tot}(t)$, the dynamic behavior of the closed-loop control system is asymptotically stable [257], that is,

$$\gamma s_s(t) + \dot{s}_s(t) = 0. \tag{4.14}$$

Both the sliding surface $s_s(t)$ and its derivative $\dot{s}_s(t)$ will eventually converge to zero since $\gamma$ is always positive value. In addition, the errors $[e(t),\ \dot{e}(t)]$ of the system output states will also converge to zero based on the definition in (4.12). Instead of the model-based calculation, this study will employ a fuzzy system to map between the sliding variables and the control law $\mathbf{u}_{fuzz}(t)$. The fuzzy control input in this case may have differences with the optimal control law $\mathbf{u}_{tot}(t)$ and can be derived from Eqs. (4.11) and (4.13) as follows:

$$\dot{s}_s(t) = \mathbf{g}(x,t)\left[\mathbf{u}_{tot}(t) - \mathbf{u}_{fuzz}(t)\right] - \gamma s_s(t). \tag{4.15}$$

By multiplying (4.15) with $s(t)$, it yields to:

$$\dot{s}_s(t)s_s(t) = s_s(t)\left(\mathbf{g}(x,t)\left[\mathbf{u}_{tot}(t) - \mathbf{u}_{fuzz}(t)\right] - \gamma s_s(t)\right). \tag{4.16}$$

**Remark 1** *There are two phases in the SMC design: 1) the reaching phase, where the trajectory of a system should reach a sliding surface in a finite time, 2) the sliding phase in which the desired system response should follow the desired sliding surface [202].*

According to the Lyapunov theorem, the sliding surface-reaching condition is $s_s(t) \cdot \dot{s}_s(t) < 0$. Hence, if the outcome of this study is to design a control signal $\mathbf{u}(t)$ that satisfies the aforementioned condition, the convergence of the control system to the phase plane is guaranteed.

### 4.3.2 Adaptation Law

From (4.3) and (4.8), the adaptation law of the proposed SAF2C controller is derived based on the gradient descent method to decrease the $s_s(t)\dot{s}_s(t)$ with respect to $c_i$. Hence, the modified $c_i$ can be expressed as follows [257]:

$$c_{new} = c_i - \Lambda \frac{\partial s_s(t)\dot{s_s}(t)}{\partial c_i(t)}, \tag{4.17}$$

where $\Lambda$ is an adaptive parameter and $c_i$ is the consequent fuzzy values at $t = 0$. Also, $c_{new}$ has the left and right terms as explained in (4.3). By employing the chain rule, (4.17) can be rewritten as:

$$c_{new} = c_i - \Lambda \frac{\partial s_s(t)\dot{s_s}(t)}{\partial u(t)} \frac{\partial u(t)}{\partial c_i(t)} = c_i + \Lambda \mathbf{g}(x,t) s_s(t) \frac{\partial u(t)}{\partial c_i(t)}. \tag{4.18}$$

For further simplification, the adaptive parameter $\Lambda$ can be combined with the system input parameter $\mathbf{g}(x,t)$ to form the learning rate $\zeta$. Hence, for practical implementation, the adaptation law of the consequent part $c_i$ can be written as:

$$\begin{aligned} c_{new} = c_i + \gamma s_s(t) \Big( & \frac{\overline{f}_i(x'))}{\sum_{i=1}^{m}(\overline{f}_i(x') + \underline{f}_i(x'))} \\ & + \frac{\underline{f}_i(x'))}{\sum_{i=1}^{m}(\overline{f}_i(x') + \underline{f}_i(x'))} \Big). \end{aligned} \tag{4.19}$$

To enhance the performance of the IT2FS and to minimize the drift of the fuzzy consequent parameters, a modification in [258] and the dead-zone concept in [259] are introduced. Therefore, the modified equation of the interval type-2 fuzzy consequent parameter can be represented as follows:

$$\begin{cases} c_{new} = 0, & if\ s_s \leq \psi \\ c_{new} = \kappa |s_s| c_i + \gamma s_s(t) \Big( \frac{\overline{f}_i(x'))}{\sum_{i=1}^{m}(\overline{f}_i(x') + \underline{f}_i(x'))} & \\ \qquad + \frac{\underline{f}_i(x'))}{\sum_{i=1}^{m}(\overline{f}_i(x') + \underline{f}_i(x'))} \Big), & if\ s_s > \psi, \end{cases} \tag{4.20}$$

where $\kappa$ is $> 0$ and $\psi$ is the dead-zone parameter.

### 4.3.3 Stability Analysis

One of the most popular stability methods is the Lyapunov stability technique. Hence, in this section, it is implemented to study the stability property of our proposed controller.

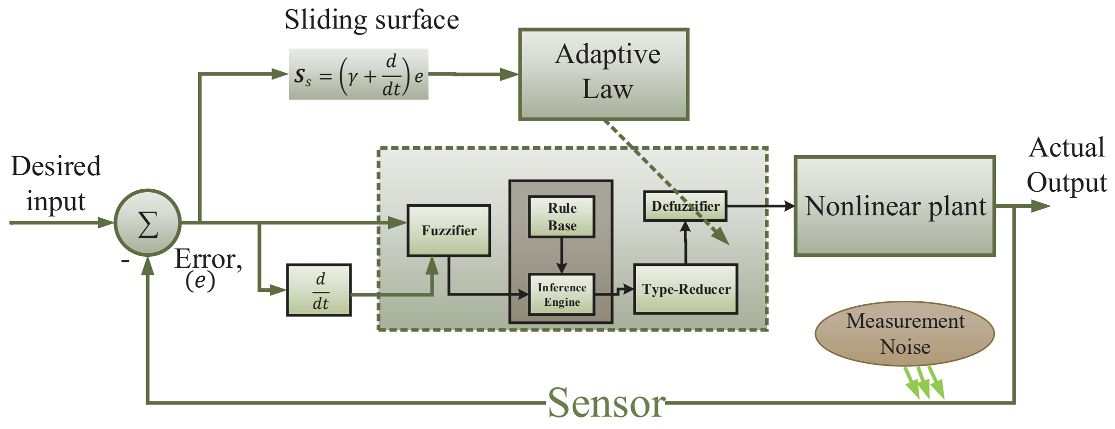


Figure 4.3: Block diagram of the control loop describing the learning algorithm of the proposed adaptive controller.

A fuzzy system can be utilised to represent and approximate any nonlinear function to obtain reasonable accuracy. The following basic assumptions were made in order to assess the stability of the proposed SAF2C.

**Lemma 4.3.1** *[257] If an optimal consequent parameters vector $\overline{C}$ exists, leading to the control law $\mathbf{u}_{fuzz}$, the total approximation of the control law $\mathbf{u}_{tot}$ has bounded error of $< \gamma$, then:*

$$max|\tilde{u}(x,\bar{C}) - \mathbf{u}_{tot}(x)| < \gamma, \tag{4.21}$$

where $\tilde{u}(x,\overline{C}) = \sum_{i=1}^{m} \Psi_i \bar{c}_i = \overline{C^T}\Psi$ and $\mathbf{u}_{tot}(t)(x) = \overline{C^T}\Psi + \gamma$.

If we define $\bar{C} = \overline{C} - \tilde{C}$ as the difference between the actual and the desired fuzzy consequent parameters, (4.15) can be rewritten as follows:

$$\dot{s}_s(t) = g[\Psi_i \bar{C}^T + \gamma] - \lambda s_s(t). \tag{4.22}$$

In this study, the Lyapunov function is chosen as:

$$V(s) = \frac{1}{2}s_\nabla^2 + \frac{g}{2\xi}\bar{C}^T\bar{C}, \quad \zeta \neq 0, \tag{4.23}$$

where $s_\nabla \equiv s_s - \psi \bullet (s_s/\psi)$ and $\psi$ represents the boundary layer thickness [257].

Differentiating (4.23) with respect to time, we finally arrive at:

$$\begin{aligned}
\dot{V} \quad &= \quad s_\nabla \dot{s}_\nabla + \tfrac{g}{\xi}\bar{C}^T\dot{\tilde{C}} + \tfrac{\dot{g}}{2\zeta}\bar{C}^T\bar{C} \\
&= \quad s_\nabla \dot{s}_\nabla + \tfrac{g}{\xi}\bar{C}^T\left[\dot{\bar{C}} - \dot{\hat{C}}\right] + \tfrac{\dot{g}}{2\zeta}\bar{C}^T\bar{C} \\
&= \quad s_\nabla\left[-\lambda s + g\left(\bar{C}^T\Psi + \gamma\right)\right] \\
&\quad\quad -g\bar{C}^T\left(s_\nabla\Psi - \tfrac{\kappa}{\zeta}|s_\nabla|\tilde{C}\right) + \tfrac{\dot{g}}{2\zeta}\bar{C}^T\bar{C} \qquad (4.24) \\
&= \quad s_\nabla\left[-\lambda(s_\nabla + \psi) + g\gamma\right] + g\kappa|s_\nabla|\bar{C}^T\tilde{C} + \tfrac{\dot{g}}{2\zeta}\bar{C}^T\bar{C} \\
&\leq \quad |s_\nabla|\left(-\lambda|s_\nabla| - \lambda\psi + g\gamma - g\tfrac{\kappa}{\zeta}\bar{C}^T\bar{C} + \tfrac{\kappa}{\zeta}g|\bar{C}||\overline{C}|\right) + \tfrac{\dot{g}}{2\zeta}\bar{C}^T\bar{C} \\
&= \quad -|s_\nabla|\Theta - \tfrac{1}{\zeta}\left[|s_\nabla|g\kappa - \tfrac{\dot{g}}{2}\right]\bar{C}^T\bar{C}
\end{aligned}$$

where $\Theta = \lambda\psi + \lambda|s_\nabla| - g\left(\gamma + \frac{\kappa}{\zeta}|\bar{C}||\overline{C}|\right)$.

By choosing appropriate parameters for $\psi$ and $\kappa$ for $\Theta > 0$, (4.24) implies that $\dot{V} < 0$ whenever $s_\nabla \notin R \equiv (|s_\nabla| < (\frac{\dot{g}}{2\zeta\kappa g}))$. Hence, the control system is stable based on the Lyapunov theorem as the Lyapunov function is steadily reduced. In addition, the sliding surface $s_s(t)$ and its derivative $\dot{s}_s(t)$ converge into the boundary layer as long as $\bar{C}$ remains bounded, where the value of the boundary layer depends on the dead-zone parameter $\psi$ and the approximate error of the fuzzy system $\gamma$.

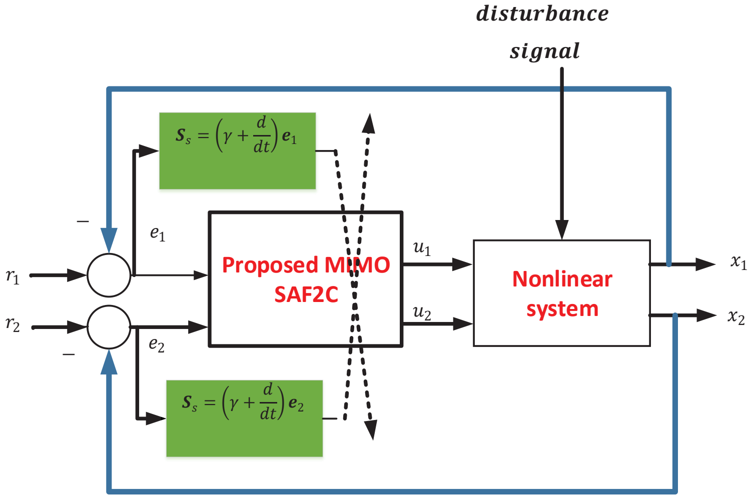


Figure 4.4: Simplified structure of the proposed SAF2C MIMO system with disturbance signal.

## 4.4 Simulation Results and Discussion

This section studies the performance of the closed-loop control system for SISO and MIMO systems. The performance of the closed-loop control system will be explored and benchmarked with respect to a type-1 fuzzy SMC controller and a conventional PID controller. In addition, the SAF2C is explored with different TR methods and different levels of *FoUs*.

### 4.4.1 Case A: second order Van Der Pol system

For a SISO system, the Van Der Pol second-order nonlinear system is chosen to highlight the efficacy of the proposed controller, which can be represented as follows:

$$m\ddot{x} + 2c(x^2 - 1)\dot{x} + x = \mathbf{u}, \tag{4.25}$$

where $x$ indicates the position coordinate, which is a function of time ($t$); $c$ is a scalar parameter expressing the nonlinearity of the damping; $m$ is a positive constant and $\mathbf{u}$ is the control force.

Several tracking trajectories were utilized to investigate the efficacy of the proposed controller, where the time step is 0.02 *sec*. The performance of the proposed SAF2C controller with respect to a step reference signal and also a sine wave reference, with a frequency of 0.25 ($rad/s$) and an amplitude of 2 $m$, can be shown in Figs. 4.5.a & 4.5.b, respectively. The performance of the closed-loop control system is compared with respect to the type-1 fuzzy and conventional PID controller. Higher tracking accuracy was obtained for using the SAF2C controller as indicated by the lower RMSE values as summarized in Table 4.1.

### 4.4.2 Case B: Altitude control of a hexacopter

To investigate the efficacy of the proposed controller in the face of nonlinearity, it is applied to control the nonlinear aerodynamics of a hexacopter described in [201, 260].

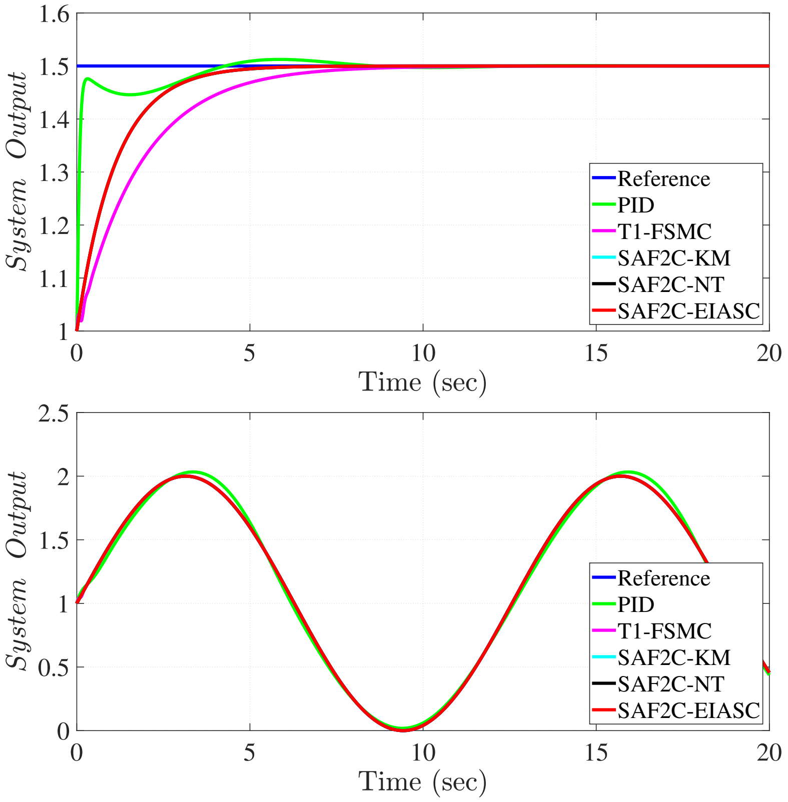


Figure 4.5: The Van Der Pol Oscillator SISO System with respect to a step and a sine reference signals.

The motion of a hexacopter model can be expressed by the following variables:

$$x = [u\ v\ w\ p\ q\ r\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}]^T$$

where $[u,\ v,\ w]^T$ denotes the velocity components along the $(x, y, z)$-axes. Likewise, $[p,\ q,\ r]^T$ are the body rotation rates. Additionally, $[\theta,\ \psi,\ \phi]^T$ are the three Euler angles, which are introduced to describe the aircraft orientation (pitch, yaw, and roll), respectively. The angular velocities of the aircraft are represented by $[\dot{\theta},\ \dot{\psi},\ \dot{\phi}]^T$. To calculate the rotor thrust and the induced velocity in both the forward flight mode and the hover mode, Glaeurt's induced flow model [261, 262] is employed.

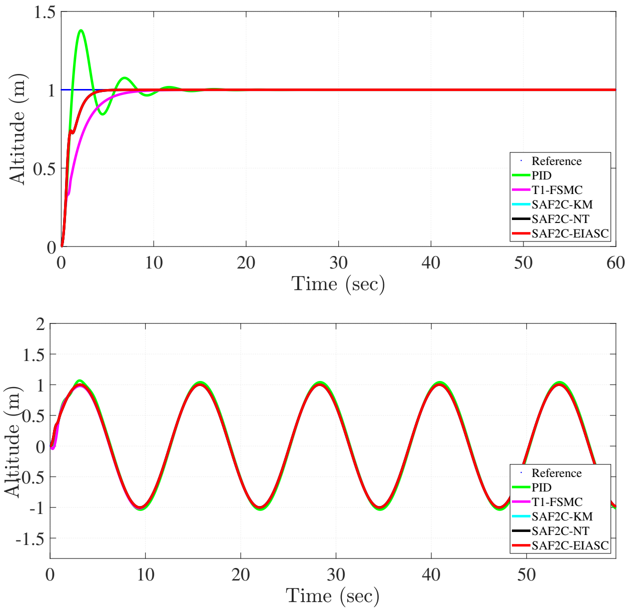


Figure 4.6: Altitude tracking performance of a Hexacopter using the SAF2C controller (step and sine wave reference signals).

In general, hexacopter rotors experience the free-stream velocity due to $V_\infty$ given by (4.26). The actuator disk deflects the air-stream by the speed of $V_i$ and this changes the downstream flow by $2V_i$. This flow consists of two components, namely the normal velocity of $V_n$ and the tangential velocity $V_t$. The computation of $V_n$ and $V_t$ can be done by summing up the normal and tangential components of $V_\infty$ to the airflow, which is created by pitching, rolling and yawing motions of each rotor.

$$V_\infty^2 = u^2 + v^2 + w^2 = V_n^2 + V_t^2. \tag{4.26}$$

In this model, we consider a uniform inflow, where it is assumed that the inflow $V_i$ remains unvaried with radius or azimuth. The thrust can be represented by considering a uniform inflow, where it is assumed that the inflow $V_i$ remains unvaried with radius or azimuth. Therefore, the model can be obtained by integrating the elemental forces as a function of the inflow relative to the rotor disk ($\lambda'$), the blade pitch ($\theta_0$) and advance ratio ($\mu$) as follows:

$$T = \frac{\rho a (\Omega R)^2 A_b}{2} \left[ \frac{1}{3} \theta_0 \left( 1 + \frac{3}{2} \mu^2 \right) - \frac{1}{2} \lambda' \right], \tag{4.27}$$

where $\Omega$ represents the rotational speed of the blade, $A_b$ denotes the total blade area ($N$ blades), $a$ is the lift curve slope and $\mu = \frac{V_t}{\Omega R}$ whereas $\lambda' = \frac{V_i + V_n}{\Omega R}$. The details of the drone mathematical model can be found in [260].

For the purpose of control analysis, we assume that the hexacopter drone is a rigid body. Hence, we utilize Newton's second law of motion to determine the relationship between all forces and moments, and also the relationship between the linear and angular accelerations. By considering our hexacopter UAV to be of a conventional mass distribution, the drone is symmetrical around the $xz$ plane, so that the cross product moments of inertia ($I_{xy}$ and $I_{yz}$) are both zero. The rigid body dynamic equations of our drone can be found in most flight mechanics textbooks, such as the textbook written by Nelson [263, 264] as follows:

$$\begin{cases} F_x &= m(\dot{u} + qw - rv) \\ F_y &= m(\dot{v} + ru - pw) \\ F_z &= m(\dot{w} + pv - qu) \\ L &= I_x \dot{p} - I_{xz} \dot{r} + qr(I_z - I_y) - I_{xz} pq \\ M &= I_y \dot{q} + rp(I_x - I_z) + I_{xz}(p^2 - r^2) \\ N &= -I_{xz\dot{p}} + I_z \dot{r} + pq(I_y - I_x) + I_{xz} qr, \end{cases} \tag{4.28}$$

where $m$ represents mass; $[L\ M\ N]^T$ denotes the net torque to the angular momentum rate of change; $F_x,\ F_y,\ F_z$ and $I_x,\ I_y,\ I_z$ denote the forces and moments around $(x,\ y,\ z)$-axes.

The simulated hexacopter attitudes are represented as a quaternion and updated using (4.29) as explained in [265]. By using this method, the requirement to deploy trigonometric functions for integrating the Euler angle differential equations is eliminated [264].

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = -0.5 \begin{bmatrix} 0 & p & q & r \\ -p & 0 & -r & q \\ -q & r & 0 & -p \\ -r & -q & p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \tag{4.29}$$

The final process for updating the position of the hexacopter drone in global coordinates relative to an earth-fixed axes system can be performed using the rotation matrix $B$ as in [260], where the linear velocities $[u,\ v,\ w]$ are first transformed to the global velocities $[\overline{X},\ \overline{Y},\ \overline{Z}]^T = B[u\ v\ w]^T$. The rotation matrix $B$ can be represented as follows:

$$B = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_2 + q_0q_3) \\ 2(q_1q_2 - q_0q_3) & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_1q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}, \tag{4.30}$$

where

$$\begin{cases} q_0 & = \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ q_1 & = \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ q_2 & = \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\ q_3 & = \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2}. \end{cases} \tag{4.31}$$

The simulated dynamic model was implemented in MATLAB/SIMULINK using $S$-function, where the state variables are: the (1) position; (2) velocity; (3) rotational rates; (4) quaternion attitude.

The proposed SAF2C controller is designed to control a nonlinear 6-DOF hexacopter plant. First, we employ our controller to regulate the altitude loop of the hexacopter plant in a nominal condition with various tracking trajectories. The effectiveness of the proposed SAF2C is evaluated with the nominal hexacopter model. For all the trajectories, the performance of our proposed SAF2C is compared with the conventional PID controller in addition to the type-1 fuzzy-based sliding mode control (T1-SMC). Fig. 4.6 shows the

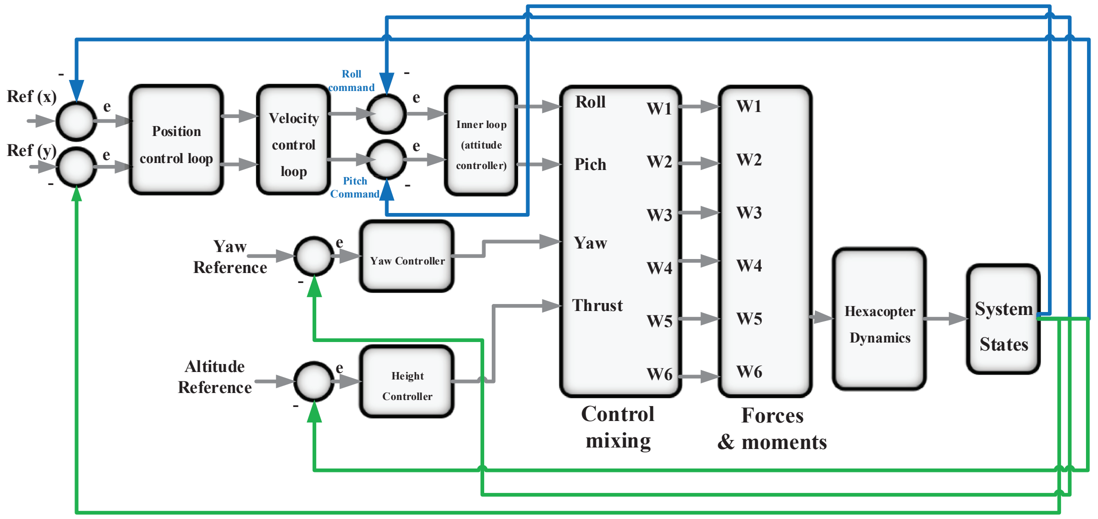


Figure 4.7: Simulated model of an over-actuated hexacopter.

response to a step and sine wave input, where the step signal has an amplitude of 1 $m$ and the sine wave signal has a frequency of 0.5 $(rad/s)$ and an amplitude of 1 $m$. As can be seen, the system can achieve faster convergence using our proposed controller and also the results of the SAF2C controller outperformed the benchmark controllers as represented by its lower Root Mean Square Error (RMSE) values as summarized in Table 4.1. Moreover, the error signal and the control signal of our proposed SAF2C are shown in Fig.4.11.

### 4.4.3 Robustness Analysis

To evaluate the robustness of the proposed SAF2C controller, we use several parameters such as the RMSE values, the settling time $(s)$, and the rise time $(s)$. For the altitude control loop, the proposed controller's performance was tested in the presence of disturbance. Also, the performance indices were evaluated in the presence of high wind-gusts.

#### 4.4.3.1 Simulation results of a hexacopter in the face of disturbance

To evaluate the efficacy of our proposed controller, a pulse signal was added at time 8 *sec* (see Fig. 4.9. a) to the feedback of the altitude state. The response of the closed-loop control system with respect to a step input and sine wave trajectory with a frequency of 0.5 $(rad/s)$ and an amplitude of 1 $m$ can be seen in Fig. 4.8. The performance is also compared with a conventional PID controller and T1-SMC. It is observed that better accuracy was obtained using our proposed controller with lower RMSE values as summarized in Table 4.1. It can also be seen that the PID controller fails to follow the desired trajectory when disturbances occurred. Lastly, the learning parameters of the consequent fuzzy parts for the hexacopter altitude control scenario in the presence of disturbances are shown in Fig.4.11.

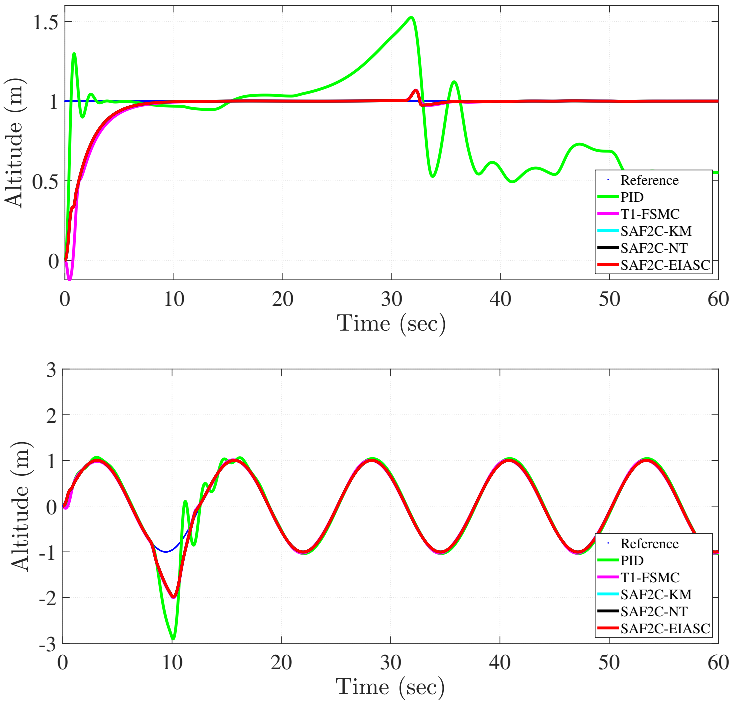


Figure 4.8: Altitude tracking performance of our hexacopter using our proposed controller in the presence of external disturbances (step and sine wave reference signals).

#### 4.4.3.2 Simulation results of a hexacopter in the face of wind-gusts

To investigate the efficacy of our proposed controller, artificial wind gusts were added to the system with a maximum velocity of 2 $(m/s)$ and an amplitude of 5 $m$ for the $xyz$-axes as shown in (Fig. 4.9.b). The signal was added to the plant after 5 $sec$ to study the

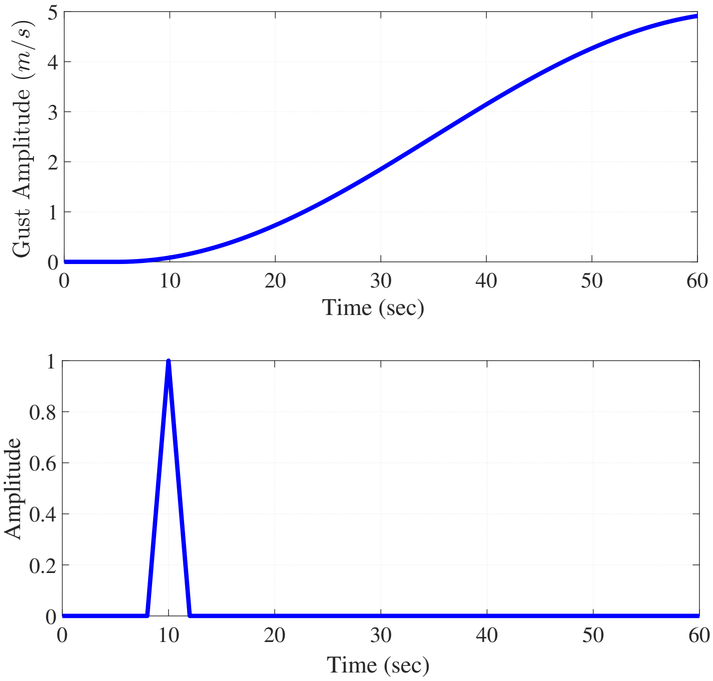


Figure 4.9: (a) The disturbance signal. (b) The wind gust disturbance signal.

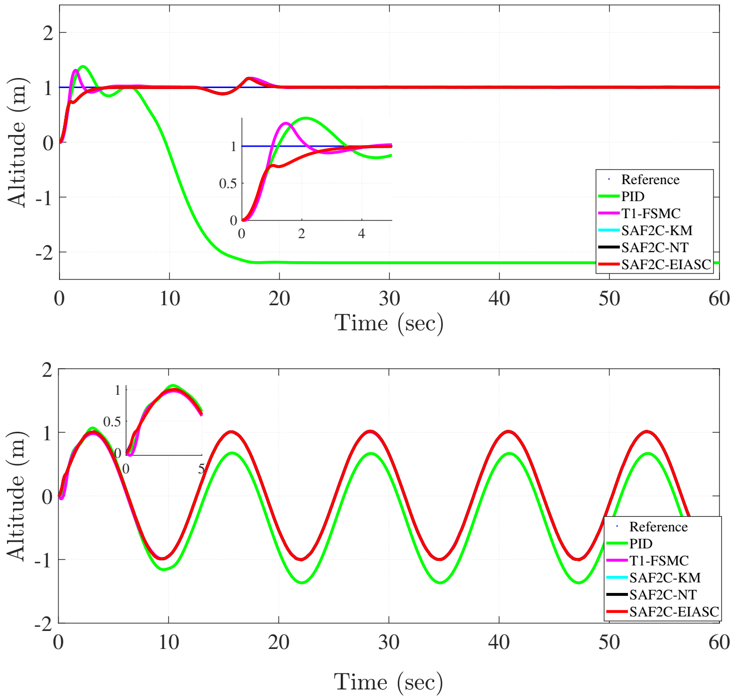


Figure 4.10: Altitude tracking capability of our hexacopter using our proposed controller with wind-gusts (step and sine wave reference signals).

Table 4.1: Comparative study of multiple controllers' performance under different flight scenarios

| **Robustness Analysis** | **Measured Features** | **Sine Wave Reference** | | | **Step Reference** | | |
|---|---|---|---|---|---|---|---|
| | | **PID** | **T1-SMC** | **SAF2C** | **PID** | **T1-SMC** | **SAF2C** |
| Van der Pol oscillator | **Test RMSE** | 0.0107 | 0.0110 | **0.0024** | 0.0538 | 0.0710 | **0.0168** |
| | **Settling time (sec)** | 2.0140 | 2.0093 | 2.0023 | 4.2235 | 6.8998 | 3.3770 |
| | **Rise time (sec)** | 0.2473 | 0.2395 | 0.2186 | 2.4470 | 4.0611 | **0.4148** |
| Altitude control in nominal condition | **Test RMSE** | 0.174 | 0.0615 | **0.00345** | 0.4229 | 0.15437 | 0.**12489** |
| | **Settling time (sec)** | 1.0409 | 1.002 | 1.0009 | 6.422 | 7.3404 | 7.0069 |
| | **Rise time (sec)** | 2.009 | 8.190 | 1.993 | 0.3505 | 3.6361 | 3.8716 |
| Altitude control with disturbance | **Test RMSE** | 0.6466 | 0.0654 | **0.02275** | 0.7474 | 0.15529 | **0.12595** |
| | **Settling time (sec)** | 1.05 | 1.003 | 1.0454 | 16.619 | 13.670 | 12.670 |
| | **Rise time (sec)** | 1.775 | 7.94 | 1.7498 | 0.3505 | 3.63619 | 3.8716 |
| Altitude control with wind-gust | **Test RMSE** | 0.16685 | 0.0615 | **0.00345** | 0.4254 | 0.1543 | **0.12489** |
| | **Settling time (sec)** | 1.03549 | 1.0028 | 1.00107 | 1.541 | 0.899 | **0.899** |
| | **Rise time (sec)** | 2.5690 | 8.192 | 1.9946 | 0.319 | 3.619 | 3.8637 |

robustness against uncertainties and perturbations. Fig. 4.10 illustrates the efficacy of our proposed SAF2C, where the effect of strong wind gusts was reduced sharply using the proposed SAF2C. The RMSE values were tabulated in Table 4.1. Moreover, a significant deterioration in tracking the desired trajectories was observed using the conventional PID controller.

#### 4.4.3.3 The RMSE criterion

The RMSE criterion in (4.32) is utilized to assess the performance of our proposed SAF2C with other benchmark controllers such as the type-1 fuzzy SMC and the conventional PID controller. The RMSE is calculated by finding the difference between the desired reference and the actual state.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(x(i) - x_d(i)\right)^2}{N}}, \tag{4.32}$$

where $x$ and $x_d$ are the actual and desired states, and $N$ denotes the total number of observations.

### 4.4.4 Case C: MIMO nonlinear systems

The concept of controlling SISO systems can be extended to MIMO systems. To examine the effectiveness of the proposed SAF2C, it was employed to control a coupled nonlinear benchmark system [202] and also a hexacopter MIMO nonlinear model as elaborated in Examples 1 and 2.

#### 4.4.4.1 Example 1: A MIMO nonlinear coupled system

To assess the tracking ability of our proposed controller, we employed our adaptive fuzzy control system to regulate the MIMO nonlinear system, whose dynamics can be described

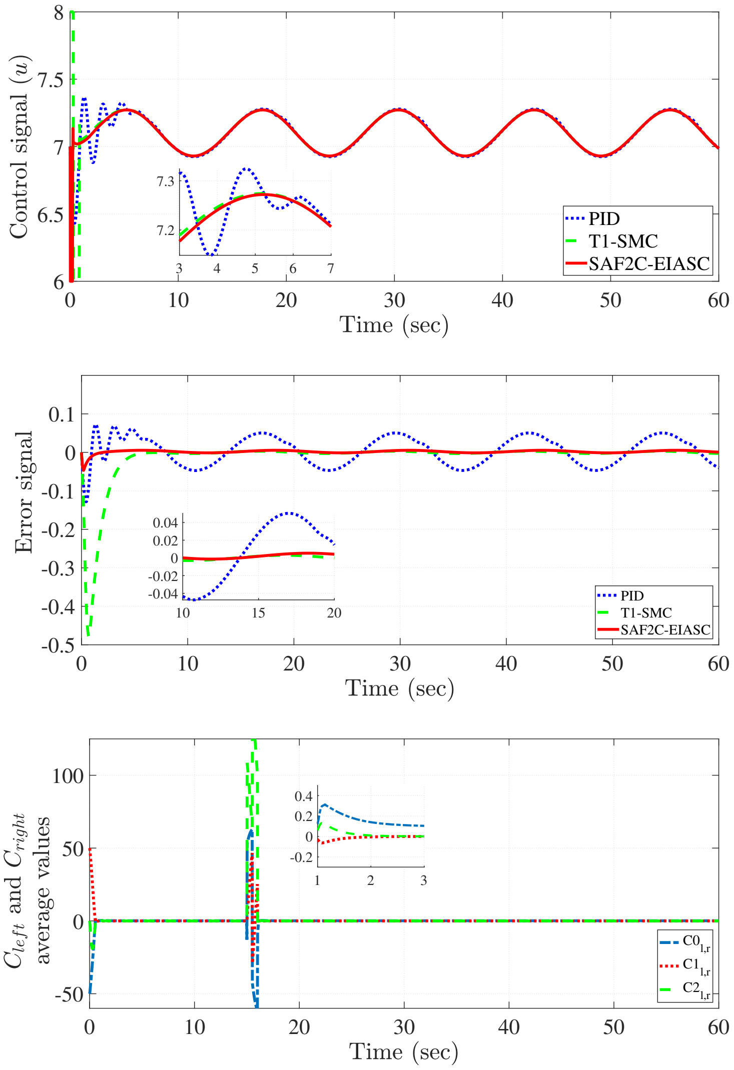


Figure 4.11: (a) Control signals of our hexacopter plant, (b) error signals of our hexacopter, (c) the Adaptation of the consequent parts of the altitude control of our hexacopter with disturbance.

as follows:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_1(t)\ x_2(t) + sin(x_1(t)) \\ x_1(t)^3 + 0.5\ sin(x_2(t)) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u(t). \tag{4.33}$$

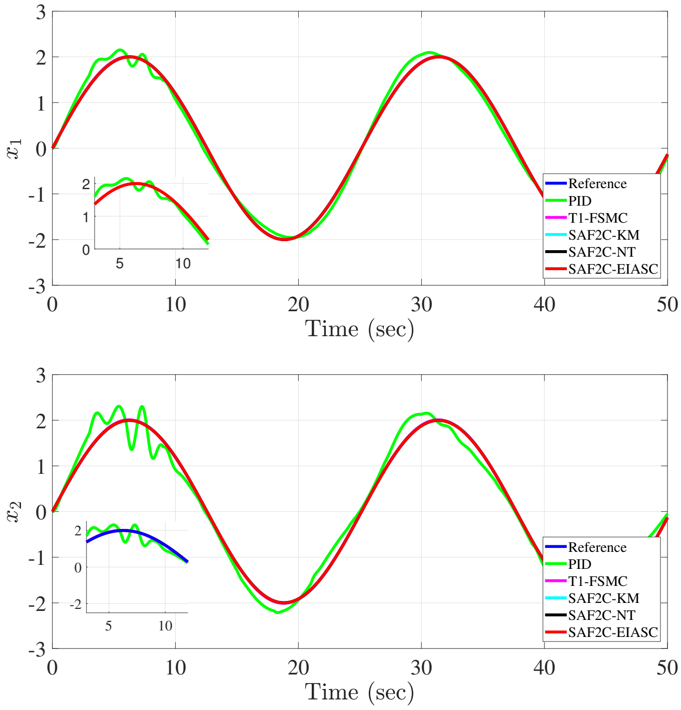


Figure 4.12: The proposed control performance for MIMO system - $x_1$, $x_2$ with sine wave reference in the presence of external disturbance.

Likewise, in the presence of external disturbance, a sine wave signal with an amplitude of 2 $m$ and a frequency of 0.25 $(rad/s)$ was utilized as a desired tracking trajectory as shown in Fig. 4.12. The system is highly nonlinear, coupled, and MIMO. The performance of our proposed controller is observed and compared with respect to the benchmark controllers, where better tracking results are achieved from the proposed SAF2C controller. The RMSE values were tabulated in Table 4.2.

#### 4.4.4.2 Example 2: Hexacopter position and velocity control

Similarly, the proposed controller was employed to control the position and the velocity of a MIMO hexacopter in the $x$ and the $y$ directions.

Table 4.2: RMS error values for MIMO nonlinear systems.

| **Variables to be controlled** | **RMSE Values** | | |
|---|---|---|---|
| | **PID** | **T1-SMC** | **Proposed SAF2C controller** |
| $x_1$ $(m)$ | 0.1413 | 0.0126 | 0.0020 |
| $x_2$ $(m)$ | 0.2234 | 0.0159 | 0.0038 |
| Position $x - axis$ $(m)$ | 0.1153 | 0.0394 | 0.0231 |
| Position $y - axis$ $(m)$ | 0.1063 | 0.1416 | 0.0379 |
| $V_x$ $(m/s)$ | 0.0767 | 0.0115 | 0.0094 |
| $V_y$ $(m/s)$ | 0.2064 | 0.0185 | 0.0168 |

1. **MIMO position control loop**. The proposed controller was employed to regulate the $xy$-positions of the simulated over-actuated hexacopter. The performance of tracking a sine wave along the $xy$-axes is presented in Fig. 4.13. It is shown that our proposed control system has closely followed the trajectory with higher accuracy compared to the other two benchmark controllers. Moreover, the overall performance of the closed-loop control system in 2-D and 3-D planes can be seen in Fig. 4.14, where lower RMSE values were obtained as tabulated in Table 4.2.

2. **MIMO velocity control loop**. The proposed SAF2C controller was implemented to control the velocity of an over-actuated hexacopter plant in the $x$-axis ($v_x$) and the $y$-axis ($v_y$). The performance was tested using a trajectory of a sine wave as shown in Fig. 4.15. In both cases, better tracking accuracy was observed from our proposed controller than that of the other two benchmark controllers. Lastly, the RMSE values were calculated in Table 4.2.

### 4.4.5 Computational time of different type-reduction methods

We compared the computational time of three different TR algorithms including KM, NT, and EIASC in Table 4.3. The computational costs for every algorithm were evaluated by

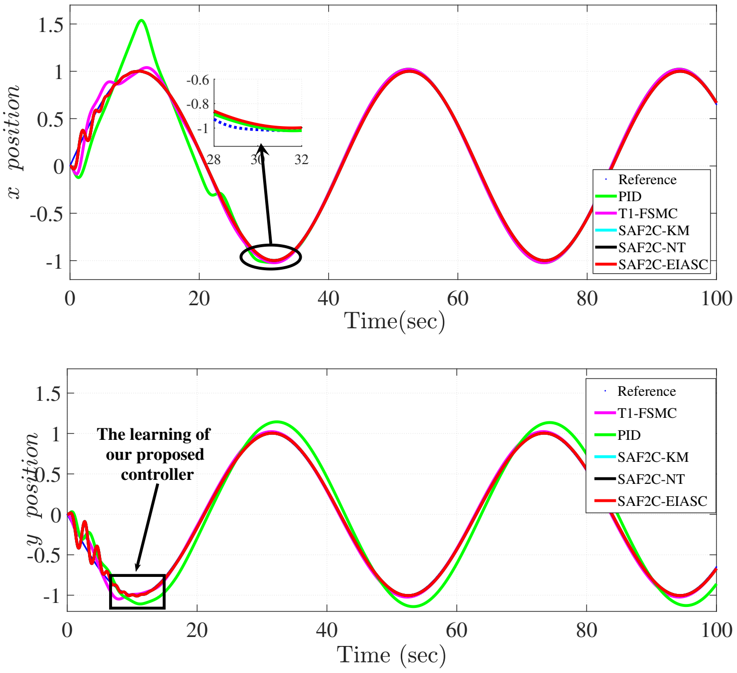


Figure 4.13: Sine wave tracking along the $xy$-axes.

the computation time, which was collected using the *tic* and *toc* MATLAB functions. It is observed that the EIASC algorithm outperformed the KM and NT algorithms. It can be also reported that all TR algorithms converge to the correct solution in a finite number of iterations, and these outcomes match up with the results from a similar study in [22].

### 4.4.6 Measurement Noise rejection performance

Another important consideration is to examine our proposed controller in the presence of measurement noise. We performed several experiments in which the actual output of the nonlinear plant was corrupted with different levels of noise power $(W)$. The noise rejection capability was investigated in our simulated hexacopter system. The RMSE values were plotted against different levels of noise power as shown in Figs. 4.16. The performance of our proposed controller was validated using different percentages of the *FoUs* and compared with the T1-SMC controller. As can be seen in Table. 4.4, at 0% *FoU*,

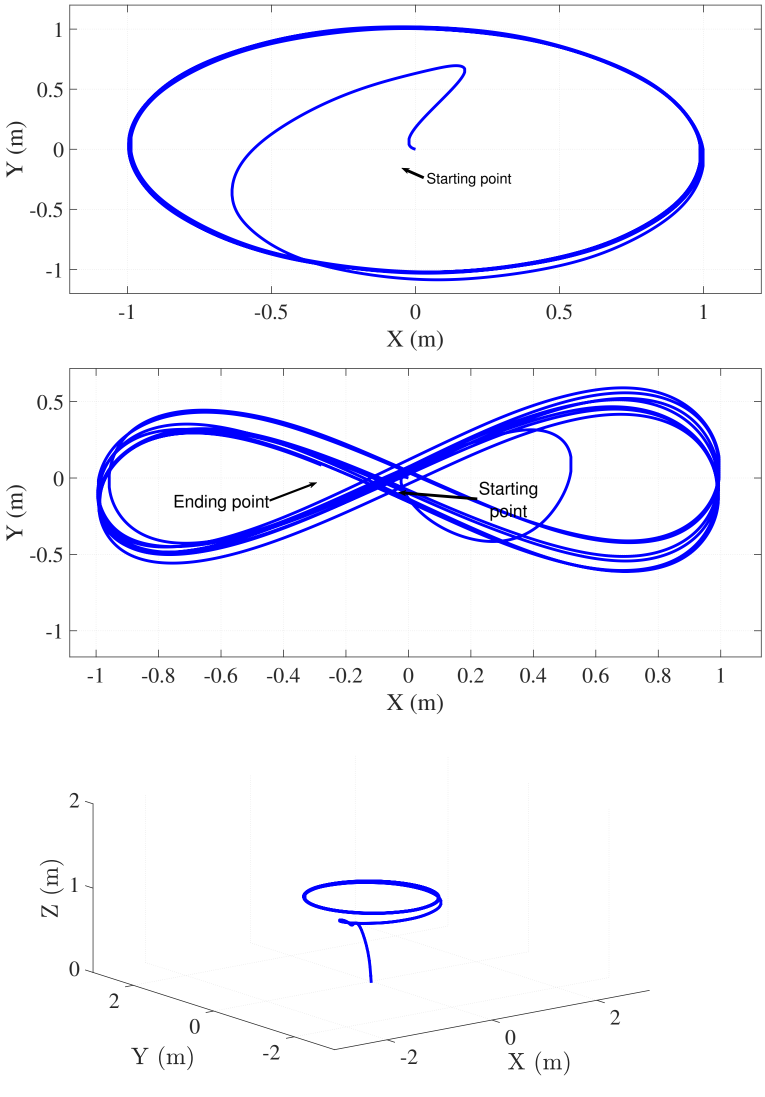


Figure 4.14: Trajectory tracking in 2D and 3D planes: top and middle figures depict the trajectory tracking along the $xy$ plane with different shapes, bottom figure shows complete 3D trajectory tracking.

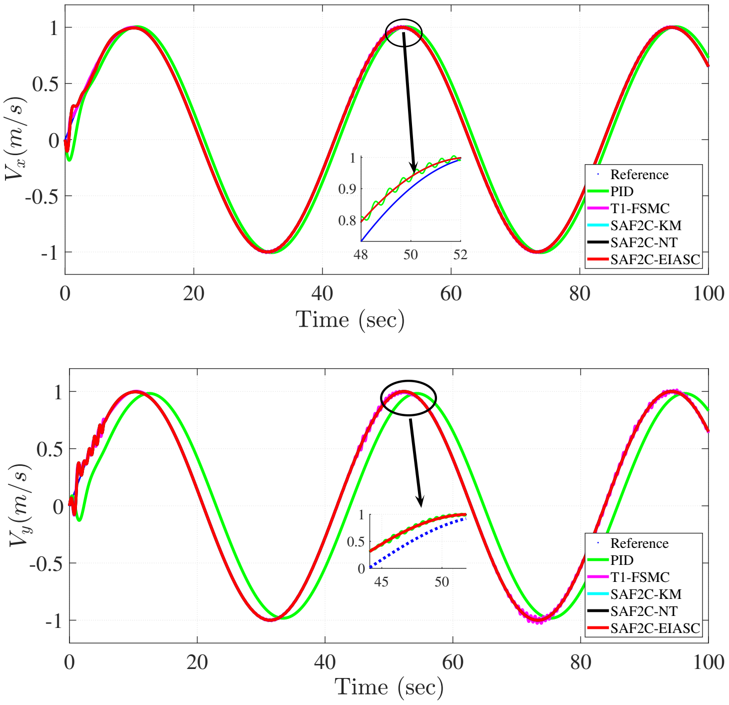


Figure 4.15: Velocity tracking in the $xy$-axes using the proposed controller.

Table 4.3: Computational time and RMSE values of different type-reductions.

| | TR | Nominal condition | with disturbances |
|---|---|---|---|
| Computation time (s) | NT | 0.8489 | 0.8918 |
| | KM | 0.9415 | 0.9825 |
| | EIASC | **0.7399** | **0.7704** |
| RMSE (m) | NT | 0.0020 | 0.038 |
| | KM | 0.0020 | 0.038 |
| | EIASC | 0.0020 | 0.038 |

Table 4.4: RMSE values with different *FoUs*.

| **Noise power (W)** | RMSE VALUES WITH MEASUREMENT NOISE | | | | |
|---|---|---|---|---|---|
| | *T1-SMC* | *SAF2C - 0% FoU* | *SAF2C - 25% FoU* | *SAF2C - 35% FoU* | *SAF2C - 50% FoU* |
| 0.0001 | 0.0489 | 0.0489 | 0.0235 | 0.0234 | **0.0229** |
| 0.001 | 0.0516 | 0.0516 | 0.0288 | 0.0297 | **0.0269** |
| 0.01 | 0.2780 | 0.2780 | 0.2744 | **0.2440** | 0.2486 |
| 0.1 | 1.9344 | 1.9344 | 2.0823 | **1.2984** | 1.3526 |

the RMSE value of our SAF2C is smaller than the one of T1-SMC especially at higher FoUs. For example, the best performance was obtained using the *50% FoU* to handle small noise power, while the *35% FoU* was sufficient to produce the lowest RMSE values for handling higher noise powers. However, for both cases, our SAF2C outperforms the T1-SMC system as summarized in Table 4.4 and Fig. 4.16.

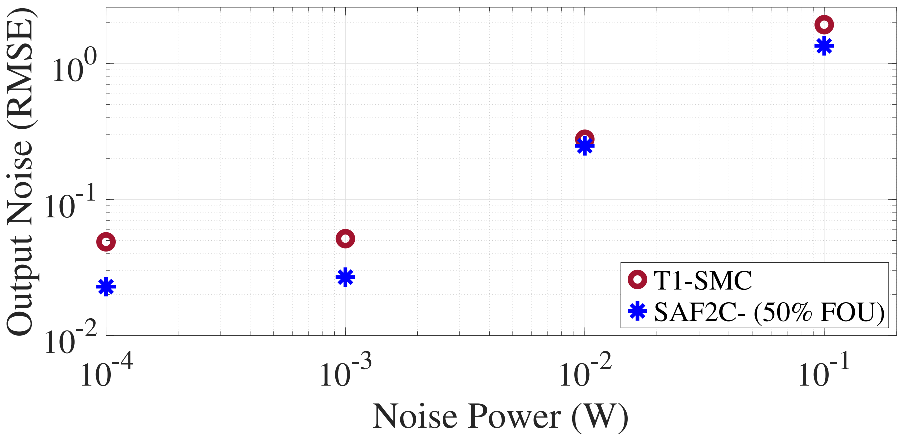


Figure 4.16: Noise rejection capability of SAF2C relative to T1-SMC.

### 4.4.7 Statistical Analysis

The parameters of our SAF2C were initialized with zero. For the purpose of statistical analysis, we ran our simulations 10 times to get the mean and the average standard deviation of the tracking error values. We initialized our SAF2C with random values in the interval [0,1] and we ran the simulation 10 times, whilst collecting the MIMO tracking errors. The average standard deviations of $x_1$ and $x_2$ were close to zero [0.00094728($m$) and 0.00026268($m$)], while the mean values were [0.0044($m$) and 0.0023($m$)], respectively. This illustrates that our SAF2C has the learning capability to minimize the tracking error of nonlinear MIMO systems.

### 4.4.8 Discussion

The simulations were performed using the MATLAB/SIMULINK environment with a time step of 0.02 *sec*. A trapezoidal interval type-2 fuzzy membership function was utilized in this design for input fuzzification. It was assumed that the trapezoidal membership function is symmetric so it can be represented by 6 points instead of the 9-points representation as in [266]. This reduction minimizes the number of variables employed. Different levels of *FoUs* were performed and compared with a T1-SMC to investigate the effects of measurement noise. Moreover, the execution time of the three different TR approaches was recorded.

The execution time of the proposed MIMO controller was reduced by 80% compared to the SISO controller. The SISO and MIMO control algorithms were executed utilizing a Core i7 Processor @ 3.4GHz, with 16 GB of RAM. The controller parameters are given as follows:

- For the altitude control loop, we used a learning rate $\mathit{\Gamma} = 0.5$ and a sliding surface gain $\gamma = 0.55$.
- For MIMO system in example 1, we used a learning rate $\mathit{\Gamma} = 0.255, 0.5$ and a sliding surface gain $\gamma = 5, 2$ for $x_1$ and $x_2$ respectively.
- For the position control loop, we used a learning rate $\mathit{\Gamma} = 0.07, 0.05$ and a sliding surface gain $\gamma = 0.7, 0.9$ for $x$ and $y$, respectively.

- For the velocity control loop, we used a learning rate $\mathit{\Gamma} = 0.007, 0.005$ and a sliding surface gain $\gamma = 0.7, 0.2$ for $v_x$ and $v_y$, respectively.

The choice of those parameters is based on Section III. The proposed controller utilizes only four rules with only two membership functions, which is computationally efficient. Since our SAF2C consequent parameters start learning from scratch, it is expected to have a slower transient response compared to fixed controllers as shown in Fig. 4 (a) and Fig. 6 (a). Thus, one way to improve the transient response is by initializing fuzzy parameters using bootstrapped values from previous experiments in order to improve the learning process, which leads to an improvement in the transient response.

Lastly, simulation results illustrate the effectiveness of our proposed SAF2C system compared to the other two benchmark controllers, especially in the face of uncertainties such as external disturbances. Although our proposed controller has several advantages, some of its parameters need to be properly tuned, including the learning rate $\mathit{\Gamma}$ and the sliding surface gain $\gamma$. These two parameters are important to ensure both stability and performance of the closed-loop control system.

## 4.5 Application to Inverted Pendulum on a Cart System

### 4.5.1 Overview

An inverted pendulum is an unstable nonlinear system. It is also an under-actuated system such that the number of control inputs is less than the degrees of freedom [267]. Therefore, controlling such system is a challenging task. Inverted pendulums are a convenient benchmark for testing algorithms and comparing the performance of different contemporary and classical control approaches. The purpose of this study is to derive a control law that can stabilize the pendulum so that it can maintain the desired vertical position while moving the cart to the desired position quickly and accurately [268]. The contribution of this research work can be demonstrated by accommodating two variables of interest, namely, the pendulum angle $\theta$ and the position of the cart $x$. To mitigate the manual tuning of fuzzy parameters, the consequent parts are tuned using the SMC theory. Moreover, the EIASC algorithm for type-reduction to reduce the computational cost is deployed.

### 4.5.2 Nonlinear Mathematical Model of an Inverted Pendulum on a Cart

The inverted pendulum plant can be demonstrated in Fig. 4.17, where $x$ represents the position of the cart and $\theta$ is the angle of the pendulum. From [268], the nonlinear model

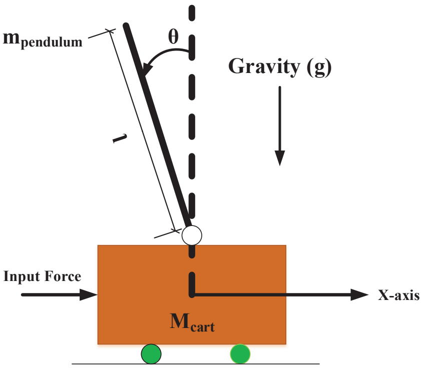


Figure 4.17: Schematic representation of an inverted pendulum on a cart

can be represented in a state space form as follows:

$$\frac{dx}{dt} = f(x, F, t) \tag{4.34}$$

The state variables can be written as:

$$x_1 = \theta; x_2 = \dot{\theta} = \dot{x}_1; x_3 = x; x_4 = \dot{x} = \dot{x}_3 \tag{4.35}$$

The final state space representation of an inverted pendulum on a cart can be written as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} x_2 \\ f_1(x_1, x_2) \\ x_4 \\ f_2(x_1, x_2) \end{bmatrix} \tag{4.36}$$

$$y = Cx = \begin{bmatrix} \theta \\ x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} \tag{4.37}$$

$$\begin{aligned} f_1(x_1, x_2) &= \frac{\left[F cos(x_1) - (m_{pend} + m_{cart}) g sin(x_1) + m_{pend} l cos(x_1) sin(x_1) x_2^2\right]}{a} \\ f_2(x_1, x_2) &= \frac{\left[F + m_{pend} l sin(x_1) x_2^2 - m g cos(x_1) sin(x_1)\right]}{b} \end{aligned}$$

where

$$\begin{aligned} a &= m_{pend} l cos^2(x_1) - (m_{pend} + m_{cart}) l \\ b &= m_{pend} + m_{cart} - m_{pend} cos^2(x_1) \end{aligned}$$

where the pendulum angle from vertical direction is $\theta = x_1$; the pendulum angular velocity is $x_2 = \dot{\theta} = \dot{x}_1$; the cart position is $x_3 = x$; the cart angular velocity is $x_4 = \dot{x} = \dot{x}_3$; the pendulum length is $l = 1m$; the force on the cart in the $x$ direction is $F$ in $(N)$; the mass of the cart $m_{cart}$ and mass of the pendulum $m_{pen}$ are $0.1kg$ and $1kg$ respectively; and the gravity $g = 9.8\frac{m}{s^2}$.

### 4.5.3 Results and Discussion

#### 4.5.3.1 Controller Design

The objective of the proposed controller is to balance an inverted pendulum on a cart where the pendulum should be in the upright position while driving the cart to the desired position. Similar to the previous section, the adaptive IT2FLCs utilizes a trapezoidal membership function with four rules.

In this proposed controller, the consequent parameters of the four fuzzy rules are initialized with zeros. The adaptation law for the type-2 consequent parameters of the proposed controller is based on sliding mode control theory [269]. This method can guarantee the system's robustness against parameter variations and unknown uncertainties. The sliding surface to control an inverted pendulum system can be expressed as $s_{surface} = \gamma e + \dot{e}$, where $e$ is the error and $\dot{e}$ is the derivative of error. The slope of the sliding surface is represented by $\gamma > 0$. Hence, the new parameters of the premise part can be expressed as, $C_{new} = C_{old} + \eta * S_{surface}$, where $\eta$ is the learning rate and it is set in such a way that the sliding surface parameters can achieve the optimal values of the premise fuzzy part in a short time.

An adaptive IT2FLCs is proposed to control both the pendulum angle $\theta$ and also the cart position $x$. The purpose of the control law is to maintain the pendulum in the upright position where $\theta = 0$. The control law is designed to control a nonlinear inverted pendulum in different scenarios. First, it was tested under nominal condition. Second, the proposed controller was tested against disturbance rejection and lastly the system was examined against random noise rejection. To evaluate the performance of the proposed interval type-2 controller, a comparison with a PID controller is drawn. Tables 4.5 & 4.6 show the performance of the proposed controller versus a PID controller under the three different scenarios where lower RMSE values are achieved from the proposed type-2 fuzzy controller. To obtain satisfactory results against disturbance and random noise, the SMC learning theory is adopted where the consequent parameters of the proposed controller are updated recursively.

#### 4.5.3.2 Simulation results under nominal conditions

The performance of the proposed controller was evaluated and compared with a conventional PID controller under nominal conditions. The responses of cart position and cart velocity are shown in Fig. 4.18 & 4.19 respectively where a faster convergence is observed using the proposed type-2 controller. Similarly, Fig. 4.20 demonstrates the responses of the pendulum angle where the response of the pendulum angle is better for the proposed controller. The control signal for the proposed controller can be depicted in Fig. 4.21.

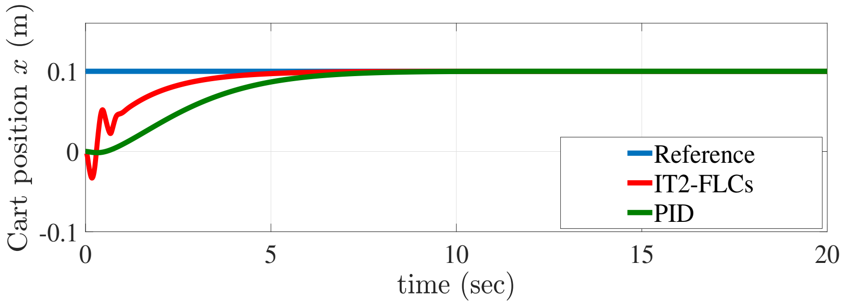


Figure 4.18: Responses of cart position $x$.

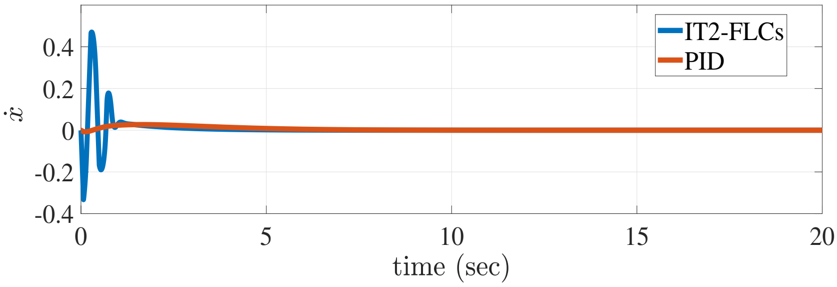


Figure 4.19: Response of cart velocity $\dot{x}$.

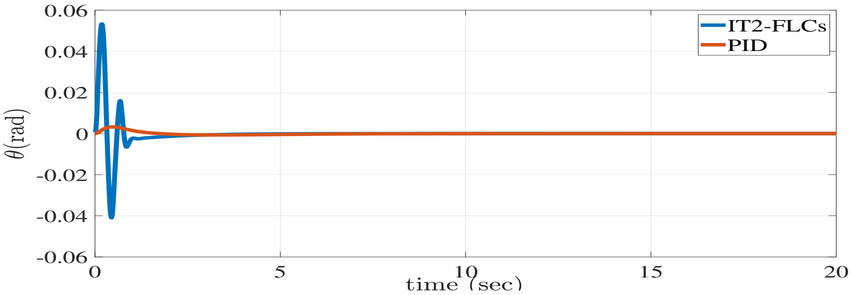


Figure 4.20: Response of pendulum angle $\theta$.

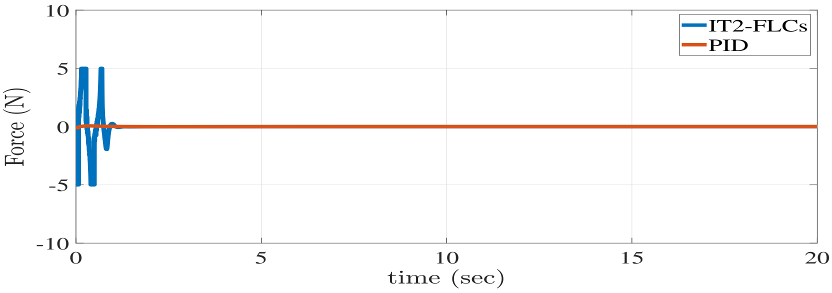


Figure 4.21: Control force (N).

#### 4.5.3.3 Simulation results in the face of random noise

To examine the robustness of the proposed controller against noise rejection, a band-limited white noise signal is added to the system as a disturbance input. The disturbance signal parameters are as follows: sampling time=0.01, and the power of the band-limited white noise = 0.001 [268]. The control input is bounded between $[-5, 5]$. From Fig. 4.22 and Fig. 4.23, it is observed that the pendulum stabilizes in the vertical position with minor oscillation in the case of type-2 fuzzy control while it stabilizes with major oscillation for the case of the PID controller. The position of the cart reached the specified position $(0.1m)$ smoothly for the case of type-2 fuzzy while it has reached the desired position with oscillations for the case of the PID controller.

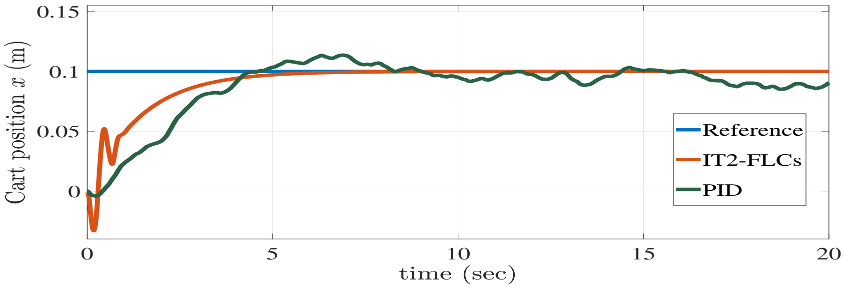


Figure 4.22: Responses of Cart position $x$ with respect to random noise.

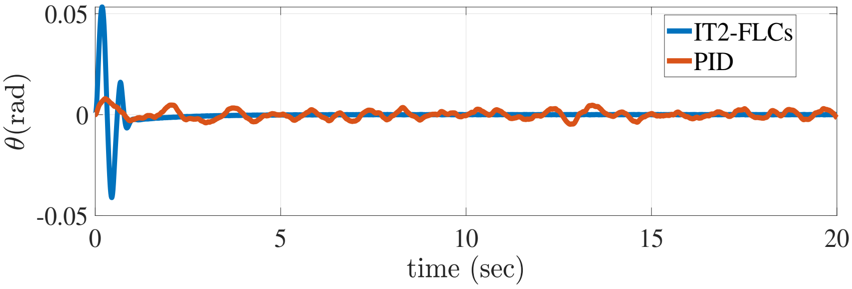


Figure 4.23: Response of pendulum angle $\theta$ with random noise.

#### 4.5.3.4 Simulation results in the face of external disturbance

Another measurement to study the robustness analysis of the proposed controller is to test it against external disturbance. The external disturbance (as shown in Fig. 4.26) is added at time 7 sec to the deflection angle of the pendulum. The performance of the pendulum angle and the cart position in the presence of external disturbance are shown in Fig. 4.24 & 4.25 respectively. It can be seen that our proposed controller is more robust than the PID controller with lower RMSE values for maintaining the pendulum angle in the vertical position.

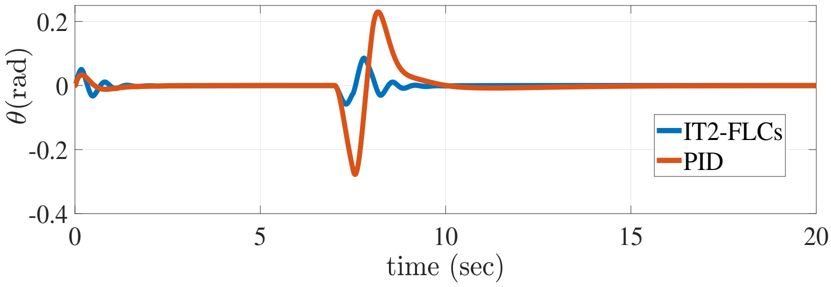


Figure 4.24: Response of pendulum angle $\theta$ with with disturbance.

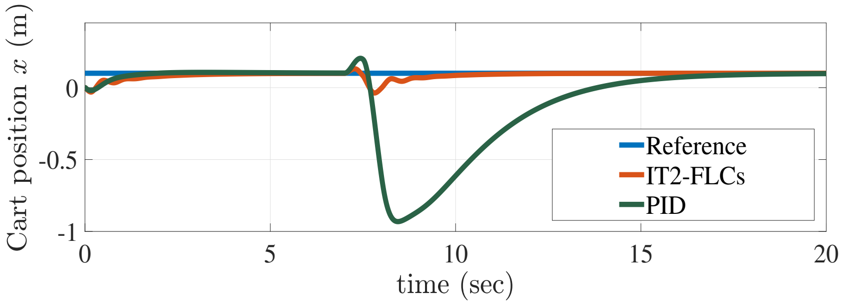


Figure 4.25: Responses of cart position $x$ with respect to disturbance

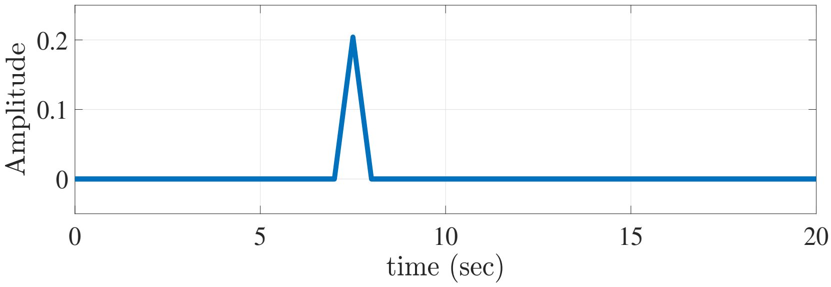


Figure 4.26: Disturbance signal.

Table 4.5: IT2FLCs and PD Controllers performance on pendulum angle control (RMSE).

| Controller | RMSE | | |
|---|---|---|---|
| | Nominal condition | Random Noise | External disturbance |
| IT2FLCs | 0.000268 | 0.0114 | 0.000203 |
| PID | 0.0012 | 0.0864 | 0.0018 |

Table 4.6: IT2FLCs and PD Controllers performance on cart position control (RMSE)

| Controller | RMSE | | |
|---|---|---|---|
| | Nominal condition | Random Noise | External disturbance |
| IT2FLCs | 0.0268 | 0.0309 | 0.0303 |
| PID | 0.0397 | 0.0373 | 0.3383 |

## 4.6 Self-Adaptive IT2FLCs for AUVs

### 4.6.1 Overview

Controlling AUVs is a challenging task due to their nonlinear and time-varying dynamics. Recently, there has been a growing interest in intelligent control techniques for various applications due to their capability to achieve autopilot control systems. Fuzzy and neural networks have been implemented to control multiple robotic platforms [3, 108, 202, 251, 270, 271]. The efficacy of fuzzy control systems to accommodate uncertainties and nonlinearities has been established beyond doubt. For instance, type-1 fuzzy logic control was utilized in [206], [253], [252] where good performance was obtained. However, most of the existing fuzzy controllers are type-1 fuzzy. As such, implementing type-2 fuzzy to control AUVs is a new research scheme.

Due to the necessity to accommodate uncertainties in the AUV control design, there has been a growing interest in utilizing type-2 fuzzy logic control [30, 34, 67]. Interval type-2 fuzzy was developed in [272] and [273] for path planning and control with obstacle avoidance. However, manual tuning of fuzzy control is a tedious task [29], [114]. Hence, there is a need for automatic tuning of fuzzy parameters. In this work, a nonlinear and time-varying autonomous underwater vehicle model is first developed. Second, a self-adaptive IT2-FLC is proposed, where the consequent parameters are tuned using the

SMC technique. Third, the robustness of the proposed control technique is investigated.

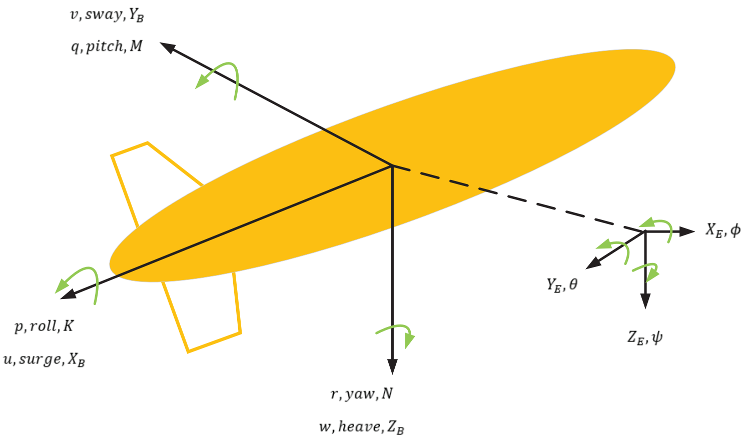


Figure 4.27: AUV six degree of freedom

### 4.6.2 AUVs nonlinear dynamics

The AUV equations of motion can be expressed with respect to a fixed frame, $(X_E, Y_E, Z_E)$, or moving body reference frame $(X_B, Y_B, Z_B)$ as shown in Fig. 4.27. The mapping between coordinate systems can be represented using the velocity transformation as follows:

$$\dot{\eta} = J(\eta)\nu \tag{4.38}$$

where $\eta$ depicts the position and orientation vector of the vehicle defined in the fixed frame, which can given as:

$$\eta = [x \ y \ z \ \phi \ \theta \ \psi]^T , \tag{4.39}$$

$\nu$ is the linear and angular velocity vector, which can be described as follows:

$$\nu = [p \ q \ r \ u \ v \ w]^T \tag{4.40}$$

and $J(\eta)$ denotes the Euler angle mapping matrix [274].

The equations of motions with respect to the earth fixed reference via the kinematics transformations in terms of position and attitude as follows [274–276]:

$$M_\eta(\eta)(\ddot{\eta}) + C_\eta(\eta, \dot{\eta})\dot{\eta} + D_\eta(\eta, \dot{\eta})\dot{\eta} + G_\eta(\eta) = \tau_\eta, \tag{4.41}$$

where

$$\begin{cases} M_\eta(\eta)(\ddot{\eta}) & = & J^{-T}(\eta)MJ^{-1}(\eta) \\ C_\eta(\eta,\dot{\eta}) & = & J^{-T}(\eta)[C(\nu) - MJ^{-1}(\eta)\dot{J}(\eta)]J^{-1}(\eta) \\ D_\eta(\eta,\dot{\eta}) & = & J^{-T}(\eta)D_\eta J^{-1}(\eta) \\ G_\eta(\eta) & = & J^{-T}(\eta)G_\eta \\ \tau_\eta(\eta) & = & J^{-T}(\eta)\tau \end{cases}$$

where $M_\eta(\eta)$ is the inertia matrix, which is a symmetric and positive definite (6x6) matrix; $C_\eta(\eta,\dot{\eta})$ is a matrix that describes the Coriolis forces; $D_\eta(\eta,\dot{\eta})$ is the matrix of the hydrodynamic damping term; $G(\eta)$ is a (6x1) matrix, which represents the vehicle's gravitational forces and moments; and $\tau$ denotes the control inputs. The detailed mathematical model can be found in [253, 275].

### 4.6.3 Results and Discussion

#### 4.6.3.1 Controller Design

Similar to previous sections, this study deploys the SMC theory for adapting fuzzy consequent parameters. The overall closed-loop control system for the AUV is depicted in Fig. 4.28. The sliding surface ($s_{AUV}$) for an AUV is designed as, $s_{AUV} = u_f = \gamma e + \frac{d}{dt}e$, where $e = \eta - \eta_d$, $\eta$ is the actual output obtained from (4.39), $\eta_d$ is the desired trajectory and $\gamma$ is a positive constant, $u_f$ is the control signal from the fuzzy controller.

The feedback control mechanism can be seen in Fig. 4.29. The objective function is to minimize the error between the desired input and the actual output. For the attitude control of an AUV, three controllers are designed for the surge, yaw and pitch controls. By utilizing the SMC theory combined with a PD controller [277], the adaptive control structure of the proposed controller is depicted in Fig. 4.28. Hence, the control law can be defined as, $u = u_{pd} - u_f$, where $u_{pd} = k_p e + k_d\ \dot{e}$ and $[k_p, k_d]$ are the controller gains.

The error and the change of error signal are the two input signals to the fuzzy inference

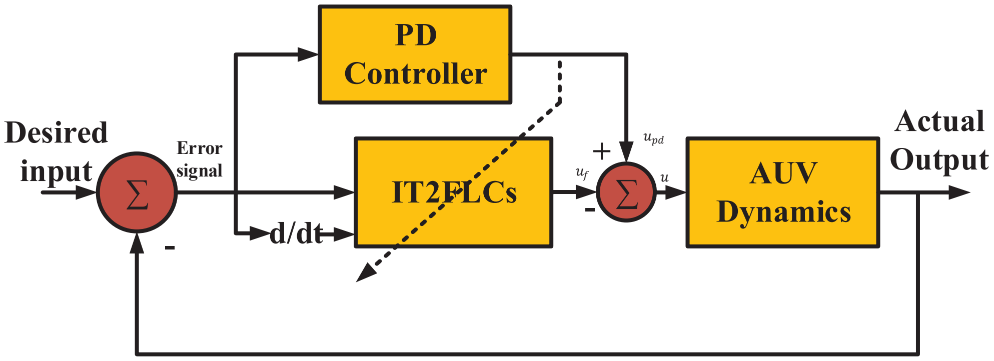


Figure 4.28: Closed-loop control system for AUV using adaptive IT2FLCs.

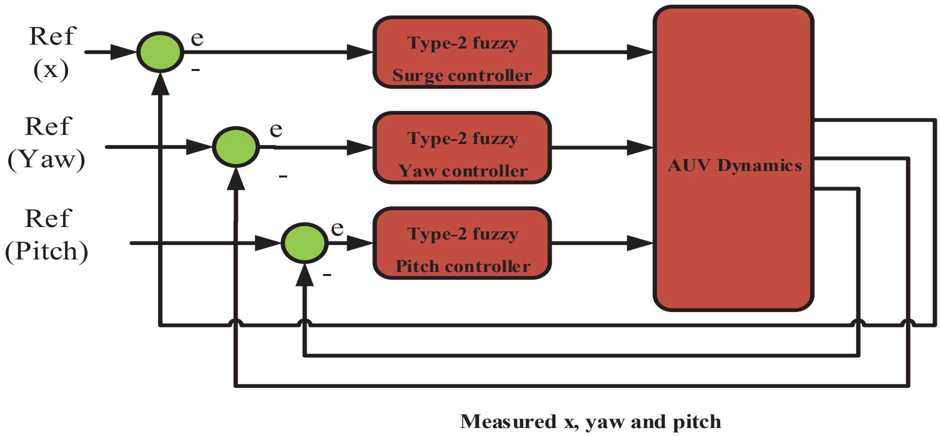

Ref (x)
Ref (Yaw)
Ref (Pitch)
e
-
Type-2 fuzzy Surge controller
Type-2 fuzzy Yaw controller
Type-2 fuzzy Pitch controller
AUV Dynamics
Measured x, yaw and pitch

Figure 4.29: Proposed IT2FLCs desgin demonstrating surge, yaw, and pitch dynamics of an AUV.

system. The purpose of the developed control law is designed so that the AUV follows the desired trajectory. Surge, pitch and yaw controllers are designed to maintain the depth of the AUV and to determine the AUV orientation in three dimensions. The controller performance is tested in a nominal case and also in the presence of disturbance.

#### 4.6.3.2 System performance in a nominal condition

The performance of the proposed adaptive IT2-FLC is evaluated in a nominal condition. It is compared with a fixed IT2-FLC. The designed fuzzy control system has four rules and two membership functions for each input. Figures 4.30 & 4.31 show the simulation results of the 6-DOF AUV response in $XY$-plane and $XZ$-plane, where good tracking is obtained. The pitch and yaw responses can be seen in Figs. 4.32 and 4.33, where a successful tracking and a reasonably shorter settling time were obtained. The RMSE values are reduced significantly by accommodating the SMC theory for tuning fuzzy consequent parameters as shown in Table 4.7.

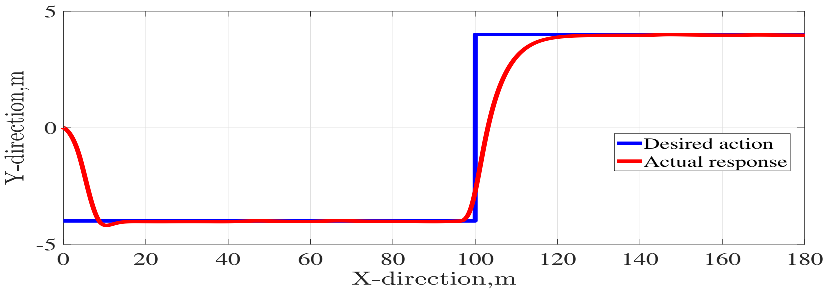


Figure 4.30: The motion of AUV in XY plane using our proposed controller

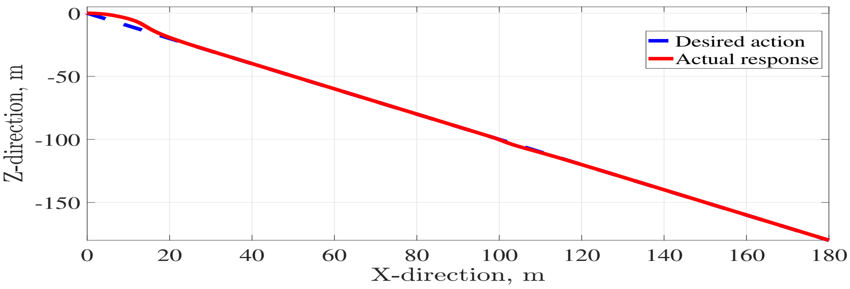


Figure 4.31: The motion of AUV in XZ plane using our proposed controller

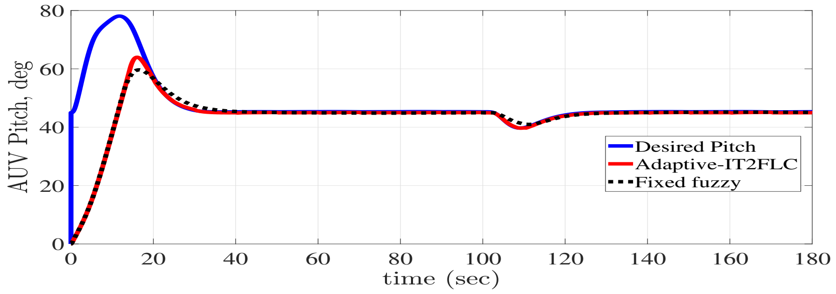


Figure 4.32: Pitch response

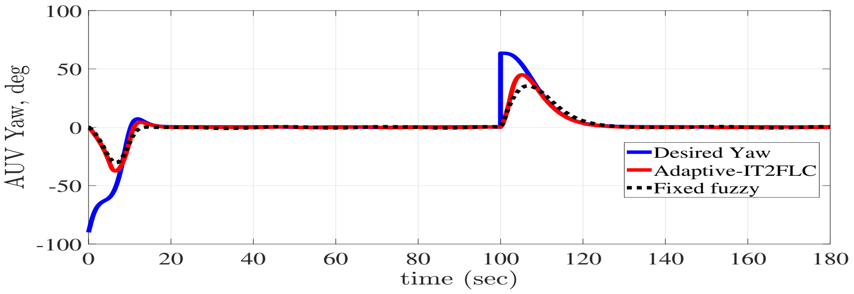


Figure 4.33: Yaw response

#### 4.6.3.3 System performance in the face of disturbance

In order to investigate the robustness of the proposed controller, an external disturbance was added between $[60, 120]$ sec. As can be seen in Figs. 4.34 & 4.35, the proposed controller can cancel it within a reasonable period of time. The comparative RMSE values of the system are presented in Table 4.7.

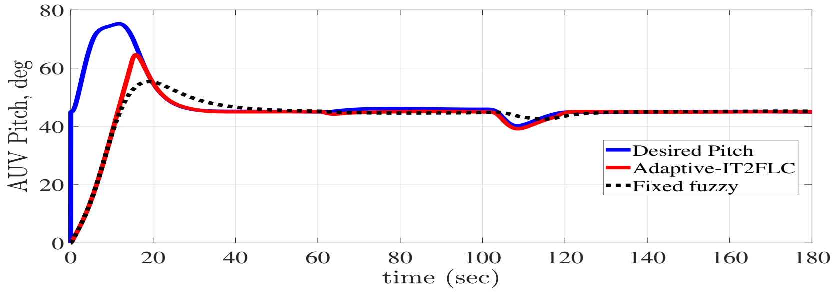


Figure 4.34: Pitch response with disturbance

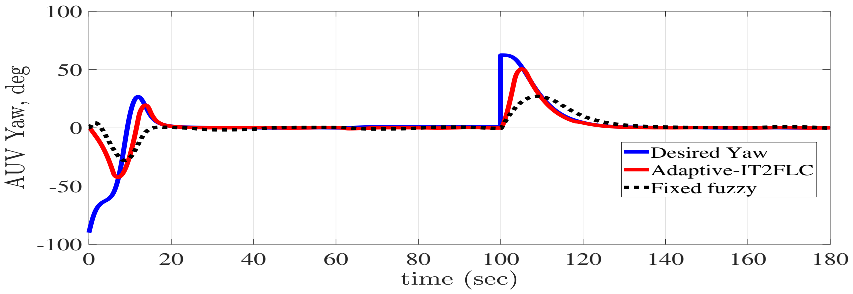


Figure 4.35: Yaw response with disturbance

Table 4.7: Adaptive IT2-FLC vs a fixed IT2-FLC performance (normalized-RMSE)

| **RMSE Values** | Angles (Nominal condition) | | Angles (with disturbance) | |
|---|---|---|---|---|
| | Pitch | Yaw | Pitch | Yaw |
| Fixed IT2-FLC | 0.403 | 0.591 | 1.204 | 1.756 |
| Adaptive IT2-FLC | 0.101 | 0.179 | 0.565 | 0.698 |

## 4.7 Summary

The proposed SAF2C controller has several advantages for accurate trajectory tracking, including its stability, learning ability and the extension to MIMO systems. Moreover, utilizing the SMC theory to adapt fuzzy parameters can improve its robustness against various uncertainties (e.g. disturbance, and wind/gust). Simulation results have demonstrated the effectiveness of our proposed fuzzy controller in regulating MIMO systems. The results of the proposed controller were compared with the type-1 counterpart and the conventional PID controller, where a 20% improvement relative to a type-1 fuzzy system was obtained and a 50% enhancement with respect to a PID controller was achieved. We also implemented three different TR methods and calculated their computation time in the existence of measurement noise.

The proposed MIMO controller is computationally efficient, where the execution time was reduced by 80%. Moreover, the proposed controller employed only four linguistic rules with only two membership functions, which is computationally efficient. This efficiency makes it more suitable for real-time experiments. Lastly, our proposed controller demonstrated better robustness against uncertainties, namely, external disturbance and wind gusts, where superior disturbance rejection was shown through extensive computer simulations compared to the benchmark type-1 fuzzy and PID controller.

Secondly, for the inverted pendulum, the proposed closed-loop control design was developed so that the pendulum was maintained in the upright position. Simulation results of the proposed controller were promising compared to the conventional PID controller. Besides, the RMSE values were also minimized.

Thirdly, for the AUV, an adaptive IT2FLCs was designed to control the position and attitude, namely, surge, pitch, and yaw motions. The tracking performance of the proposed control system was compared with a fixed fuzzy controller, where better results were observed from the adaptive IT2FLC system with smaller RMSE values.

# Chapter 5

# Enhanced Self-Adaptive Interval Type-2 Fuzzy System (ESAF2C) for Aerial Robotics: Control Design and Real-Time Flight Tests

The work illustrated in this chapter has been submitted to the following journal:

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). Self-Learning in Aerial Robotics Using Type-2 Fuzzy Systems: Case Study in Hovering Quadrotor Flight Control. *IEEE Access* [278].

## 5.1 Introduction

Drones, a prominent nickname for unmanned aerial vehicles (UAVs), have been attracting a large amount of consideration in the last few decades. They have been used in a myriad of applications (e.g. transportation [180], inspection of power transmission lines [181], search and rescue missions [182], and collecting traffic information [183]). One major benefit of multirotor UAVs is their capability to hover, to perform vertical take-off and landing (VTOL), and to fly in confined spaces [14, 60, 187].

The classical model-based control approaches for UAVs, such as PID controller [116], LQR [117], and MPC [118] can provide optimal control performance when the model is well-defined, precise, and there are no uncertainties. Nevertheless, there are inevitable uncertainties in aerial robots, (e.g., lack of modeling, mechanical wear, rotor damage, battery drain and sensor and actuator faults [25, 119]). Another aerodynamic challenge when flying a rotorcraft vehicle at a reasonably low altitude is the ground effect, which occurs due to the blockage and distortion of the rotor downwash by the ground beneath a hovering UAV [279, 280]. The ground effect brings significant nonlinearity (altering the thrust characteristic) and introducing uncertainties into the closed-loop control system [280]. As such, the control performance of conventional model-based methods may deteriorate in the face of uncertainty. [23, 30, 121, 122, 281].

As a recap, Takagi-Sugeno fuzzy logic control systems allow a description of a nonlinear system via a set of local linear system domains via relevant membership functions [125]. Nevertheless, since uncertainty is not incorporated in the membership function of T1-FSs, it seems contradict with the definition of "fuzzy" itself. As such, T1-FSs have limited ability to handle uncertainties [126]. Hence, IT2FSs were proposed in [15] based on type-2 fuzzy sets. Although there are several studies of multirotor drone control based FLCs, most of these studies are based on T1-FLCs.

Since IT2FLCs provide an extra degree of freedom to handle the footprint-of-uncertainties, they have been applied for controlling quadcopter UAVs. In [74], IT2FLCs were proposed for quadcopter altitude control, where good results were achieved. Nevertheless, the fuzzy parameters were tuned manually. Manual tuning of FLCs can be a time-consuming, inefficient, and tedious task [29]. There are several studies where IT2FLCs were combined with sliding mode control (SMC), resulting in improvement of the overall control performance by canceling the chattering effect on SMC control systems [282]. In [29], an IT2FLC based SMC theory was proposed for a quadcopter UAV (QUAV), achieving reasonably good control performance compared to a T1-FLC and conventional PID controllers. However, the fuzzy control law was designed as a combination of both an IT2FLC system and a PID controller, resulting in extra effort for tuning PID parameters. In the work by [107], a fault-tolerant control based on interval type-2 fuzzy neural networks and sliding mode controller was developed for a 6-DOF octocopter UAV. Although their proposed controller has the ability to guarantee the stability of the proposed control system, it lacks experimental validation. Furthermore, IT2FLC parameters are reduced to type-1 fuzzy sets, resulting in similar behavior to T1-FLCs, as the *FOU* is not explicitly included in the fuzzy design. Lastly, there is also a lack of real-time experiments using IT2FLCs.

The remainder of this chapter is structured as follows. The contributions are listed in Section 5.2. The dynamic model of the QUAV is provided in Section 5.3. The proposed closed-loop control system is explained in Section 5.4, including the stability analysis. This is followed by the computer simulation results illustrating the efficacy of the proposed control system in Section 5.5. Real-time flight test results are presented in Section 5.6 for a hovering quadcopter in the face of wind disturbance. Finally, Section 5.7 provides a summary of this chapter.

## 5.2 Contributions of this chapter

Motivated by the aforementioned research gaps, the contributions of this chapter can be highlighted as follows:

- A novel stand-alone enhanced self-adaptive interval type-2 fuzzy controller, named the *ESAF2C*, is proposed for position control of a hovering QUAV, whose type-2 fuzzy parameters are tuned online using the sliding mode control theory. Unlike most of the state-of-the-art work in the literature, the chattering phenomenon is eliminated by smoothing out the control discontinuity around the sliding surface. The ‘Enhanced Iterative Algorithm with Stop Condition’ (EIASC) type-reducer is accommodated in designing the ESAF2C, which is more suitable for real-time implementation than other type-reducers for its computational efficiency. Moreover, the proposed control law does not require augmentation by other control approaches making it a stand-alone controller.

- The robustness of the proposed controller is investigated in the face of external disturbances (e.g. ground effects, wind gust, measurement noise) and a rigorous comparative study is performed with respect to T1-FLC and its conventional counterpart. This research demonstrates the ability of our algorithm to improve disturbance rejection of the controller. Moreover, the proposed closed-loop control system proves its ability to filter measurement noise as investigated in the simulation section.
- Real-time flight tests are conducted for a hovering QUAV under stochastic wind disturbances to validate the efficacy of the theoretical claims. Specifically, the control performance in the face of external wind disturbance is investigated, using an industrial fan in the hover mode. The research findings show that the proposed control technique has the capability to learn its parameters in an online manner and to handle external stochastic wind gust disturbances better than its T1-FLC and conventional PID counterparts.
- The stability analysis of the proposed control system is investigated using the Lyapunov theory.

## 5.3 QUAV dynamic model

QUAVs have six degrees of freedom (DOF), with high mobility and four rotors. The motion of a QUAV can be expressed by the following twelve state variables, namely,

$$\vec{x} = [X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}, \theta, \phi, \psi, p, q, r]^T, \tag{5.1}$$

where $[X, Y, Z]^T$ represents the linear positions in the inertial frame $\{A\}$; $[\dot{X}, \dot{Y}, \dot{Z}]^T$ denotes the linear velocities across the $xyz$-axes; $[\theta, \phi, \psi]^T$ are the three Euler's angles, namely, the pitch, roll and yaw, respectively; $[p, q, r]^T$ denotes the angular rates in the

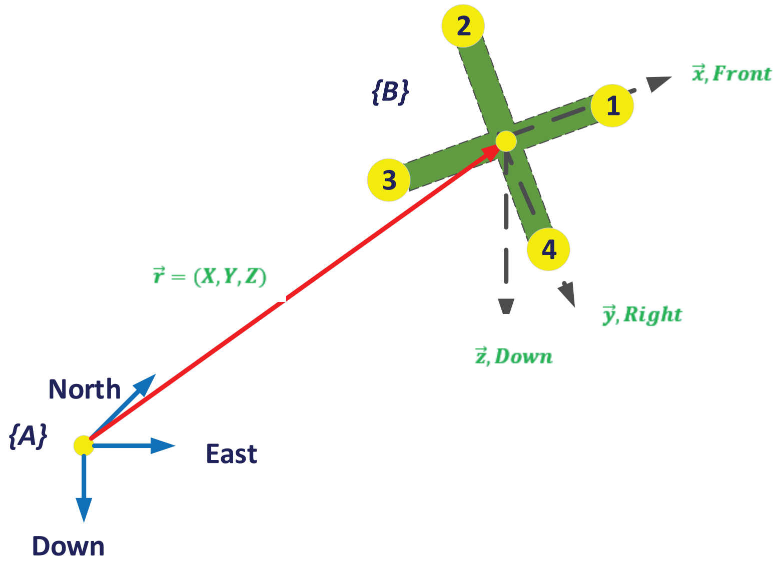


Figure 5.1: Coordinate frame of a QUAV.

body frame $\{B\}$ of the QUAV (see Fig. 5.1). The total thrust and moments for the four control inputs on each control axis can be described as follows:

$$\vec{U} = [U_1, U_2, U_3, U_4]^T = \left[T_{\sum}, M_1, M_2, M_3\right]^T, \tag{5.2}$$

where $\vec{U}$ represents control variables; $T_{\sum} = \sum_{i=1}^{N} T_i$ is the total sum of thrusts along $\{B\}$; and $M_{1,2,3}$ denote the moments generated by $N$ number of rotors [283]. In the case of QUAV, $N = 4$, hence the angular speed of each rotor can be described as:

$$\begin{bmatrix} T_{\sum} \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ l & 0 & -l & 0 \\ K_q/K_T & -K_q/K_T & K_q/K_T & -K_q/K_T \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}, \tag{5.3}$$

where $T_{1,2,3,4}$ are the thrust from each individual motor; $K_q$ $[kg.m^2]$ represents the lumped rotor torque coefficient; $K_T$ $[kg.m]$ denotes the lumped rotor thrust coefficient; and $l$ $[m]$

is the arm length. Finally, the equations of motion can be summarized as follows:

$$\begin{cases} \ddot{X} & = -(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\frac{T_\Sigma}{m} \\ \ddot{Y} & = -(-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\frac{T_\Sigma}{m} \\ \ddot{Z} & = g - (\cos\theta\cos\phi)\frac{T_\Sigma}{m} \\ \dot{p} & = \frac{I_{yy}-I_{zz}}{I_{xx}} \cdot qr + \frac{1}{I_{xx}} \cdot M_1 \\ \dot{q} & = \frac{I_{zz}-I_{xx}}{I_{yy}} \cdot rp + \frac{1}{I_{yy}} \cdot M_2 \\ \dot{r} & = \frac{I_{xx}-I_{yy}}{I_{zz}} \cdot pq + \frac{1}{I_{zz}} \cdot M_3, \end{cases} \tag{5.4}$$

where $I_{xx,yy,zz}$ represent the moment of inertia on the $xyz$-axes. The detailed physical parameters of the QUAV and the meaning of each symbol can be found in [283]. Fig. 5.2 illustrates the position control structure for the QUAV.

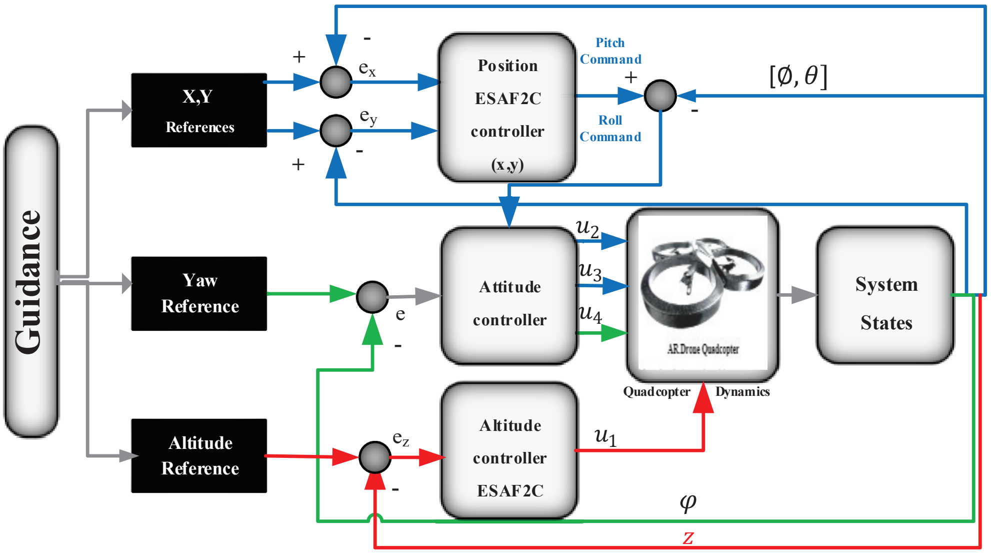


Figure 5.2: Control structure based on ESAF2C for position tracking of a nonlinear quadcopter plant, where we employ the attitudes and the thrust to create a position control loop outside the velocity loop of the Parrot AR.Drone.

## 5.4 ESAF2C Design

### 5.4.1 ESAF2C Structure

As a recap, the structure of IT2FLCs is constructed of four major elements, namely, fuzzifier, rule base fuzzy system, type reduction, and defuzzifier. First, the trapezoidal MFs were chosen for the fuzzification process to transform crisp input values into T2-FSs. The reason for choosing trapezoidal MFs is that their analytical structure is easy to derive [284]. Similar to the previous chapter, a symmetric trapezoidal IT2 FS was adopted for its simplicity, and it has been widely used in IT FSs [82,285]. In IT2FLCs, MFs provide a three-dimensional (3D) representation, which combines the upper membership function (UMF) and the lower membership function (LMF) as shown in Fig. 5.3. The upper and the lower IT2-trapezoidal MFs can be represented as follows [285]:

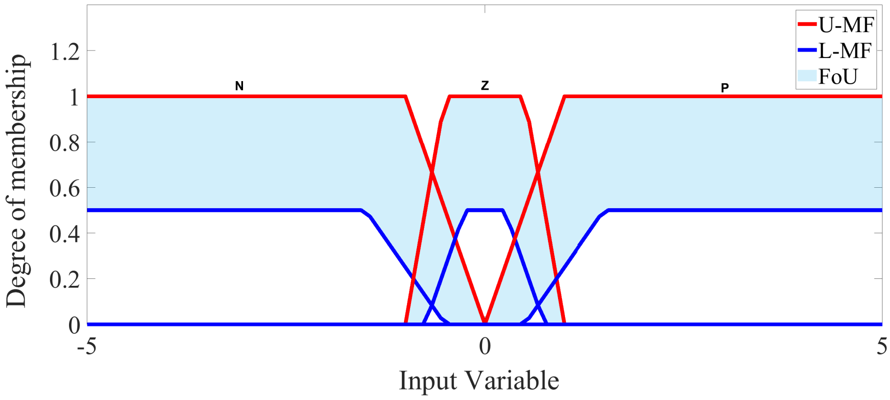


Figure 5.3: General interval type-2 trapezoidal membership function.

$$\underline{\mu}_{\tilde{F}}(x) = \begin{cases} 0, & x \leq \hat{m} - \hat{a} \; or \; x > \hat{m} + \hat{a} \\ w\frac{x-\hat{m}+\hat{a}}{\hat{a}-\hat{c}}, & \hat{m} - \hat{a} < x \leq \hat{m} - \hat{c} \\ w, & \hat{m} - \hat{c} < x \leq \hat{m} + \hat{c} \\ w\frac{\hat{m}+\hat{a}-x}{\hat{a}-\hat{c}} & \hat{m} + \hat{c} < x \leq \hat{m} + \hat{a} \end{cases} \tag{5.5}$$

$$\overline{\mu}_{\tilde{F}}(x) = \begin{cases} 0, & x \leq \hat{m} - \hat{b} \; or \; x > \hat{m} + \hat{b} \\ \frac{x - \hat{m} + \hat{b}}{\hat{b} - \hat{d}}, & \hat{m} - \hat{b} < x \leq \hat{m} - \hat{d} \\ 1, & \hat{m} - \hat{d} < x \leq \hat{m} + \hat{d} \\ \frac{\hat{m} + \hat{b} - x}{\hat{b} - \hat{d}} & \hat{m} + \hat{d} < x \leq \hat{m} + \hat{b} \end{cases} , \tag{5.6}$$

where $\hat{a},\ \hat{b},\ \hat{c},\ \hat{d}$ and $w$ are constrained. To simplify the design and for the sake of real-time implementation, we set $\hat{a} = \hat{b},\ \hat{c} = 0,\ w = 1$. Hence, there are only three free parameters to be tuned.

An IT2 FS $\tilde{F}$ is symmetric with respect to its centre $\hat{m}$ if, for $\forall x \in X$, its upper and lower MFs satisfy Eq. (5.7) [285].

$$\underline{\mu}_{\tilde{F}}(\hat{m} + x) = \underline{\mu}_{\tilde{F}}(\hat{m} - x), \qquad \overline{\mu}_{\tilde{F}}(\hat{m} + x) = \overline{\mu}_{\tilde{F}}(\hat{m} - x). \tag{5.7}$$

Second, the TS rule base fuzzy system was chosen in this proposed control design, having the following structure:

$$\begin{aligned} &IF\ e_1\ is\ \tilde{F}_1^i\ and\ e_2\ is\ \tilde{F}_2^i\ and\ e_n\ is\ \tilde{F}_{\substack{i\\n}}, \\ &THEN\ Y^i = [\upsilon_i^l\ \upsilon_i^r], \end{aligned}$$

where $e_n$ represents an input variable; $\tilde{F}_{ni}$ labels IT2-FSs antecedents; $Y^i$ denotes the output of the $i^{th}$ rule base; and $\upsilon_i^l$ and $\upsilon_i^r$ express the lower and upper consequent parameters, respectively. In this work, nine rules were utilized. The interpretation of these rules can be shown in Table 5.1-5.3, where $N$ means negative, $Z$ denotes zero, and $P$ represents positive.

Third, the firing level is represented by a firing interval. This process can be achieved using the product (t-*norm*) operation. The firing interval can be calculated as $F^i\ (x_1,\ ...,\ x_n) = [\underline{f}^i,\ \overline{f}^i]$, where $\underline{f}^i$ and $\overline{f}^i$ are given as $\underline{f}_i = \prod_{i=1}^n \underline{\mu}_{\tilde{F}_n^i}, \qquad \overline{f}_i = \prod_{i=1}^n \overline{\mu}_{\tilde{F}_n^i}.$

Finally, a center-of-sets with the EIASC algorithm type-reduction was implemented in this design, $Y_{cos} = [y_l,\ y_r]$. EIASC algorithm has been found faster than the widely used KM type-reducer as no rule ordering is needed [22]. Lastly, the final output $y$ was calculated by taking the average of $y_l$ and $y_r$ as $y = (y_l + y_r)/2$.

Table 5.1: X-control fuzzy rules representation

| **Rules** | **Inputs** | | **Outputs** | |
|---|---|---|---|---|
| | $e_x$ | $\dot{e}_x$ | $x$ **position** | |
| $r_1$ | N | N | $\upsilon_1^l = \upsilon_{01}^l + \upsilon_{11}^l e_x + \upsilon_{21}^l \dot{e}_x$ | $\upsilon_1^r = \upsilon_{01}^r + \upsilon_{11}^r e_x + \upsilon_{21}^r \dot{e}_x$ |
| $r_2$ | N | Z | $\upsilon_2^l = \upsilon_{02}^l + \upsilon_{12}^l e_x + \upsilon_{22}^l \dot{e}_x$ | $\upsilon_2^r = \upsilon_{02}^r + \upsilon_{12}^r e_x + \upsilon_{22}^r \dot{e}_x$ |
| $r_3$ | N | P | $\upsilon_3^l = \upsilon_{03}^l + \upsilon_{13}^l e_x + \upsilon_{23}^l \dot{e}_x$ | $\upsilon_3^r = \upsilon_{03}^r + \upsilon_{13}^r e_x + \upsilon_{23}^r \dot{e}_x$ |
| $r_4$ | Z | N | $\upsilon_4^l = \upsilon_{04}^l + \upsilon_{14}^l e_x + \upsilon_{24}^l \dot{e}_x$ | $\upsilon_4^r = \upsilon_{04}^r + \upsilon_{14}^r e_x + \upsilon_{24}^r \dot{e}_x$ |
| $r_5$ | Z | Z | $\upsilon_5^l = \upsilon_{05}^l + \upsilon_{15}^l e_x + \upsilon_{25}^l \dot{e}_x$ | $\upsilon_5^r = \upsilon_{05}^r + \upsilon_{15}^r e_x + \upsilon_{25}^r \dot{e}_x$ |
| $r_6$ | Z | P | $\upsilon_6^l = \upsilon_{06}^l + \upsilon_{16}^l e_x + \upsilon_{26}^l \dot{e}_x$ | $\upsilon_6^r = \upsilon_{06}^r + \upsilon_{16}^r e_x + \upsilon_{26}^r \dot{e}_x$ |
| $r_7$ | P | N | $\upsilon_7^l = \upsilon_{07}^l + \upsilon_{17}^l e_x + \upsilon_{27}^l \dot{e}_x$ | $\upsilon_7^r = \upsilon_{07}^r + \upsilon_{17}^r e_x + \upsilon_{27}^r \dot{e}_x$ |
| $r_8$ | P | Z | $\upsilon_8^l = \upsilon_{08}^l + \upsilon_{18}^l e_x + \upsilon_{28}^l \dot{e}_x$ | $\upsilon_8^r = \upsilon_{08}^r + \upsilon_{18}^r e_x + \upsilon_{28}^r \dot{e}_x$ |
| $r_9$ | P | P | $\upsilon_9^l = \upsilon_{09}^l + \upsilon_{19}^l e_x + \upsilon_{29}^l \dot{e}_x$ | $\upsilon_9^r = \upsilon_{09}^r + \upsilon_{19}^r e_x + \upsilon_{29}^r \dot{e}_x$ |

Table 5.2: Y-control fuzzy rules representation

| **Rules** | **Inputs** | | **Outputs** | |
|---|---|---|---|---|
| | $e_y$ | $\dot{e}_y$ | $y$ **position** | |
| $r_1$ | N | N | $\upsilon_1^l = \upsilon_{01}^l + \upsilon_{11}^l e_y + \upsilon_{21}^l \dot{e}_y$ | $\upsilon_1^r = \upsilon_{01}^r + \upsilon_{11}^r e_y + \upsilon_{21}^r \dot{e}_y$ |
| $r_2$ | N | Z | $\upsilon_2^l = \upsilon_{02}^l + \upsilon_{12}^l e_y + \upsilon_{22}^l \dot{e}_y$ | $\upsilon_2^r = \upsilon_{02}^r + \upsilon_{12}^r e_y + \upsilon_{22}^r \dot{e}_y$ |
| $r_3$ | N | P | $\upsilon_3^l = \upsilon_{03}^l + \upsilon_{13}^l e_y + \upsilon_{23}^l \dot{e}_y$ | $\upsilon_3^r = \upsilon_{03}^r + \upsilon_{13}^r e_y + \upsilon_{23}^r \dot{e}_y$ |
| $r_4$ | Z | N | $\upsilon_4^l = \upsilon_{04}^l + \upsilon_{14}^l e_y + \upsilon_{24}^l \dot{e}_y$ | $\upsilon_4^r = \upsilon_{04}^r + \upsilon_{14}^r e_y + \upsilon_{24}^r \dot{e}_y$ |
| $r_5$ | Z | Z | $\upsilon_5^l = \upsilon_{05}^l + \upsilon_{15}^l e_y + \upsilon_{25}^l \dot{e}_y$ | $\upsilon_5^r = \upsilon_{05}^r + \upsilon_{15}^r e_y + \upsilon_{25}^r \dot{e}_y$ |
| $r_6$ | Z | P | $\upsilon_6^l = \upsilon_{06}^l + \upsilon_{16}^l e_y + \upsilon_{26}^l \dot{e}_y$ | $\upsilon_6^r = \upsilon_{06}^r + \upsilon_{16}^r e_y + \upsilon_{26}^r \dot{e}_y$ |
| $r_7$ | P | N | $\upsilon_7^l = \upsilon_{07}^l + \upsilon_{17}^l e_y + \upsilon_{27}^l \dot{e}_y$ | $\upsilon_7^r = \upsilon_{07}^r + \upsilon_{17}^r e_y + \upsilon_{27}^r \dot{e}_y$ |
| $r_8$ | P | Z | $\upsilon_8^l = \upsilon_{08}^l + \upsilon_{18}^l e_y + \upsilon_{28}^l \dot{e}_y$ | $\upsilon_8^r = \upsilon_{08}^r + \upsilon_{18}^r e_y + \upsilon_{28}^r \dot{e}_y$ |
| $r_9$ | P | P | $\upsilon_9^l = \upsilon_{09}^l + \upsilon_{19}^l e_y + \upsilon_{29}^l \dot{e}_y$ | $\upsilon_9^r = \upsilon_{09}^r + \upsilon_{19}^r e_y + \upsilon_{29}^r \dot{e}_y$ |

Table 5.3: Z-control fuzzy rules representation

| **Rules** | **Inputs** | | **Outputs** | |
|---|---|---|---|---|
| | $e_z$ | $\dot{e}_z$ | $z$ **position** | |
| $r_1$ | N | N | $\upsilon_1^l = \upsilon_{01}^l + \upsilon_{11}^l e_z + \upsilon_{21}^l \dot{e}_z$ | $\upsilon_1^r = \upsilon_{01}^r + \upsilon_{11}^r e_z + \upsilon_{21}^r \dot{e}_z$ |
| $r_2$ | N | Z | $\upsilon_2^l = \upsilon_{02}^l + \upsilon_{12}^l e_z + \upsilon_{22}^l \dot{e}_z$ | $\upsilon_2^r = \upsilon_{02}^r + \upsilon_{12}^r e_z + \upsilon_{22}^r \dot{e}_z$ |
| $r_3$ | N | P | $\upsilon_3^l = \upsilon_{03}^l + \upsilon_{13}^l e_z + \upsilon_{23}^l \dot{e}_z$ | $\upsilon_3^r = \upsilon_{03}^r + \upsilon_{13}^r e_z + \upsilon_{23}^r \dot{e}_z$ |
| $r_4$ | Z | N | $\upsilon_4^l = \upsilon_{04}^l + \upsilon_{14}^l e_z + \upsilon_{24}^l \dot{e}_z$ | $\upsilon_4^r = \upsilon_{04}^r + \upsilon_{14}^r e_z + \upsilon_{24}^r \dot{e}_z$ |
| $r_5$ | Z | Z | $\upsilon_5^l = \upsilon_{05}^l + \upsilon_{15}^l e_z + \upsilon_{25}^l \dot{e}_z$ | $\upsilon_5^r = \upsilon_{05}^r + \upsilon_{15}^r e_z + \upsilon_{25}^r \dot{e}_z$ |
| $r_6$ | Z | P | $\upsilon_6^l = \upsilon_{06}^l + \upsilon_{16}^l e_z + \upsilon_{26}^l \dot{e}_z$ | $\upsilon_6^r = \upsilon_{06}^r + \upsilon_{16}^r e_z + \upsilon_{26}^r \dot{e}_z$ |
| $r_7$ | P | N | $\upsilon_7^l = \upsilon_{07}^l + \upsilon_{17}^l e_z + \upsilon_{27}^l \dot{e}_z$ | $\upsilon_7^r = \upsilon_{07}^r + \upsilon_{17}^r e_z + \upsilon_{27}^r \dot{e}_z$ |
| $r_8$ | P | Z | $\upsilon_8^l = \upsilon_{08}^l + \upsilon_{18}^l e_z + \upsilon_{28}^l \dot{e}_z$ | $\upsilon_8^r = \upsilon_{08}^r + \upsilon_{18}^r e_z + \upsilon_{28}^r \dot{e}_z$ |
| $r_9$ | P | P | $\upsilon_9^l = \upsilon_{09}^l + \upsilon_{19}^l e_z + \upsilon_{29}^l \dot{e}_z$ | $\upsilon_9^r = \upsilon_{09}^r + \upsilon_{19}^r e_z + \upsilon_{29}^r \dot{e}_z$ |

### 5.4.2 PROBLEM FORMULATION

Consider a nonlinear dynamic system $n^{th}$ order as

$$x^{(n)} = \overline{f}(\mathbf{x}, t) + \overline{b}(\mathbf{x}, t)u(t) + d(t), \tag{5.8}$$

where the state vector $\mathbf{x}$ can be represented as $\mathbf{x} = \begin{bmatrix} x & \dot{x} & ... & x^{(n-1)} \end{bmatrix}^T$; $\overline{f}(\mathbf{x}, t)$ and $\overline{b}(\mathbf{x}, t)$ denote the state vector of nonlinear functions; $u(t)$ represents the control input; and $d(t)$ expresses an external disturbance.

Considering the tracking error that is, the difference between the desired and the actual values as:

$$\mathbf{e} = \mathbf{x}_d - \mathbf{x} = \begin{bmatrix} e(t) & \dot{e}(t) & ... & e(t)^{(n-1)} \end{bmatrix}^T, \tag{5.9}$$

where $\mathbf{x}_d = \begin{bmatrix} x_d & \dot{x}_d & ... & x_d^{(n-1)} \end{bmatrix}^T$ is the desired tracking values, and $\mathbf{x}$ is the actual value. Eq. (5.9) can be expanded as follows:

$$\begin{cases} \dot{e}_1 & = & e_2(t) \\ \dot{e}_2 & = & e_3(t) \\ & \vdots & \\ \dot{e}_n(t) & = & \dot{\mathbf{x}}_{dn} - \overline{f}(\mathbf{x}, t) - \overline{b}(\boldsymbol{x}, t)u(t) - d(t) \end{cases} \tag{5.10}$$

**Remark 2** *Practically, it is difficult to measure all state variables of the system. Hence, this study selects two major control variables, named, the error $e_n(t)$ and its derivative $\dot{e}_n(t)$.*

### 5.4.3 Design of Sliding Surface

**Remark 3** *The design of SMC consists of two stages: 1) design of the reaching phase, and 2) design of a sliding surface phase. Such control techniques employ a discontinuous control law that has the ability to drive the system to a specified sliding surface $S(t)$, and also to preserve its motion along $S(t)$ [121, 286].*

A sliding surface $S(e, t) = 0$ can be defined as follows:

$$S(t) = \delta e(t) + \dot{e}(t), \tag{5.11}$$

where $\delta$ is a strictly positive constant. For the theoretical study, it is appropriate to assume the nonlinear terms in Eq. (5.8) are known. However, this may not be true in real life. Hence, the control law $u_{final}(t)$ can be constructed as follows [246]:

$$u_{final}(t) = \overline{b}(\mathbf{x}, t)^{-1}[\dot{\mathbf{x}}_{dn}(t) - \overline{f}(\mathbf{x}, t) - \dot{e}_n(t) + \dot{S}(t) + \delta S(t)]. \tag{5.12}$$

The inputs to the proposed fuzzy controller are $e(t)$ and $\dot{e}(t)$. Moreover, for producing the control signal $u(t)$ in Eq. (5.8), fuzzy operations are deployed to approximate the equivalent control law $u_{final}(t)$. It is worth mentioning that the purpose of using SMC is to derive the dynamics of the system so that $S(t) = \delta e(t) + \dot{e}(t) = 0$. By following the equivalent control law $u_{final}(t)$ and by considering the sliding surface and its derivative as the inputs to the fuzzy control system, the system is asymptotically stable [257], so that $\delta S(t) + \dot{S}(t) = 0$.

The convergence of $S(t)$ and $\dot{S}(t)$ to zero is guaranteed as $\delta$ is always a positive number. Likewise, the convergence of $e(t)$ and $\dot{e}(t)$ to zero is always guaranteed according to the definition in Eq. (5.11). To avoid the complexity of model-based computations, this study utilized fuzzy system capability to map between the input variables and the control law $u(t)$. In this case, the control input might have differences from the optimal control law $u_{final}(t)$. Hence, using Eqs. (5.10) and (5.12), the following equation can be derived as

$$\dot{S}(t) = \overline{b}(\mathbf{x}, t)[u_{final}(t) - u(t)] - \delta S(t). \tag{5.13}$$

Multiplying Eq. (5.13) with $S(t)$, it yields:

$$\dot{S}(t)S(t) = S(t)\left(\overline{b}(\mathbf{x}, t)[u_{final}(t) - u(t)] - \delta S(t)\right) \tag{5.14}$$

Following the Lyapunov theory leads to $\dot{S}(t)S(t) < 0$, which represents the reaching phase of the sliding surface. Therefore, the purpose of this study is to design a control signal $u(t)$ that satisfies the reaching condition to guarantee the convergence of the overall control system.

**Remark 4** *To obtain a robust control performance against system dynamics uncertainties, a discontinuous term is added to the final control part across the sliding surface $S(t)$. In other words, the discontinuous term acts as a robustness term, and can be added as the reaching control element of the control effort [287].*

The discontinuous term can be represented as:

$$u_{robust} = \beta(t) sgn(S(t)), \tag{5.15}$$

where $\beta(t) > 0$ and *sgn* represents the *signum* function, which can be defined as follows:

$$sgn(S(t)) = \begin{cases} 1, & if \ S(t) > 0 \\ -1, & if \ S(t) < 0 \end{cases} \tag{5.16}$$

where $sgn(S(t)) = 0 \ if \ S(t) = 0$. Nevertheless, employing the *signum* function causes a chattering phenomenon. One way to eliminate the chattering phenomenon is by smoothing out the continuity of the *signum* function and employing a smooth function such as *sat* or *tanh*. In this design, the *sat* function is deployed, which can be expressed as follows [288]:

$$sat\left(S, \iota\right) = \begin{cases} \frac{S}{\iota}, & if \ \ |S| \leq |\iota| \\ sgn(S), & otherwise \end{cases} \tag{5.17}$$

where $\iota$ is a design factor representing the thickness of the boundary layer. Finally, the constructed control law considering uncertainties can be written as:

$$u(t) = u_{final}(t) + u_{robust}(t), \tag{5.18}$$

where $u_{robust}(t) = \beta(t) sat(S(t)/\iota)$.

### 5.4.4 ESAF2C Adaptive Law

The proposed adaptation law for the ESAF2C is obtained using the gradient descent method, that is to minimize the $S(t)\dot{S}(t)$ with respect to $\upsilon^l$ and $\upsilon^r$ in Eq. (5.4.1). The proposed structure can be depicted in Fig. 5.4. Hence, the modified $\upsilon^l$ and $\upsilon^r$ can be expressed as follows:

$$\begin{cases} \upsilon^l_{t+1} = \upsilon^l(t) - \Lambda \frac{\partial S(t)\dot{S}(t)}{\partial \upsilon^l(t)} \\ \upsilon^r_{t+1} = \upsilon^r(t) - \Lambda \frac{\partial S(t)\dot{S}(t)}{\partial \upsilon^r(t)}, \end{cases} \tag{5.19}$$

where $\Lambda$ is a design adaptive parameter and $[\upsilon^l,\ \upsilon^r]$ are the consequent fuzzy parameters. By applying the chain rule, Eq. (5.19) can be rewritten as:

$$\begin{cases} \upsilon^l_{t+1} = \upsilon^l(t) - \Lambda \frac{\partial S(t)\dot{S}(t)}{\partial u(t)} \frac{\partial u(t)}{\partial \upsilon^l(t)} = \upsilon^l(t) + \Lambda \bar{b}(\mathbf{x},t) S(t) \frac{\partial u(t)}{\partial \upsilon^l(t)}. \\ \upsilon^r_{t+1} = \upsilon^r(t) - \Lambda \frac{\partial S(t)\dot{S}(t)}{\partial u(t)} \frac{\partial u(t)}{\partial \upsilon^r(t)} = \upsilon^r(t) + \Lambda \bar{b}(\mathbf{x},t) S(t) \frac{\partial u(t)}{\partial \upsilon^r_( t)}. \end{cases} \tag{5.20}$$

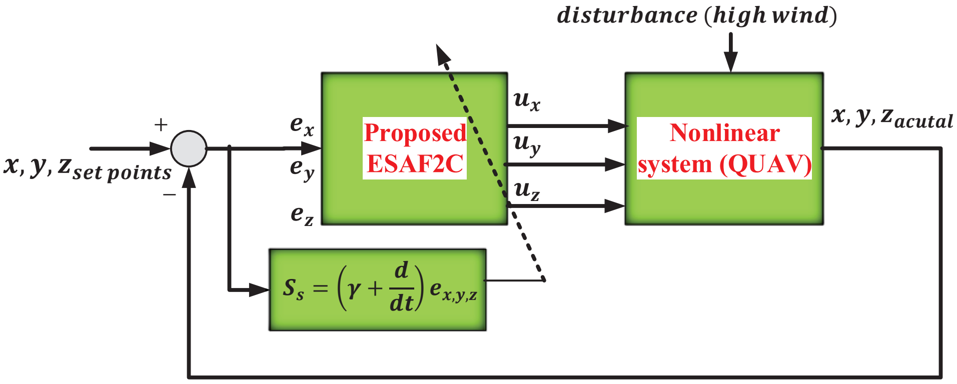


Figure 5.4: Proposed ESAF2C structure for QUAV control, where $u_x, u_y, u_z$ represent the control signals for the $xyz$-axes, respectively.

To simplify (5.20) further, we combine the design adaptive parameter $\Lambda$ with the overall system input parameter $\bar{b}(\mathbf{x}, t)$ as a learning parameter, $\lambda$, [246, 289]. Therefore, for the sake of practical implementation, the adaptive law with respect to the firing strength can be rewritten as follows:

$$\begin{cases} \upsilon^l_{t+1} = \upsilon^l(t) + \lambda S(t) \left( \frac{\underline{f}^i(t))}{\sum_{i=1}^m \left( \overline{f}^i(t) + \underline{f}^i(t) \right)} \right) \\ \upsilon^r_{t+1} = \upsilon^r(t) + \lambda S(t) \left( \frac{\overline{f}^i(t))}{\sum_{i=1}^m \left( \overline{f}^i(t) + \underline{f}^i(t) \right)} \right). \end{cases} \tag{5.21}$$

In the work by [258, 259], a dead-zone concept was introduced to reduce the drift effect of the fuzzy consequents. Hence, the modified equations for updating the consequent parameters can be rewritten as follows:

$$\begin{cases} \upsilon^l_{t+1} = 0, & if\ S \leq \iota \\ \upsilon^r_{t+1} = 0, & if\ S \leq \iota \\ \upsilon^l_{t+1} = \kappa|S|\upsilon^l(t) + \lambda S(t) \left( \frac{\underline{f}^i(t))}{\sum_{i=1}^m \left( \overline{f}^i(t) + \underline{f}^i(t) \right)} \right), & if\ S > \iota \\ \upsilon^r_{t+1} = \kappa|S|\upsilon^r(t) + \lambda S(t) \left( \frac{\overline{f}^i(t))}{\sum_{i=1}^m \left( \overline{f}^i(t) + \underline{f}^i(t) \right)} \right), & if\ S > \iota \end{cases} \tag{5.22}$$

where $\kappa$ is a design parameter $> 0$; and $\iota$ is the thickness of the boundary layer as discussed in (5.17).

### 5.4.5 Stability Analysis

A fuzzy system can be utilised to represent and approximate any nonlinear function to obtain reasonable accuracy. The following basic assumptions were made in order to assess the stability of the proposed ESAF2C as described in Remark 5.4.1 using the Lyapunov theory.

**Lemma 5.4.1** *[246] If optimal upper and lower consequent parameters $\upsilon^l$ and $\upsilon^r$ exist, which lead to the control law $\tilde{u}$, the final approximate of the control law $\boldsymbol{u}_{final}$ has bounded error of $\zeta$, for each of the control inputs of the system, that leads to:*

$$max|\tilde{u}(\mathbf{x}, \bar{\upsilon}) - \mathbf{u}_{final}(x)| < \zeta, \tag{5.23}$$

where $\tilde{u}(\mathbf{x}, \overline{\upsilon}) = \overline{\upsilon}^T\mathbf{W}$; $\mathbf{u}_{final}(t)(x) = \overline{\upsilon}^T\mathbf{W} + \zeta$;
$\boldsymbol{W} = [W_l, W_r] = \left[ \frac{\underline{f}^i(t))}{\sum_{i=1}^m \left( \overline{f}^i(t) + \underline{f}^i(t) \right)}, \frac{\overline{f}^i(t))}{\sum_{i=1}^m \left( \overline{f}^i(t) + \underline{f}^i(t) \right)} \right]$ and $\bar{\upsilon} = \left[ \bar{\upsilon}^l, \bar{\upsilon}^r \right]$.

If we describe $\bar{\upsilon} = \overline{\upsilon} - \tilde{\upsilon}$, which can be defined as the difference between the desired and actual consequent, we can rewrite (5.13) as:

$$\dot{S}(t) = \overline{b}[\bar{\upsilon}^T\mathbf{W} + \zeta] - \delta S(t). \tag{5.24}$$

This study selects the Lyapunov function as follows:

$$V = \frac{1}{2}s_{\nabla}^2 + \frac{\bar{b}}{2\lambda}\bar{\upsilon}^T\bar{\upsilon}, \quad \lambda \neq 0, \tag{5.25}$$

where $s_{\nabla} \equiv S - \iota \bullet sat(S/\iota)$ and $\iota$ denotes the thickness of the boundary layer [257].

By differentiating (5.25) with respect to time, it leads to:

$$\begin{cases} & = & s_{\nabla}\dot{s}_{\nabla} + \frac{\bar{b}}{\lambda}\bar{\upsilon}^T\dot{\bar{\upsilon}} + \frac{\dot{\bar{b}}}{2\lambda}\bar{\upsilon}^T\bar{\upsilon} \\ & = & s_{\nabla}\dot{s}_{\nabla} + \frac{\bar{b}}{\lambda}\bar{\upsilon}^T\left[\dot{\bar{\upsilon}} - \dot{\tilde{\upsilon}}\right] + \frac{\dot{\bar{b}}}{2\lambda}\bar{\upsilon}^T\bar{\upsilon} \\ & = & s_{\nabla}\left[-\delta s + \bar{b}\left(\bar{\upsilon}^T\mathbf{W} + \zeta\right)\right] \\ & & -\bar{b}\bar{\upsilon}^T\left(s_{\nabla}\mathbf{W} - \frac{\kappa}{\lambda}|s_{\nabla}|\tilde{\upsilon}\right) + \frac{\dot{\bar{b}}}{2\lambda}\bar{\upsilon}^T\bar{\upsilon} \\ \dot{V} & = & s_{\nabla}\left[-\delta(s_{\nabla} + \iota) + \bar{b}\zeta\right] + \frac{\bar{b}}{\lambda}\kappa|s_{\nabla}|\bar{\upsilon}^T\tilde{\upsilon} + \frac{\dot{\bar{b}}}{2\lambda}\bar{\upsilon}^T\bar{\upsilon} \\ & \leq & |s_{\nabla}|\left(-\delta|s_{\nabla}| - \delta\iota + \bar{b}\zeta - \frac{1}{\lambda}\bar{b}\kappa\bar{\upsilon}^T\bar{\upsilon}\right. \\ & & \left.+\frac{1}{\lambda}\bar{b}\kappa|\bar{\upsilon}||\overline{\upsilon}|\right) + \frac{\dot{\bar{b}}}{2\lambda}\bar{\upsilon}^T\bar{\upsilon} \\ & = & -|s_{\nabla}|\Theta - \frac{1}{\lambda}\left[|s_{\nabla}|\bar{b}\kappa - \frac{\dot{\bar{b}}}{2}\right]\bar{\upsilon}^T\bar{\upsilon}, \end{cases} \tag{5.26}$$

where $\Theta = \delta\iota + \delta|s_{\nabla}| - \bar{b}\left(\zeta + \frac{1}{\lambda}\kappa|\bar{\upsilon}||\overline{\upsilon}|\right)$.

Following the work in [246], choosing suitable parameters for $\iota$ and $\kappa$ for $\Theta > 0$, (5.26) indicates that $\dot{V} < 0$ at any time that $s_{\nabla} \notin R \equiv \left(|s_{\nabla}| < (\frac{\dot{\bar{b}}}{2\kappa\bar{b}\lambda})\right)$. Therefore, the stability of the control system is guaranteed based on the Lyapunov theory.

## 5.5 Simulation Results

In this section, the efficacy of the proposed adaptive control system in stabilizing the QUAV is investigated. The performance of the proposed ESAF2C is examined in the presence of external disturbances and measurement noise while tracking various reference signals along the $(xyz)$-axes. A comparison of the proposed method with a conventional PID controller and both T1-FLC and IT2FLC counterparts is reported. The performance indices such as the root mean square error (RMSE), rising time ($t_r$), and settling time ($t_s$) are reported in this section.

Nine rules were employed, with three membership functions for each input. The inputs to the proposed fuzzy controllers are the error and its derivative as depicted in Fig. 5.3. The sliding surface parameters were chosen as: $\delta_x = 0.8, \delta_y = 0.7, \delta_z = 1.1$. The consequent parameters started learning from scratch and were initialized with zeros.

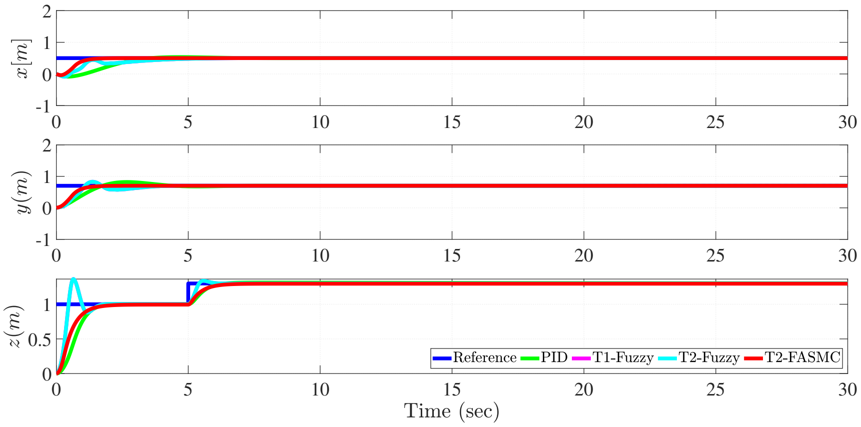


Figure 5.5: Simulation results for position control in the $xyz$-axes for different controllers on the nominal system (step input).

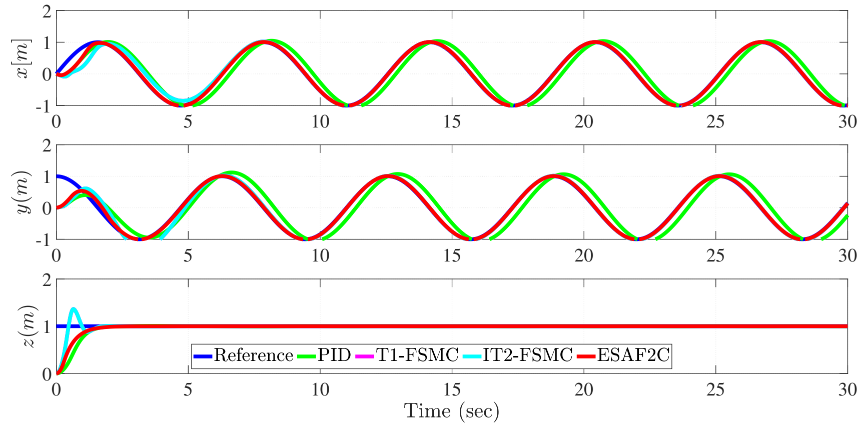


Figure 5.6: Simulation results for position control in the $xyz$-axes for different controllers on the nominal system (sine input).

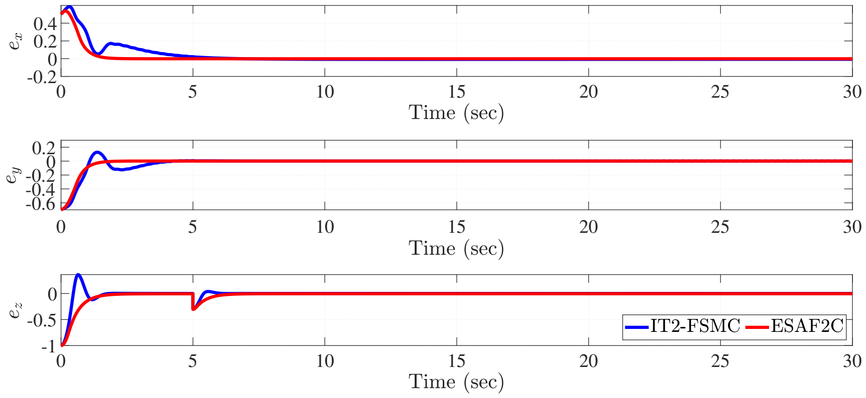


Figure 5.7: Simulation results for the error signals for IT2-FSMC and ESAF2C on the nominal system.

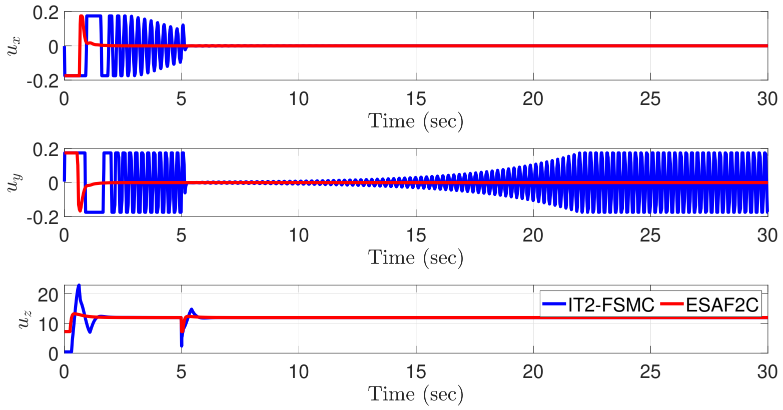


Figure 5.8: Simulation results for the control signals for IT2-FSMC and ESAF2C on the nominal system.

### 5.5.1 Case 1: nominal condition

First, a step reference was fed as the desired trajectory at a hovering point of $[0.7, 0.8, 1]$ $m$ for the $xyz$-axes respectively. For the altitude position, the set-point was increased to 1.2 $m$ after 5 *sec*. A better control system performance was obtained using the proposed ESAF2C compared to its T1-fuzzy and T2-fuzzy counterparts, which demonstrates comparable performance to a conventional PID controller in the case of nominal condition as shown in Fig. 5.5. Besides, the analysis was performed with different reference signals (e.g., sine wave signal), where better closed-loop control performance was observed using the proposed method compared to other controllers as shown in Fig. 5.6. The tracking errors and the control efforts throughout the experiment are shown in Fig. 5.7 and Fig. 5.8, demonstrating a favorable performance using the ESAF2C.

### 5.5.2 Case 2: disturbance condition

For further investigation of the proposed technique, an artificial stochastic wind/gust disturbance was injected to the system. For simulating the wind/gust disturbance, it is generated using a discrete wind/gust MATLAB/SIMULINK block, which have the following setting: gust length $[dx = 50m,\ dy = 50m,\ dz = 30m]$, wind velocity$=2m/s$ and gust amplitude of $[ug = 5m/s,\ vg = 5m/s,\ wg = 5m/s]$. The discrete wind/gust block can be found in the Aerospace MATLAB blockset library.

As can be seen in Figs. 5.9 & 5.10, the conventional PID controller failed to stabilize the system in the presence of disturbances, while a stable performance was observed using the ESAF2C technique. For more visualization, the dotted squares and dotted circles in Figs. 5.9, 5.10, 5.11 are presented, which demonstrate the improvement of the proposed technique compared to the other controllers. The proposed ESAF2C performed better than their type-1 and type-2 fuzzy counterparts as indicated by its lower error values shown in Fig. 5.11 and Table. 5.4. Moreover, as can be seen in Fig. 5.12, the chattering effect was eliminated using the ESAF2C by adopting the *saturation* function to smooth out the chattering control discontinuity of the sliding surface.

Table 5.4: Simulation results for multiple controllers with disturbances

| | | | | | |
|---|---|---|---|---|---|
| | | | $\boldsymbol{x - position}$ | | |
| **Metrics** | **PID** | **T1-FSMC** | **IT2-FSMC** | **ESAF2C** | **Units** |
| $\boldsymbol{t_r}$ | 0.78 | 3.63 | 3.52 | 1.79 | $(s)$ |
| $\boldsymbol{t_s}$ | 25.92 | 17.85 | 15.85 | 2.35 | $(s)$ |
| $\boldsymbol{RMSE}$ | 0.356 | 0.125 | 0.109 | 0.0721 | $(m)$ |
| | | | $\boldsymbol{y - position}$ | | |
| **Metrics** | **PID** | **T1-FSMC** | **IT2-FSMC** | **ESAF2C** | **Units** |
| $\boldsymbol{t_r}$ | 1.26 | 21.52 | 20.56 | 5.86 | $(s)$ |
| $\boldsymbol{t_s}$ | 24.36 | 3.89 | 3.85 | 2.593 | $(s)$ |
| $\boldsymbol{RMSE}$ | 0.221 | 0.115 | 0.095 | 0.0833 | $(m)$ |
| | | | $\boldsymbol{z - position}$ | | |
| **Metrics** | **PID** | **T1-FSMC** | **IT2-FSMC** | **ESAF2C** | **Units** |
| $\boldsymbol{t_r}$ | 1.52 | 4.12 | 4.08 | 3.25 | $(s)$ |
| $\boldsymbol{t_s}$ | 29.81 | 16.42 | 14.25 | 5.84 | $(s)$ |
| $\boldsymbol{RMSE}$ | 0.542 | 0.215 | 0.1152 | 0.1013 | $(m)$ |

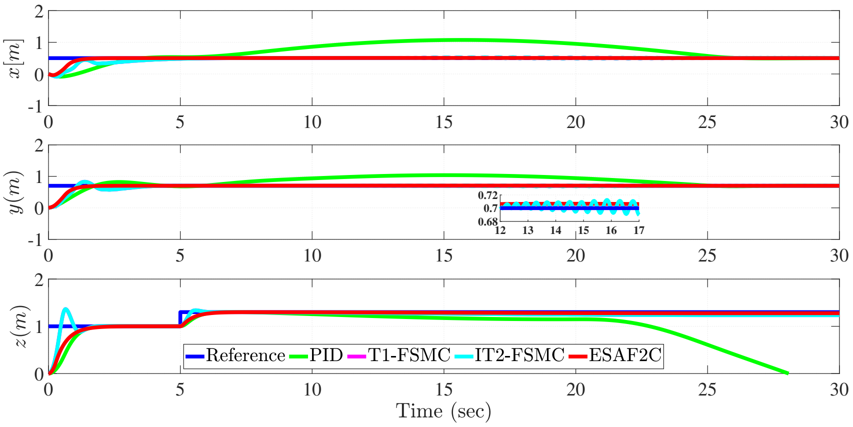


Figure 5.9: Simulation results for different controllers for position control in the $xyz$-axes under disturbances (step input).

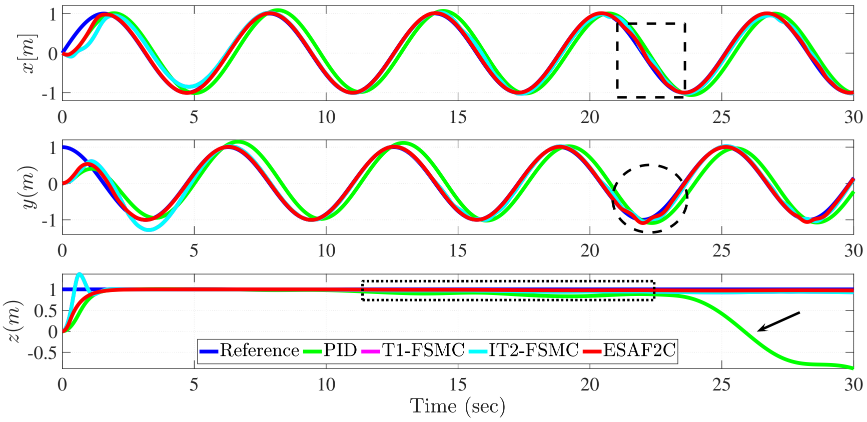


Figure 5.10: Simulation results for different controllers for position control in the $xyz$-axes under disturbances (Sine input).

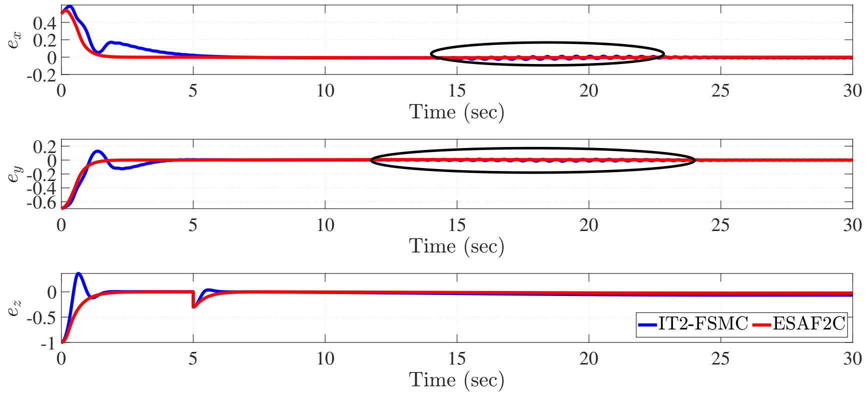


Figure 5.11: Simulation results for the error signals for IT2-FSMC and ESAF2C under uncertainties.

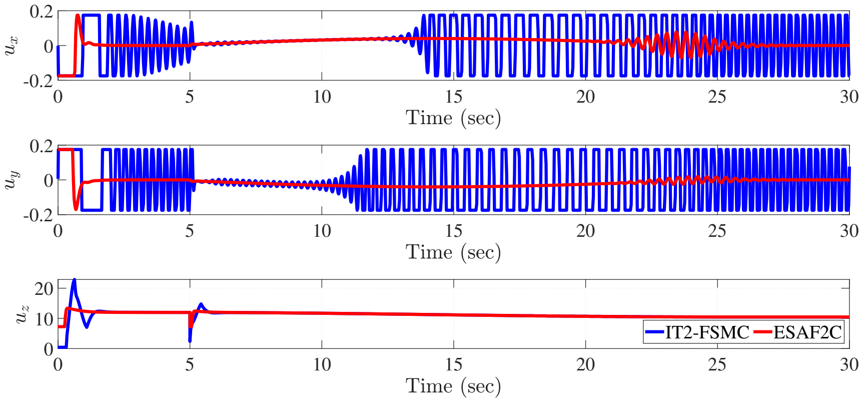


Figure 5.12: Simulation results for the control signals for IT2-FSMC and ESAF2C under uncertainties.

### 5.5.3 Case 3: sensor measurement noise effect

For further investigation of robustness, band-limited white noises with various noise powers were added to the sensor data, while having the stochastic artificial winds present. In other words, the noise was added to the position signals $(x, y, z)$. As shown in 5.13, ESAF2C proved its robustness and ability to handle noisy sensor data efficiently compared to T1-FSMC and IT2-FSMC counterparts.

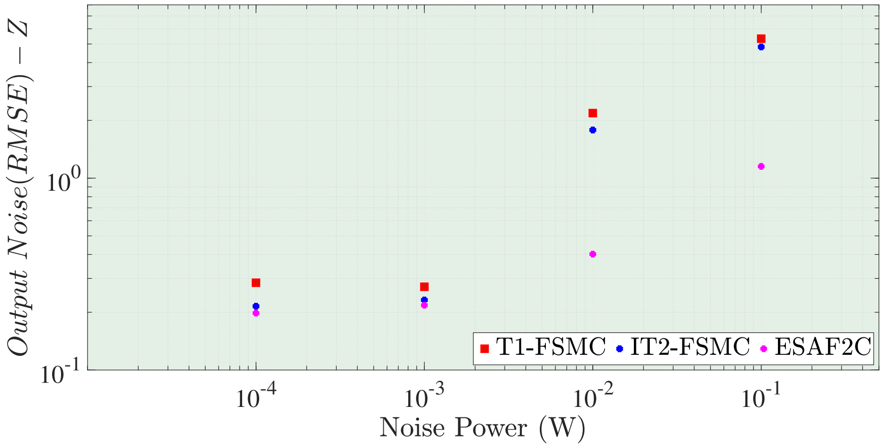


Figure 5.13: Measurement noise effect (comparison between ESAF2C, IT2-FSMC and T1-FSMC), $Z$-position control.

### 5.5.4 Case 4: Computational Load

The quantification of the computation load was evaluated under Core i7-10750H CPU @ 2.6GHz with 16.0 GB of RAM, which was collected using the *tic* and *toc* MATLAB functions. The sampling time = $0.01sec$ and the simulation time is $30sec$. Table 5.5 illustrates that the computation cost for the proposed method is higher than the PID and the T1-FSMC controllers. Despite being more computationally expensive, the performance of our adaptive fuzzy ESAF2C is superior compared to benchmark controllers, especially in the face of uncertainty. Today, with current speed of computer, this burden should not be a problem as it was in the past.

Table 5.5: Computation load for various controllers.

| **Control method** | **Computational load** ($sec$) |
|---|---|
| **PID** | 10.16 |
| **T1-FSMC** | 33.34 |
| **IT2-FSMC** | 56.46 |
| **ESAF2C** | 53.68 |

## 5.6 Experimental Results

To investigate the practical capability of the proposed method, multiple real flight tests were conducted using the Parrot AR.Drone (see Fig. 5.15). The experimental setup was discussed in chapter 3, except that the platform utilized in this chapter is the Parrot AR.Drone. This platform was chosen due to its lightweight and susceptibility to wind/-gusts.

The experiments were performed in the indoor flight test space, which is equipped with 19 VICON motion capture cameras. These powerful cameras have the ability to track all motions of the QUAV. We utilize the VICON Motion Tracker software to analyze and store real-time flight data (e.g., position, velocity, Euler angles, acceleration, and angular rates). The AR.Drone autonomy package, installed in a ground station computer, was subsequently utilized to control the QUAV via Wi-Fi at a frequency of 100 $Hz$, thanks to the Robot Operating System (ROS) protocol. The autonomy package enabled the QUAV to be remotely controlled through yaw rate, pitch rate, roll rate and thrust commands. Fig. 5.14 shows the information flow loops used for the flight tests.

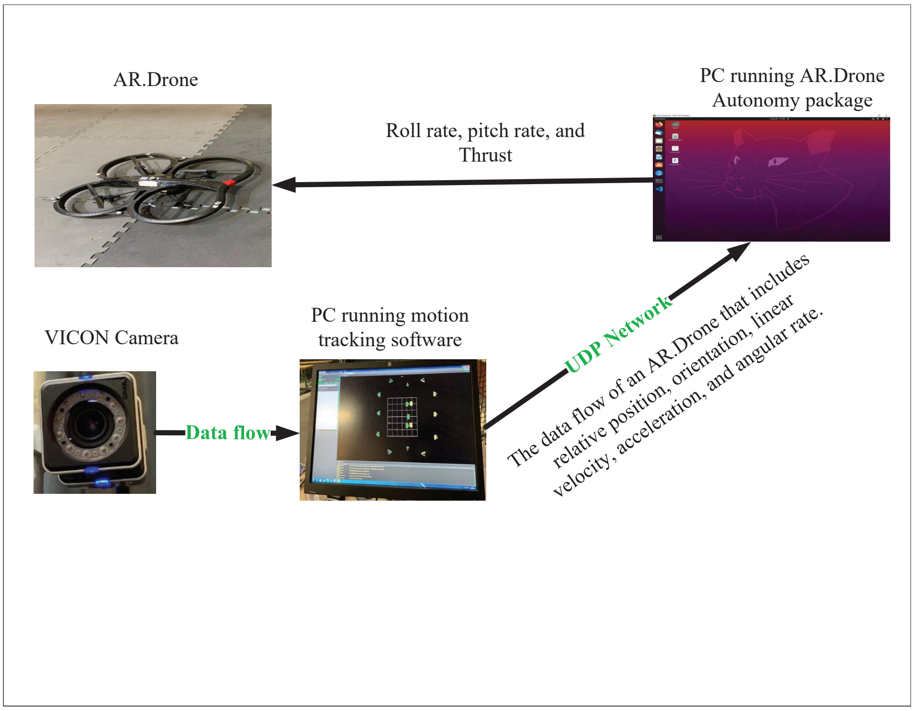

AR.Drone
PC running AR.Drone Autonomy package
Roll rate, pitch rate, and Thrust
VICON Camera
PC running motion tracking software
Data flow
UDP Network
The data flow of an AR.Drone that includes relative position, orientation, linear velocity, acceleration, and angular rate.

Figure 5.14: Information flow loop used for the flight tests.

Besides, we utilized nine fuzzy rules and a trapezoidal MF for real-time implementation, where the proposed model was first designed using computer simulation. Moving from simulation to real-time implementation, nine fuzzy rules were selected to achieve a delicate balance between computational complexity and accuracy. The initial parameters were also tuned using computer simulation, which then transferred to real-time implementation. The proposed controller was used to control the outer loop (position control loop) to track a given trajectory reference, while the inner loop is controlled using the PID control method as shown in Fig. 5.2.

The experimental setup and the data flow of the overall system architecture are demonstrated in Fig. 5.16. The performance indices were reported in Table 5.6. Besides, we demonstrated the adaptation trajectory of the upper and the lower ESAF2C parameters during the flight tests.

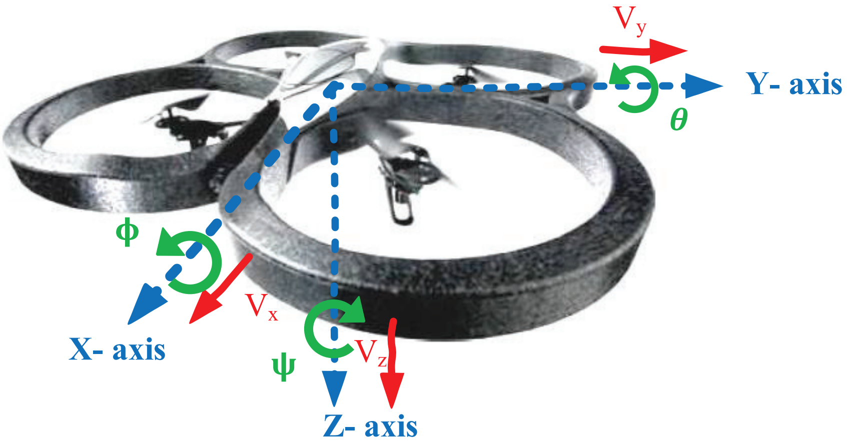


Figure 5.15: The body-frame $\{B\}$ of a Parrot AR.Drone, where $(\theta, \phi, \psi)$ represents the rotation along the $xyz$-axes respectively.

### 5.6.1 Case 1: nominal closed-loop position control

Various experiments were conducted to control the $(xyz)$ positions of the QUAV. As depicted in Fig. 5.17, the performance of three different controllers was compared. In this flight test, the desired tracking set points were set to be $[0, 0, 1 - 1.2]m$ for the $xyz$-axes, respectively. Fig. 5.17 shows a comparable performance of the PID and type-2 fuzzy counterparts with respect to the proposed ESAF2C technique. Besides, a shorter settling time was reported from the conventional PID controller, as the proposed ESAF2C requires time to adapt its parameters.

### 5.6.2 Case 2: external disturbance closed-loop position control

To validate the theoretical study, the performance under external disturbance is investigated. To do that, an industrial fan was utilized to act as a wind disturbance. The experimental results indicated that the performance of the proposed ESAF2C is better than its type-1 fuzzy and conventional PID counterparts as shown in Figs. 5.18 & 5.19, thanks to the *FOU* in IT2FLCs, that incorporate uncertainties efficiently. The online learning of the upper and the lower ESAF2C parameters are plotted in Fig. 5.20. The fuzzy parameters were varying according to the amount of applied gusts. In other words, Fig. 5.20, shows the convergence of the six parameters. Moreover, it is worth mentioning that the upper and the lower parameters started learning from scratch, where they were initialized with zeros. Table 5.6 shows the experimental evaluation for three different controllers under external wind disturbance, where lower RMSE values were recorded using the ESAF2C system.

Table 5.6: Experimental evaluation using three different controllers in hovering mode with high-wind disturbance

| $\boldsymbol{X-position}$ | | | | |
|---|---|---|---|---|
| **Metrics** | **PID** | **T1-Fuzzy** | **ESAF2C** | **Units** |
| $\boldsymbol{t_r}$ | 13.86 | 5.55 | 2.43 | $(s)$ |
| $\boldsymbol{t_s}$ | 59.39 | 58.94 | 55.90 | $(s)$ |
| $\boldsymbol{RMSE}$ | 0.1936 | 0.26 | 0.14 | $(m)$ |
| $\boldsymbol{Y-position}$ | | | | |
| **Metrics** | **PID** | **T1-Fuzzy** | **ESAF2C** | **Units** |
| $\boldsymbol{t_r}$ | 36.9 | 5.96 | 2.34 | $(s)$ |
| $\boldsymbol{t_s}$ | 58.98 | 42.54 | 35.32 | $(s)$ |
| $\boldsymbol{RMSE}$ | 0.30 | 0.23 | 0.19 | $(m)$ |
| $\boldsymbol{Z-position}$ | | | | |
| **Metrics** | **PID** | **T1-Fuzzy** | **ESAF2C** | **Units** |
| $\boldsymbol{t_r}$ | 4.0711 | 0.2479 | 0.1216 | $(s)$ |
| $\boldsymbol{t_s}$ | 60.4351 | 58.9833 | 52.3981 | $(s)$ |
| $\boldsymbol{RMSE}$ | 0.201 | 0.163 | 0.095 | $(m)$ |

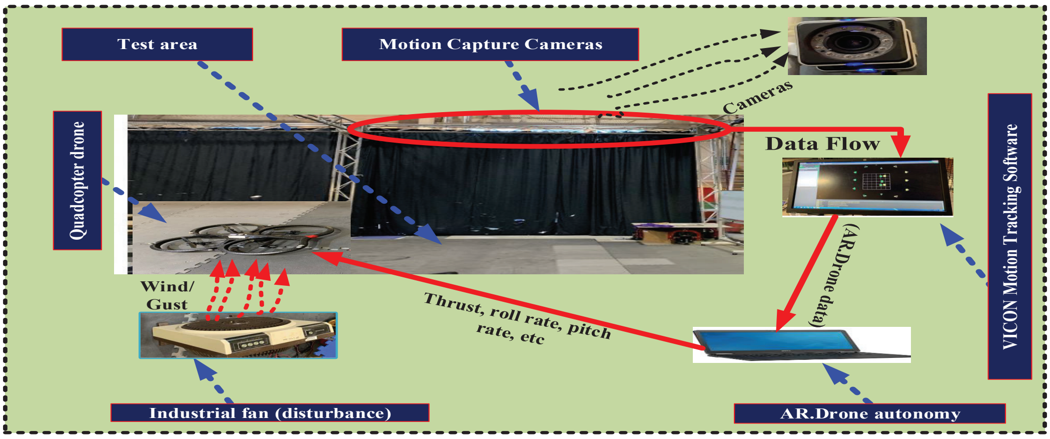


Figure 5.16: Data flow of the overall system architecture, demonstrating the information flow of the QUAV including the position, orientation, velocity, acceleration, angular rates, etc.

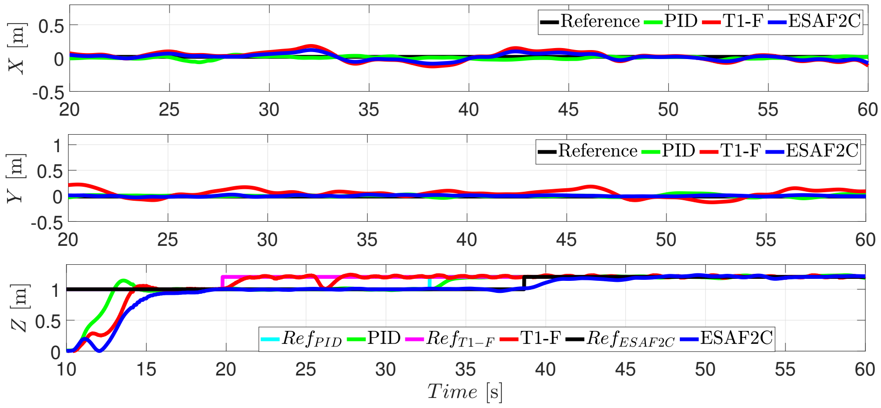


Figure 5.17: Output $xyz$ positions of the QUAV during real-time flight tests (hover mode) for different controllers.

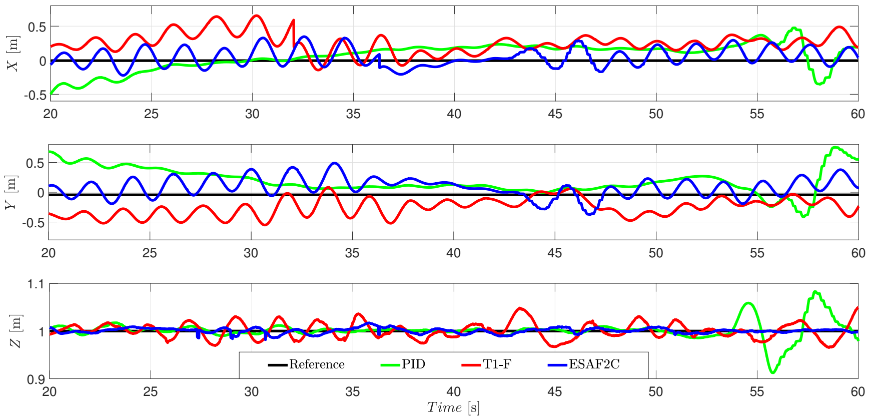


Figure 5.18: Output $xyz$ positions under external disturbances during real-time flight tests (hover mode) for different controllers.

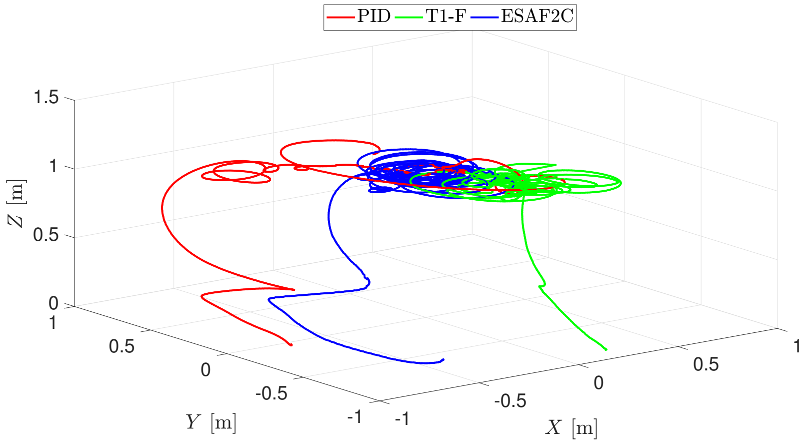


Figure 5.19: Output $xyz$ positions under external disturbance during real-time flight tests (hover mode) in 3-D shape.

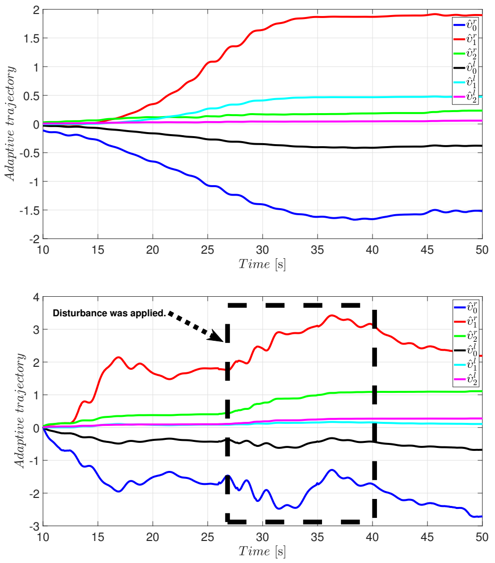


Figure 5.20: Upper figure shows the online parameters learning using the ESAF2C in hover mode (nominal condition), lower figure shows online parameters learning using the ESAF2C (under disturbance).

## 5.7 Summary

In this chapter, a novel ESAF2C controller was proposed to stabilize a quadcopter drone under external disturbances. On the basis of the SMC scheme, a self-tuning technique was implemented to tune SMC parameters, where SMC was utilized to tune fuzzy upper and lower variables. This approach was validated using extensive numerical simulations and real-time flight tests. The proposed method was performed and implemented in the ROS environment. The stability analysis was examined using the Lyapunov theory.

The proposed fuzzy control system approach can be considered as a promising technique. The proposed control scheme was computationally efficient as it requires only three membership functions, resulting in nine fuzzy rules only. Moreover, the proposed technique system demonstrated better control performance in the face of uncertainties (ground effects) and external disturbance (wind gusts) when compared to its type-1 fuzzy and conventional PID counterparts. Furthermore, the proposed closed-loop control system proved its ability to filter measurement noise that might occur during real-time implementation.

In addition, the outcomes of this chapter provided extra flexibility to fine-tune IT2FLCs and demonstrated the capability to implement IT2FLCs in real-time.

# Chapter 6

# Evolving Interval Type-2 Fuzzy Control for Uncertain Dynamic Systems

The work furnished in this chapter has been taken from the following publications:

- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020). A Novel Learning-from-Scratch Evolving Type-2 Fuzzy System for Robotic Control. *Preparing to be submitted to IEEE Transactions on Intelligent Transportation Systems*, [103].
- **Al-Mahturi, A.**, Santoso, F., Garratt, M. A., and Anavatti, S. G. (2020).A Simplified Model-Free Self-Evolving TS Fuzzy Controller for Nonlinear Systems with Uncertainties. *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, [290].

## 6.1 Introduction

The development of nonlinear control systems for mobile robots has been an important research area in recent years. For example, modern control methods have been developed for trajectory tracking [223, 224], navigation [225], parking [226], and obstacle avoidance [114]. Traditional control methods have been proposed for tracking control of mobile robots [223] and feedback linearization [291]. Nevertheless, these algorithms can provide a satisfactory tracking performance only if the plant's mathematical dynamic model is properly known, which may be impractical due to various uncertainties, such as, modelling errors, noisy measurements, etc [130].

On the other hand, computational intelligence control approaches such as FLC and ANN have been applied successfully to deal with uncertainties, such as in trajectory tracking control problems, where multiple uncertainties exist in the environment. Intelligent controllers can learn the dynamics of robots while accommodating the footprint-of-uncertainties (structured and unstructured) in an online manner. This way, the controller parameters and structure can be automatically adjusted based on the operating conditions of the robot without the need for a comprehensive mathematical model of the plant [130].

For instance, T1-FLCs have been implemented for mobile robots [292]. However, T1-FLCs have limited ability to handle uncertainties in the system, especially to handle new operating conditions, where the performance of the closed loop control system may decrease significantly [114]. Accordingly, T2-FLCs can be considered as an alternative to solve the limitation of T1-FLCs. T2-FLCs have been proposed for mobile robot control problems [114, 254, 293–295], where better tracking performance compared to T1-FLCs were reported.

Nevertheless, these conventional T2-FLCs have a static structure and lack the ability to evolve their structure in an online manner. Generating type-2 fuzzy rules and their associated membership functions is a potential challenge especially for systems with many variables [35,36,169]. Employing evolutionary algorithms to find optimal fuzzy parameters is not desirable due to the large population space which results in slow performance. Hence, the development of a type-2 evolving fuzzy control system (T2-EFCS) is essential to facilitate self-learning (either from scratch or from a certain predefined rule). As a recap, EFSs can be defined as adaptive intelligent systems, which can learn their fuzzy structure and parameters simultaneously in an online manner [10, 55, 296, 297]. Recently, EFSs have become popular in various engineering applications from system identification and regression problems to classification and control problems [28, 37–49, 51, 298].

T2-EFCS has two phases, namely, structure learning and parameters learning. The structure of T2-EFCS does not require previous information about the fuzzy structure, and it can start the construction of its rules from scratch with only one rule. The rules are then added and pruned in an online fashion to achieve the desired set-point. In this chapter, a novel technique is deployed to control an unmanned ground vehicle (UGV) in the presence of multiple external disturbances, where the research outcomes in this chapter demonstrate the robustness of the proposed control systems as will be presented in the following sections.

The remainder of this chapter is structured as follows. The contributions of this chapter are listed in Section 6.2. Section 6.3 presents the proposed type-1 evolving fuzzy-based control system (T1-EFCS) design. This is followed by a novel design of an evolving type-2 fuzzy logic controller (T2-EFCS) in Section 6.4 including the simulation results. Lastly, Section 6.5 provides a summary of this chapter.

## 6.2 Contributions of this chapter

Inspired by the previous work in EFSs, the contributions of this chapter can be summarized as follows:

1. First, a novel type-1 evolving fuzzy-based control system (T1-EFCS) is developed, where the structure of the proposed controller does not require previous information about the fuzzy structure and it starts its construction from scratch with only one

rule. The rules are then added and deleted in an online manner to achieve the desired trajectory. The adaptive law for tuning its parameters is derived using the SMC theory, making the system robust to variations in system parameters and external disturbance. The proposed technique is investigated in the presence of external disturbances.

2. Second, to achieve better tracking performance over T1-EFCS, a novel evolving type-2 fuzzy logic controller (T2-EFCS) is introduced, where a new adding and deleting mechanism of its type-2 fuzzy rules is proposed. The adaptive law is derived using the SMC theory to guarantee robustness against uncertainties. The proposed closed-loop control system is employed to control a simulated mobile robot, where its robustness is investigated in the presence of external disturbance (e.g., in the face of measurement noise and external disturbance). Besides, a rigorous comparative study of T2-EFCS is performed with respect to three different controllers such as T1-FLC, T2-FLC, and T1-EFCS, where the outcomes of this study show the superiority of the proposed method with lower RMSE values. Lastly, the stability analysis of the proposed method is implemented using the Lyapunov stability theory.

## 6.3 T1-EFCs Design

### 6.3.1 Architecture of T1-EFCs

In this section, the notation used to represent FLS is described. From previous chapters, the Takagi-Sugeno fuzzy logic structure has the following general representation:

$$\begin{aligned} RULE^m:\ If\ (x_1\ is\ A_1^m,\ x_2\ is\ A_2^m, \ldots,\ x_n\ is\ A_n^m), \\ THEN\ y^m = a_0^m + a_1^m x_1 + a_2^m x_2 + \ldots + a_n^m x_n, \end{aligned}$$

where $x = [x_1, \ldots, x_n]$ are the inputs to the fuzzy controller, $m = 1, 2, .., M$, where $M$ is the number of fuzzy rules, $A_n^m$ denotes the membership functions, $y_m$ is the fuzzy output,

$a_n^m$ is the adjustable consequent parameters. For each input, the degree of membership function $A_n^m$ is $\mu_{m,n}(x_n)$ and the firing strength of every rule is given as follows:

$$f_m = A_1^m(x_1) \times \ldots \times A_n^m(x_n). \tag{6.1}$$

The output of the TS fuzzy controller can be represented as follows:

$$\begin{aligned} Y_{ts} &= \frac{\sum_{m=1}^{M} f_m \cdot y_m}{\sum_{m=1}^{M} f_m} \\ &= \frac{\sum_{m=1}^{M} \left(f_m \cdot \left[\sum_{m=1}^{M} a_m x_m + a_0\right]\right)}{\sum_{m=1}^{M} f_m} \end{aligned} \tag{6.2}$$

This can be written as:

$$Y_{ts} = \sum_{m=1}^{M} \widetilde{f}_m \cdot y_m, \tag{6.3}$$

where $\widetilde{f}_m = \frac{f_m}{\sum_{m=1}^{M} f_m}$.

The Gaussian membership function is represented as:

$$A_i = exp\left(\frac{\parallel x_i - c_i \parallel^2}{2\sigma_i^2}\right), \tag{6.4}$$

where $c_j$ represents the center of the Gaussian function and $\sigma_i$ is the width vector of the Gaussian function. The input vector to our controller are $x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$, where $(e, \dot{e})$ is the error and its derivative.

Our fuzzy controller has the ability to add new rules, prune the existing rules, and learn the fuzzy parameters. Figs. 6.1 depicts the structure of our proposed model-free self-evolving controller.

### 6.3.2 T1-EFCs Structure Learning

#### 6.3.2.1 Rule adding mechanism

The first data sample of the $e$ and $\dot{e}$ are used to generate the first rule of the fuzzy controller. Since we are considering the error and its derivative as the inputs to our controller, two

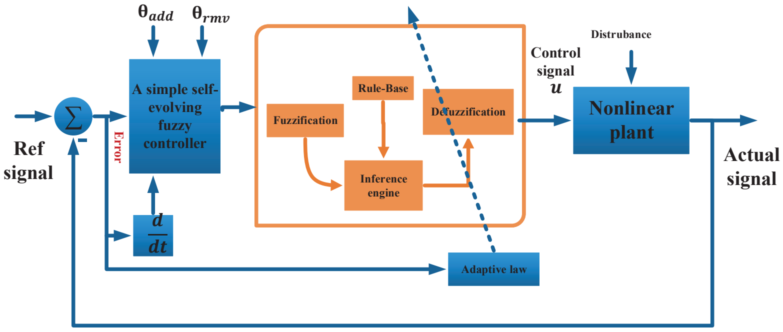


Figure 6.1: A simplified model-free self-evolving controller structure

membership functions are generated initially.

For a new incoming data $x(t)$, we calculate the Euclidean distance using (6.5).

$$\begin{aligned} D(x,c) &= \sqrt{\textstyle\sum_{i=1}^{n}(x_i-c_i)^2}, \\ &= \parallel x_i - c_i \parallel \end{aligned}, \tag{6.5}$$

where $c_i \in R$ is the nearest vector center to the current input state $x(t)$, which constructs the Gaussian membership function. For a new incoming data $x(t)$, the proposed controller generates a new rule if the following condition is satisfied:

$$\parallel x_i - c_i \parallel_{min} > d_{add}, \tag{6.6}$$

where $d_{add} > 0$ is a predefined thresholds value $\in (0,1)$. Once a new rule $n = n + 1$ is constructed, a corresponding fuzzy set for each input variable is generated. The initial fuzzy parameters of the newly added fuzzy rule are as follows:

$$\begin{cases} a_{n+1} &= & 0 \\ c_{n+1} &= & x_i \\ \sigma_{n+1} &= & \alpha \parallel x_i - c_i \parallel \end{cases} \tag{6.7}$$

where $\alpha$ is an overlap value between different rules and $a_n$ is the adjustable consequent parameter as explained in section 6.3.

**Remark 5** *the threshold value is confined in* $(0,1)$ *based on the fuzzy set reduction method in [147]. If the threshold value* $d_{add}$ *is set to one, then the number of fuzzy sets in each* *input variable is equal to the number of rules. Conversely, if the threshold value $d_{add}$ is set to zero, then only one fuzzy set is generated in each input variable [147].*

#### 6.3.2.2 T1-EFCs Rule Pruning Mechanism

Some existing methods in the literature have the ability to generate rules. However, once the rules are generated, they cannot be deleted. In this proposed technique, deleting unnecessary rules is considered. The process of pruning existing rules is based on the contribution of membership grade, so when it is smaller than the prior threshold value, the rule is deleted. The rule pruning is done if the following condition is satisfied:

$$f_m < d_{del} \tag{6.8}$$

where $f_m$ is the firing strength and can be calculated using Equation (6.1), $d_{del}$ is a predefined thresholds value $\in (0, 1)$. Hence, the redundant fuzzy rule is removed, hence, the fuzzy parameters are as follows:

$$\begin{cases} a &= \phi \\ c &= \phi \\ \sigma &= \phi \\ n &= n-1. \end{cases} \tag{6.9}$$

### 6.3.3 T1-EFCs Parameters Learning

The parameters of our fuzzy controller are tuned based on the SMC theory. Let the desired output be $y_d$ and the actual output be $y$, where the error and its derivative can be defined as:

$$e(t) = y - y_d, \quad \dot{e} = \dot{y} - \dot{y}_d. \tag{6.10}$$

We design the sliding surface function as follows:

$$s(t) = \delta e(t) + \dot{e}, \tag{6.11}$$

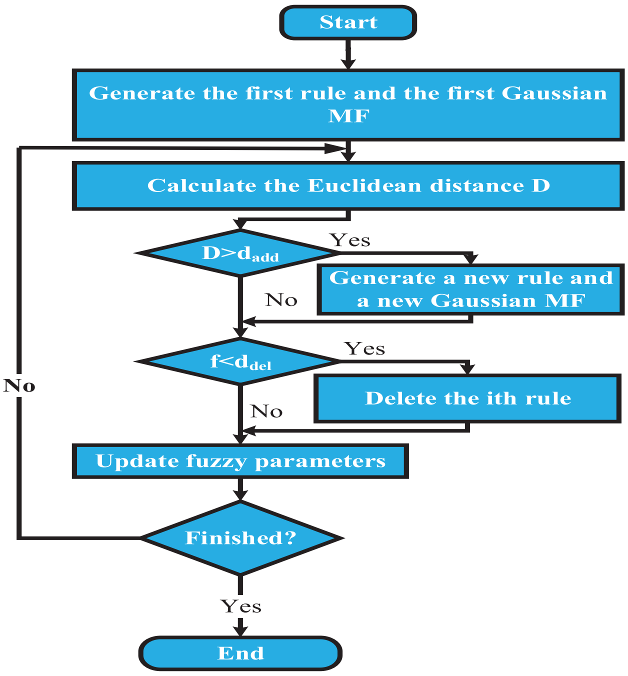

Start
Generate the first rule and the first Gaussian MF
Calculate the Euclidean distance D
D>d_add
Yes
Generate a new rule and a new Gaussian MF
No
f<d_del
Yes
Delete the ith rule
No
No
Update fuzzy parameters
Finished?
Yes
End

Figure 6.2: Flowchart of the evolving fuzzy controller

$$\dot{s}(t) = \ddot{e} + \delta\dot{e}, \tag{6.12}$$

where $\delta$ must be $> 0$. The adaptive law to minimize $s\dot{s}$ with respect to the consequent parameters $a_m$ can be derived using the gradient descent method as follows [30]:

$$a_{n+1} = a_n + \tau\frac{\partial s(t)\dot{s}(t)}{\partial a_n(t)}, \tag{6.13}$$

where $\tau$ is the learning rate. Equation (6.13) can be rewritten as follows:

$$\begin{aligned} a_{n+1} &= a_n + \tau\frac{\partial s(t)\dot{s}(t)}{\partial u(t)}\frac{\partial u(t)}{\partial a_n(t)} \\ &\vdots \\ &= a_n + \tau s(t)\frac{f_m(x)}{\sum_{i=1}^{m} f_m(x)}. \end{aligned} \tag{6.14}$$

In our design, $\tau = 2$ and $\delta = 0.05$. The flowchart of our proposed controller is described in Fig. 6.2.

### 6.3.4 System Description

Consider a single inverted pendulum system with a disturbance signal in Fig. 6.3, where the dynamic equation can be described as follows:

$$\dot{x}_1 = x_2, \tag{6.15}$$

$$\dot{x}_2 = f(x) + g(x)u + d(t), \tag{6.16}$$

$$f(x) = \frac{gsinx_1 - mlx_2^2cosx_1sinx_1/(m_{cart}+m)}{l(4/3 - mcos^2x_1/(m_{cart}+m))}, \tag{6.17}$$

$$g(x) = \frac{cosx_1(m_{cart}+m)}{l(4/3 - mcos^2x_1/(m_{cart}+m))}. \tag{6.18}$$

The angle and its angular speed are $x_1$ and $x_2$; while $u$ is the control signal; and $d(t)$ is an external disturbance. The parameters of the single pendulum are given as: $g = 9.8m/s^2$; the mass of the pendulum $m = 0.1kg$; the mass of the cart $m_{cart} = 1kg$; and the half length of the pendulum $l = 0.5m$.

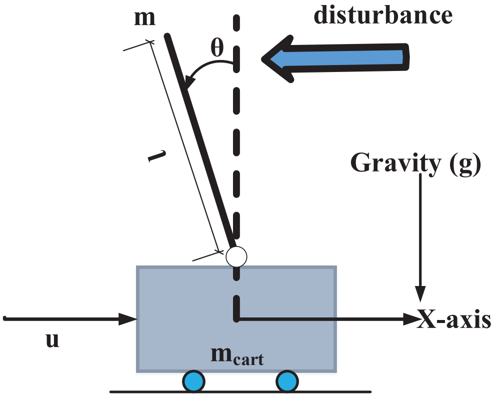


Figure 6.3: Single inverted pendulum plant

#### 6.3.4.1 T1-EFCs Results and Discussion

In this section, the system performance is investigated in two different scenarios: under the nominal condition and in the presence of an external disturbance. The simulation time is 20 *sec*, with a fixed step size of $0.01s$. The initial values of the consequent parameters are zeros. In this proposed design, the distance indicating the similarity measure threshold value is 0.1, while the firing strength threshold value is 0.2.

### 6.3.5 System performance under nominal condition

The performance of the proposed evolving controller is evaluated under a nominal condition. It is compared with an adaptive fuzzy controller [299]. The design of the adaptive fuzzy controller has five rules and two Gaussian membership functions for each input. However, for the proposed evolving controller, only two rules are evolved to track the desired trajectory. It can be seen that a lower number of rules are obtained from the evolving controller compared to the adaptive fuzzy controller. Also, better tracking results are obtained from the proposed controller as shown in Figs. 6.4 & 6.5 with less RMSE values as summarized in Table 1. The error, number of fuzzy rules, and the control signals under nominal conditions are plotted in Fig. 6.8.

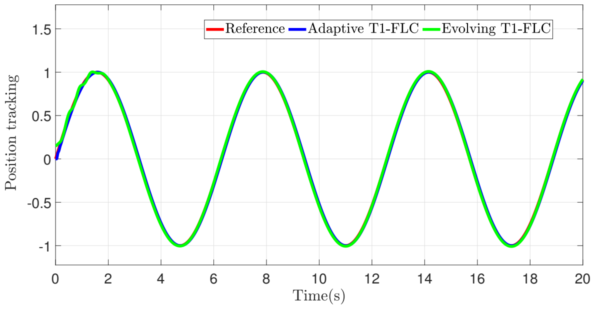


Figure 6.4: Position tracking in normal condition

#### 6.3.5.1 System Performance Under Uncertainties

For robustness analysis, the performance of the proposed controller is studied in the presence of external disturbance, where a disturbance signal $d = 0.5cos(t)$ is injected into the simulated model. As can be seen from Figs. 6.6 & 6.7, the proposed evolving controller can handle the disturbance within a reasonable period of time. The sliding surface and the

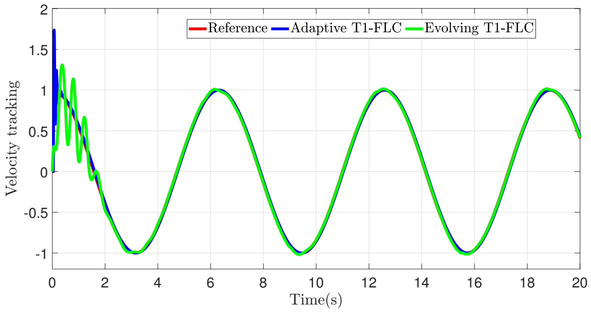


Figure 6.5: Velocity response in normal condition

Table 6.1: RMSE values for tracking a sine wave reference

| Controller Type | Nominal condition | with disturbance |
|---|---|---|
| Adaptive fuzzy | 0.0124 | 0.0928 |
| Evolving Fuzzy | **6.5694e-04** | **0.0195** |

adaptation of the consequent parameters with time in the case of uncertainties are plotted in Fig. 6.10. The error, number of fuzzy rules, and the control signals under uncertainties are plotted in Fig. 6.9.

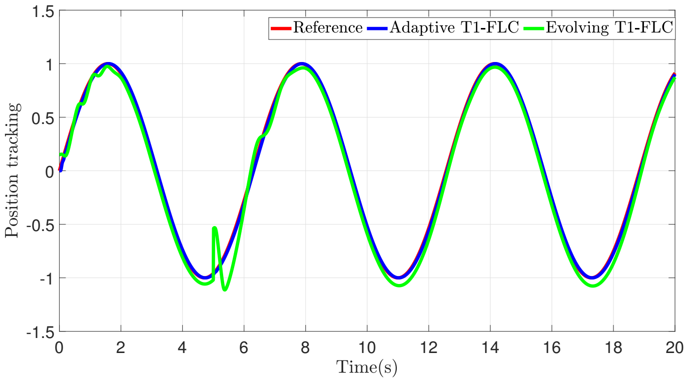


Figure 6.6: Position tracking in the presence of uncertainties

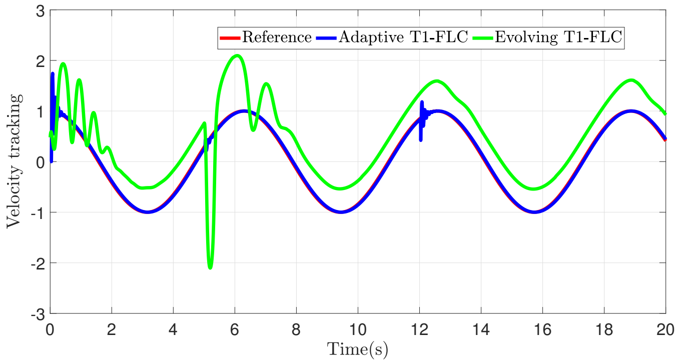


Figure 6.7: Velocity response in the presence of uncertainties

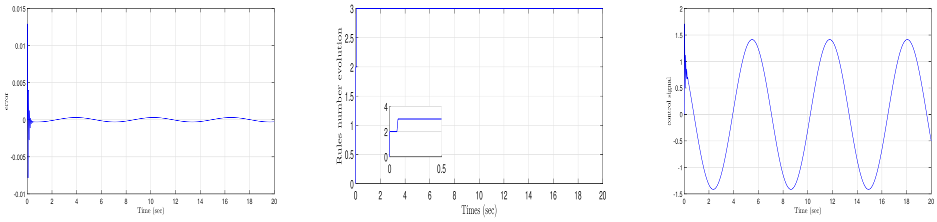


Figure 6.8: Performance under nominal condition (a) Error, (b) Number of evolving fuzzy rules, (c) Control signal

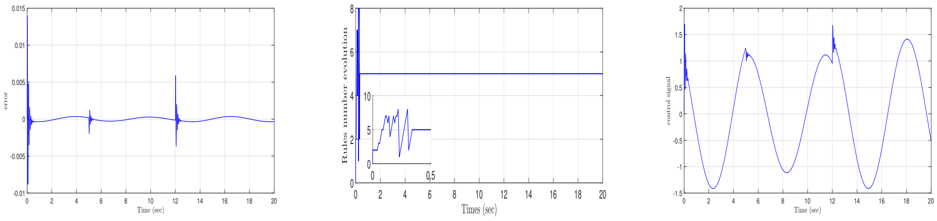


Figure 6.9: Performance under uncertain condition (a) Error, (b) Number of evolving fuzzy rules, (c) Control signal

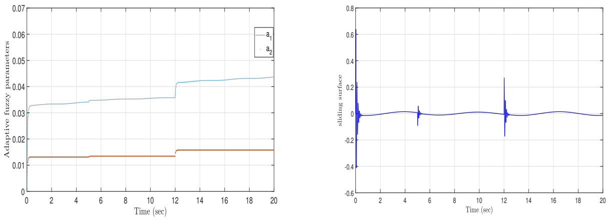


Figure 6.10: Performance under uncertain condition (a) Adaptive parameters in the presence of uncertainties, (b) Sliding surface changes with time in the presence of uncertainties.

## 6.4 T2-EFCS Design for a Mobile Robot

### 6.4.1 Problem Formulation

A class of a $n-th$ order nonlinear dynamic system can be described as follows:

$$\begin{cases} x^{(n)} = \underline{f}(\underline{x}) + \underline{b}(\underline{x})u + d(\underline{x}) \\ y = x \end{cases} \tag{6.19}$$

where the state vector $\underline{x} \in \Re^m$ can be defined as $\underline{\boldsymbol{x}} = \begin{bmatrix} x & \dot{x} & \ldots & x^{(n-1)} \end{bmatrix}^T \in \Re^m$; $x \in \Re^m$represents the state; $\underline{f}(\underline{x}) \in \Re^m$ and $\underline{b}(\underline{x}) \in \Re^{m\times m}$ are the system nonlinear functions; $u$ depicts the control input; $y$ is the system output; and $d(\underline{x}) \in \Re^m$ expresses unknown uncertainties. For the theoretical study, it is assumed that the nonlinear terms in (6.19) are known, bounded and $\underline{b}^{-1}(\underline{x})$ exists for all $x$.

**Assumption 1** *If there exist inevitable modeling uncertainties between the real system and the simulated systems, they can be absorbed in the uncertainty function.*

The purpose of the closed-loop control system is that the system output $y$ has the ability to track a desired signal $y_d$. Consider the tracking error as

$$\boldsymbol{e} = \boldsymbol{y_d} - \boldsymbol{y}, \tag{6.20}$$

and the system tracking vector can be written as

$$\underline{e} = \begin{bmatrix} e(t) & \dot{e}(t) & \ldots & e(t)^{(n-1)} \end{bmatrix}^T \in \Re^{nm}. \tag{6.21}$$

Define an integrated sliding surface as:

$$S_{EFC} \equiv e^{n-1} + \kappa_1 e^{n-2} + \ldots + \kappa_n \int_0^t e(\tau)d\tau, \tag{6.22}$$

where $\kappa_i \in \Re^{m\times m}$ is a strictly positive constant matrix, where $i = 1, 2, 3\ldots, n$; and $\kappa = [\kappa_1, .., \kappa_n]^T \in \Re^{nm\times m}$. If the nonlinear terms $\underline{f}(\underline{x})$ and $\underline{b}(\underline{x})$ and also the $d(\underline{x})$ are known, an ideal control law $u_{fin}(t)$ can be designed as follows [156, 174, 177]:

$$u_{fin} = \underline{b}(\underline{x})^{-1} \left[ y_d^{(n)} - \underline{f}(\underline{x}) - d(\underline{x}) + \kappa^T \underline{e} \right]. \tag{6.23}$$

Using (6.23) and (6.19), we can derive the following error dynamic equation as:

$$\dot{S}_{EFC} = e^{(n)} + \kappa^T \underline{e} = 0 \tag{6.24}$$

From (6.24), it is clear that if $\kappa$ is chosen to correspond to the Hurwitz polynomial coefficients, it leads to a convergence of the tracking error to zero when the time approaches infinity [174]. Nevertheless, in real-time applications, the uncertainty term $d(\underline{x})$ cannot be precisely known. Therefore, the ideal control law in (6.23) is not available. Hence, the proposed closed-loop control system is designed to meet the desired control objective.

### 6.4.2 T2-EFCS control system design

#### 6.4.2.1 T2-EFCS Architecture

The structure of the proposed evolving interval type-2 fuzzy system is discussed in this section. It consists of five layers, namely, the input layer, the fuzzification layer, the firing strength layer, the consequent layer, and the output layer as shown in Fig. 6.11. Each rule has the following form:

$$\begin{aligned} RULE^m:\ If\ (x_1\ is\ \tilde{X}_1^m,\ x_2\ is\ \tilde{X}_2^m, \ldots,\ x_n\ is\ \tilde{X}_n^m),\\ THEN\ \tilde{y}_m = \sum_{n=1}^{M} W_m^n(t) x_n(t),\end{aligned}$$

where $x = x_1, \ldots, x_n$ denotes the inputs variables to T2-EFCS, $m = 1, 2, .., M$; where $M$ represents the number of fuzzy rules, $\tilde{X}_n^m$ and $\tilde{y}$ are the IT2F membership functions for the input and output, respectively. Each layer can be described as follows:

- **Layer 1 (Input layer)**: this layer is the input signals with $(n \times 1)$ vector. In this study, the error and its derivative are the two inputs to the system.
- **Layer 2 (Fuzzification layer)**: this layer is first hidden layer, which can be expressed by IT2 membership functions. In this study, a Gaussian MF with fixed width

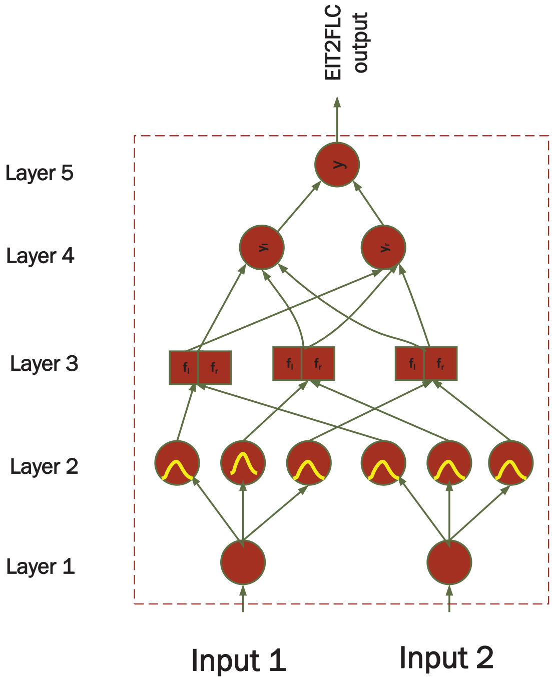


Figure 6.11: Structure of EIT2FLCs

$\sigma_n^m$ and uncertain means $[\overline{m}_{mn}, \underline{m}_{mn}]$ is deployed, which can be given as follows:

$$\overline{\mu}_{mn} = exp\left\{-\frac{1}{2}\left(\frac{x_n - \overline{m}_{mn}}{\sigma_{mn}}\right)^2\right\}, \tag{6.25}$$

$$\underline{\mu}_{mn} = exp\left\{-\frac{1}{2}\left(\frac{x_n - \underline{m}_{mn}}{\sigma_{mn}}\right)^2\right\}, \tag{6.26}$$

- **Layer 3 (Firing layer)**: the firing strength $F_m$ is computed in this layer to perform the aggregation operation.

$$F_m = [\underline{f}_m, \overline{f}_m] \tag{6.27}$$

where

$$\begin{cases} \underline{f}_m = \prod_{n=1}^{p} \underline{\mu}_{mn} \\ \overline{f}_m = \prod_{n=1}^{p} \overline{\mu}_{mn} \end{cases}$$

- **Layer 4 (Consequent layer)**: the output of this layer has two consequent values as follows:

$$\begin{cases} \underline{y}(t) = \dfrac{\sum_{m=1}^{M} \underline{f}_m(t) y_m(t)}{\sum_{n=1}^{M} \underline{f}_m(t)} \\ \overline{y}(t) = \dfrac{\sum_{m=1}^{M} \overline{f}_m(t) y_m(t)}{\sum_{m=1}^{M} \overline{f}_m(t)} \end{cases} \tag{6.28}$$

$$y_m(t) = \sum_{n=1}^{M} W_{nm}(t) x_n(t) \tag{6.29}$$

where $\overline{y}(t)$ and $\underline{y}(t)$are the upper and lower outputs of the consequent part, respectively; $y_m(t)$ represents T1-FSs consequent parameters. In this layer, the center-of-sets and the 'Enhanced Iterative Algorithm with Stop Condition' type-reducer were utilized to calculate the interval outputs $[\overline{y}(t), \underline{y}(t)]$.

- **Layer 5 (Output layer)**: the computation of the output value of the last layer is given as follows:

$$y = \varrho \overline{y}(t) + (1 - \varrho) \underline{y}(t) \tag{6.30}$$

where $\varrho \in [0, 0.5]$ represents a weighting parameter.

#### 6.4.2.2 T2-EFCS Structure Learning

- **Rule Adding Mechanism**: The rule generation of T2-EFCS is based on the distance between the incoming data and the upper and lower means of the type-2 Gaussian function, so that when max $D_{T2} > T_{add}$, a new rule is generated. The Euclidean distance of the upper and lower means can be computed using the following equation:

$$D_{T2}(x_n, m_M) \quad = \quad \| x_n - m_M \|_2 \ , \tag{6.31}$$

where $x_n$ represents the incoming data of $e$ and $\dot{e}$; $m_M = ((\overline{m}_M + \underline{m}_M)/2)$ is the mean. If we define a MAX-MIN approach to identify when to add a new type-2 fuzzy rule as, $\hat{M} = \arg \underset{1 \leq M \leq n_M}{min} D(x_n, m_M)$, the T2-EFCS finds:

$$\textit{If} \quad (\max \ D_{T2} > T_{add}) \quad \text{THEN} \quad \textit{Generate} \text{ new type-2 fuzzy rule} \tag{6.32}$$

where $T_{add}$ denotes a prior threshold value for rule generation, and $\hat{M}$ is the mathematical representation for the argument of the minimum.

The initial type-2 fuzzy MF parameters are set as:

$$\begin{cases} [m_{1n}^1, m_{1n}^2] & = [x_n - \Delta x, x_n + \Delta x] \\ \sigma & = \sigma_{fixed} \end{cases} \tag{6.33}$$

where $\sigma_{fixed}$ denotes a predefined value (in this work, $\sigma_{fixed} = 0.5$), which determine the width of the membership functions associated with a new generated rule.

Once a new type-2 fuzzy rule is generated, the same procedure implemented for the first rule is utilized to assign the uncertain mean and the width as follows:

$$\begin{cases} \left[m_{1n}^{M(t)+1}, m_{1n}^{M(t)+2}\right] & = [x_n - \Delta x, x_n + \Delta x] \\ \sigma^{M(t)+1} & = \xi . \left| x_n - \left(\frac{m_{12}^I + m_{12}^I}{2}\right)\right| \end{cases} \tag{6.34}$$

where $\xi$ represents an overlapping parameter ($\xi$ is set to 0.5 in this work); and the $M(t)$ represents the total number of type-2 fuzzy rules at the $t^{th}$ step.

**Remark 6** *If the uncertainty associated with the mean $\Delta x$ is very small, the type-2 GF becomes similar to type-1 GF. Nevertheless, if the uncertain region of type-2 GF with is extremely large, it covers all input domains, where a lower number of rules is generated [109, 156].*

- **Rule Pruning Mechanism**: In this proposed technique, deleting unnecessary rules is considered. The process of pruning existing rules is based on the contribution of membership grade so when it is smaller than the prior threshold value, the rule is deleted. This approach can be expressed as follows:

$$\textit{If} \quad (F_m < T_{del}) \quad \textit{THEN} \quad \text{delete type-2 fuzzy rule} \tag{6.35}$$

where $T_{del}$ denotes a prior threshold value for rule deletion; and $F_{T2} = F_m$ which denotes the firing strength in (6.27) for each incoming data.

Automatic rules generation and pruning is efficient, which determine the optimum number of fuzzy rules. Fig. 6.12 illustrates the flowchart of the proposed method. The online updates of type-2 fuzzy parameters is presented in the following section.

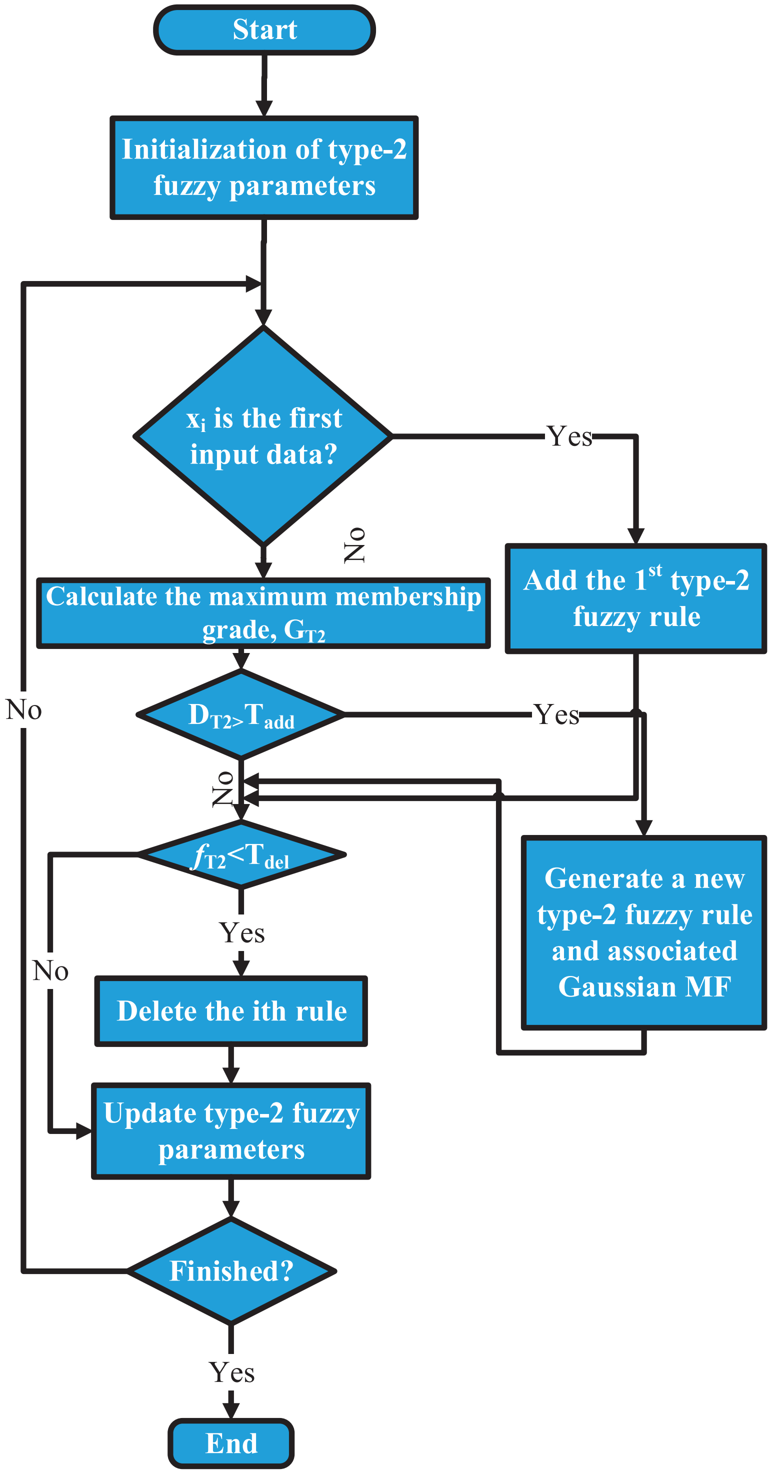

Start
Initialization of type-2 fuzzy parameters
xᵢ is the first input data?
Yes
No
Add the 1st type-2 fuzzy rule
Calculate the maximum membership grade, G_T2
No
D_T2>T_add
Yes
No
Generate a new type-2 fuzzy rule and associated Gaussian MF
f_T2<T_del
Yes
No
Delete the ith rule
Update type-2 fuzzy parameters
Finished?
Yes
End

Figure 6.12: Flowchart of the T2-EFCS

**Remark 7** *The selection of the adding/pruning threshold parameters $T_{add}$ and $T_{del}$ is based on the maximum number of fuzzy rules (MNFR), where MNFR is a design parameter that determines the maximum number of fuzzy rules when a fuzzy structure evolves. Similar to 5, these threshold values are confined in (0,1). The empirical relationship between the adding/pruning thresholds values ($T_{add}$, $T_{del}$) and MNFR is described as: $[T_{add} \geq (1/MNFR)]$ and $[T_{del} \geq (1/MNFR)]$. This way, the parameters of the T2-EFCS are meaningful to the user. For instance, if MNFR is selected to be 7, then $T_{add}$ and $T_{del}$ should be selected $\geq 0.143$. If MNFR is selected to be 15, then $T_{add}$ and $T_{del}$ should be $\geq 0.067$. Higher MNFR values result in higher accuracy, but also higher processing time. The selection of MNFR is a trade-off between the processing unit capabilities and controllers' accuracy [49].*

#### 6.4.2.3 T2-EFCS Parameters Learning

The gradient descent method is applied to minimize the error function between the desired and the actual output. The online adaptation law of the proposed T2-EFCS is given in the following equations as [156, 177]:

$$\hat{\overline{m}}_{mn}(t+1) = \hat{\overline{m}}_{mn}(t) - \hat{\eta}_m \frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{\overline{m}}_{mn}} \tag{6.36}$$

$$\hat{\underline{m}}_{mn}(t+1) = \hat{\underline{m}}_{mn}(t) - \hat{\eta}_m \frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{\underline{m}}_{mn}} \tag{6.37}$$

$$\hat{\sigma}_{mn}(t+1) = \hat{\sigma}_{mn}(t) - \hat{\eta}_\sigma \frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{\sigma}_{mn}} \tag{6.38}$$

$$\hat{W}_{nk}(t+1) = \hat{W}_{nk}(t) - \hat{\eta}_W \frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{W}_{nk}} \tag{6.39}$$

where $\hat{\eta}_m$, $\hat{\eta}_\sigma$, and $\hat{\eta}_W$ are the learning rates to update T2-EFCS parameters. By applying the chain rule, the following equations can be derived:

$$\begin{aligned} \frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{\overline{m}}_{mn}} &= \frac{1}{2}\frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{y}_{T2-EFCS}} \left( \frac{\partial \underline{y}}{\partial \underline{f}_m} \frac{\partial \underline{f}_m}{\partial \hat{\overline{m}}_{mn}} + \frac{\partial \overline{y}}{\partial \overline{f}_m} \frac{\partial \overline{f}_m}{\partial \hat{\overline{m}}_{mn}} \right) \\ &= \frac{1}{2} s_{EFC}(t) \left( \frac{(w_m - \underline{y})}{\sum_{m=1}^{M} \underline{f}_m} \frac{\partial \underline{f}_m}{\partial \hat{\overline{m}}_{mn}} + \frac{(w_m - \overline{y})}{\sum_{m=1}^{M} \overline{f}_m} \frac{\partial \overline{f}_m}{\partial \hat{\overline{m}}_{mn}} \right) \end{aligned} \tag{6.40}$$

$$\begin{aligned}\frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \underline{\hat{m}}_{mn}} &= \frac{1}{2}\frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{y}_{T2-EFCS}}\left(\frac{\partial \underline{y}}{\partial \underline{f}_m}\frac{\partial \underline{f}_m}{\partial \underline{\hat{m}}_{mn}} + \frac{\partial \overline{y}}{\partial \overline{f}_m}\frac{\partial \overline{f}_m}{\partial \underline{\hat{m}}_{mn}}\right)\\ &= \frac{1}{2}s_{EFC}(t)\left(\frac{\left(w_m-\underline{y}\right)}{\sum_{m=1}^{M}\underline{f}_m}\frac{\partial \underline{f}_m}{\partial \underline{\hat{m}}_{mn}} + \frac{(w_m-\overline{y})}{\sum_{m=1}^{M}\overline{f}_m}\frac{\partial \overline{f}_m}{\partial \underline{\hat{m}}_{mn}}\right)\end{aligned} \tag{6.41}$$

$$\begin{aligned}\frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{\sigma}_{mn}} &= \frac{1}{2}\frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{y}_{T2-EFCS}}\left(\frac{\partial \underline{y}}{\partial \underline{f}_m}\frac{\partial \underline{f}_m}{\partial \hat{\sigma}} + \frac{\partial \overline{y}}{\partial \overline{f}_m}\frac{\partial \overline{f}_m}{\partial \hat{\sigma}}\right)\\ &= \frac{1}{2}s_{EFC}(t)\left(\frac{\left(w_m-\underline{y}\right)}{\sum_{m=1}^{M}\underline{f}_m}\frac{\partial \underline{f}_m}{\partial \hat{\sigma}} + \frac{(w_m-\overline{y})}{\sum_{m=1}^{M}\overline{f}_m}\frac{\partial \overline{f}_m}{\partial \hat{\sigma}}\right)\end{aligned} \tag{6.42}$$

$$\begin{aligned}\frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{W}_{nk}} &= \frac{1}{2}\frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial \hat{y}_{T2-EFCS}}\frac{\partial \hat{y}_{T2-EFCS}}{\partial y}\frac{\partial y}{\partial W}\\ &= \frac{1}{2}s_{EFC}(t)\left(\frac{\underline{f}_m}{\sum_{m=1}^{M}\underline{f}_m} + \frac{\overline{f}_m}{\sum_{m=1}^{M}\overline{f}_m}\right)\end{aligned} \tag{6.43}$$

where

$$\begin{aligned}\frac{\partial \underline{f}_m}{\partial \hat{\overline{m}}_{mn}} &= \frac{\partial \underline{f}_m}{\partial \underline{\mu}_{mn}}\frac{\partial \underline{\mu}_{mn}}{\partial \hat{\overline{m}}_{mn}} &&= \underline{f}_m \quad \frac{\left(x_n-\hat{\overline{m}}_{mn}\right)^2}{\left(\hat{\sigma}_{mn}\right)^3}\\ \frac{\partial \overline{f}_m}{\partial \hat{\overline{m}}_{mn}} &= \frac{\partial \overline{f}_m}{\overline{\mu}_{mn}}\frac{\overline{\mu}_{mn}}{\partial \hat{\overline{m}}_{mn}} &&= \overline{f}_m \quad \frac{\left(x_n-\hat{\overline{m}}_{mn}\right)^2}{\left(\hat{\sigma}_{mn}\right)^3}\\ \frac{\partial \underline{f}_m}{\partial \underline{\hat{m}}_{mn}} &= \frac{\partial \underline{f}_m}{\partial \underline{\mu}_{mn}}\frac{\partial \underline{\mu}_{mn}}{\partial \underline{\hat{m}}_{mn}} &&= \underline{f}_m \quad \frac{\left(x_n-\underline{\hat{m}}_{mn}\right)^2}{\left(\hat{\sigma}_{mn}\right)^3}\\ \frac{\partial \overline{f}_m}{\partial \underline{\hat{m}}_{mn}} &= \frac{\partial \overline{f}_m}{\overline{\mu}_{mn}}\frac{\overline{\mu}_{mn}}{\partial \underline{\hat{m}}_{mn}} &&= \overline{f}_m \quad \frac{\left(x_n-\underline{\hat{m}}_{mn}\right)^2}{\left(\hat{\sigma}_{mn}\right)^3}\\ \frac{\partial \underline{f}_m}{\partial \sigma_{\hat{m}n}} &= \frac{\partial \underline{f}_m}{\partial \underline{\mu}_{mn}}\frac{\partial \underline{\mu}_{mn}}{\partial \hat{\sigma}} &&= \underline{f}_m \quad \frac{\left(x_n-\underline{\hat{m}}_{mn}\right)^2}{\left(\hat{\sigma}_{mn}\right)^3}\\ \frac{\partial \overline{f}_m}{\partial \hat{\sigma}_{mn}} &= \frac{\partial \overline{f}_m}{\partial \overline{\mu}_{mn}}\frac{\overline{\mu}_{mn}}{\partial \hat{\sigma}} &&= \overline{f}_m \quad \frac{\left(x_n-\hat{\overline{m}}_{mn}\right)^2}{\left(\hat{\sigma}_{mn}\right)^3}\end{aligned} \tag{6.44}$$

#### 6.4.2.4 T2-EFCS Robustness Term

Similar to the work in chapter 5, for obtaining a robust control performance in the face of uncertainties, a disturbance elimination term, $u_{robust}$, is added to the final control input. Therefore, the final control input can be expressed as:

$$u_{total} = u_{fin} + u_{robust}, \tag{6.45}$$

where $u_{fin}$ is obtained using Eq. (6.23) and $u_{robust}$ is a robustifying term [287,288], which can be represented as follows:

$$u_{robust} = \beta \; sat(S_{EFC}(t)), \tag{6.46}$$

where $\beta$ denotes a design parameter and $sat$ can be defined as follows:

$$sat\left(S_{EFC}, \iota\right) = \begin{cases} \frac{S_{EFC}}{\iota}, & if \;\; |S_{EFC}| \leq |\iota| \\ sgn(S_{EFC}), & otherwise \end{cases} \tag{6.47}$$

where $\iota$ is a design factor representing the thickness of the boundary layer. The *sgn* represents the *signum* function, which can be defined as follows:

$$sgn(S_{EFC}(t)) = \begin{cases} 1, & if\ S_{EFC}(t) > 0 \\ 0, & if\ S_{EFC}(t) = 0 \\ -1, & if\ S_{EFC}(t) < 0 \end{cases} \tag{6.48}$$

#### 6.4.2.5 T2-EFCS Stability Proof

The Lyapunov function is defined as [156, 177]:

$$V\left(s_{EFC}(t)\right) = \frac{1}{2} s_{EFC}^2(t) \tag{6.49}$$

$$\dot{V}\left(s_{EFC}(t)\right) = s_{EFC}(t)\dot{s}_{EFC}(t) \tag{6.50}$$

Following the derivation on [156], one can introduce a matrix such that $Q_{\mathcal{O}}(t) = \frac{\partial \hat{y}_{T2-EFCS}}{\partial O}$, for $O = \underline{\hat{m}}, \hat{\overline{m}}, \hat{\sigma}, \hat{W}$ where

$$\begin{aligned} Q_{\underline{\hat{m}}}(t) = \frac{\partial \hat{y}_{T2-EFCS}}{\partial \underline{\hat{m}}} = & \left[ \frac{\partial \hat{y}_{T2-EFCS}}{\partial \underline{\hat{m}}_{11}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \underline{\hat{m}}_{1n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \underline{\hat{m}}_{21}} \right. \\ & \left. \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \underline{\hat{m}}_{2n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \underline{\hat{m}}_{n_i n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \underline{\hat{m}}_{n_i n_j}} \right] \end{aligned}$$

$$\begin{aligned} Q_{\hat{\overline{m}}}(t) = \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\overline{m}}} = & \left[ \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\overline{m}}_{11}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\overline{m}}_{1n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\overline{m}}_{21}} \right. \\ & \left. \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\overline{m}}_{2n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\overline{m}}_{n_i n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\overline{m}}_{n_i n_j}} \right] \end{aligned}$$

$$\begin{aligned} Q_{\hat{\sigma}}(t) = \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\sigma}} = & \left[ \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\sigma}_{11}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\sigma}_{1n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\sigma}_{21}} \right. \\ & \left. \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\sigma}_{2n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\sigma}_{n_i n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{\sigma}_{n_i n_j}} \right] \end{aligned}$$

$$\begin{aligned} Q_{\hat{W}}(t) = \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{W}} = & \left[ \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{W}_{11}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{W}_{1n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{W}_{21}} \right. \\ & \left. \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{W}_{2n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{W}_{n_i n_j}}, \ldots, \frac{\partial \hat{y}_{T2-EFCS}}{\partial \hat{W}_{n_i n_j}} \right] \end{aligned}$$

By applying the gradient descent technique, (6.50) can be represented as

$$\begin{array}{rcl} \dot{V}\left(s_{EFC}(t+1)\right) & = & \dot{V}\left(s_{EFC}(t)\right)+\mathit{\Delta}\dot{V}\left(s_{EFC}(t)\right) \\ & \cong & \dot{V}\left(s_{EFC}(t)\right)+\left[\frac{\dot{V}(s_{EFC}(t))}{\partial O}\right]^T \Delta O \end{array} \tag{6.51}$$

where $\mathit{\Delta}\dot{V}\left(s_{EFC}(t)\right)$ represents the change in $\dot{V}\left(s_{EFC}(t)\right)$; $\Delta O$ is the change in $O$.

By utilizing the chain rule, the following equation can be derived as:

$$\frac{\partial\dot{V}\left(s_{EFC}(t)\right)}{\partial O}=\frac{\partial\dot{V}\left(s_{EFC}(t)\right)}{\partial\hat{y}_{T2-EFCS}}\frac{\partial\hat{y}_{T2-EFCS}}{\partial O}=\frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial\hat{y}_{T2-EFCS}}\frac{\partial\hat{y}_{T2-EFCS}}{\partial O} \tag{6.52}$$

and by utilizing (6.40-6.43) and (6.51), it yields to:

$$\frac{\partial\dot{V}\left(s_{EFC}(t)\right)}{\partial O}=-s_{EFC}(t)\frac{\partial\hat{y}_{T2-EFCS}}{\partial O}=-s_{EFC}(t)Q_{\mathcal{O}}(t) \tag{6.53}$$

where

$$\Delta O=-\hat{\eta}_O\frac{\partial s_{EFC}(t)\dot{s}_{EFC}(t)}{\partial O}=\hat{\eta}_O s_{EFC}(t)Q_{\mathcal{O}}(t) \tag{6.54}$$

By substituting (6.53), (6.54) into (6.51),

$$\begin{array}{rcl} \mathit{\Delta}\dot{V}\left(s_{EFC}(t)\right) & = & \left[\frac{\dot{V}(s_{EFC}(t))}{\partial O}\right]^T \Delta O \\ & = & \left[-s_{EFC}(t)Q_{\mathcal{O}}(t)*\hat{\eta}_O s_{EFC}(t)Q_{\mathcal{O}}(t)\right] \\ & = & -\hat{\eta}_O s_{EFC}^2(t)Q_{\mathcal{O}}(t) \end{array} \tag{6.55}$$

From (6.55), if $\hat{\eta}_O$ is selected as $\hat{\eta}_O > 0$ , it yields to $\mathit{\Delta}\dot{V}\left(s_{EFC}(t)\right) < 0$. Hence, the convergence of the proposed fuzzy parameters is guaranteed by the Lyapunov stability theory [156].

### 6.4.3 System Description for Autonomous Mobile Robot

The absolute position of the robot can be represented in the Cartesian plane, with respect to the global frame, by the following three variables as:

$$p=[x,y,\theta]^T \tag{6.56}$$

where $x$ and $y$ denotes the coordinates of the robot center of the mass, while $\theta$ represents the robot orientation, as illustrated in Fig. 6.13. The robot can be controlled by:

$$q=[\upsilon,w]^T \tag{6.57}$$

where $[v, w]$ are the linear and angular velocities, respectively. Therefore, the mobile robot kinematic model can be described by the following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ -sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \tag{6.58}$$

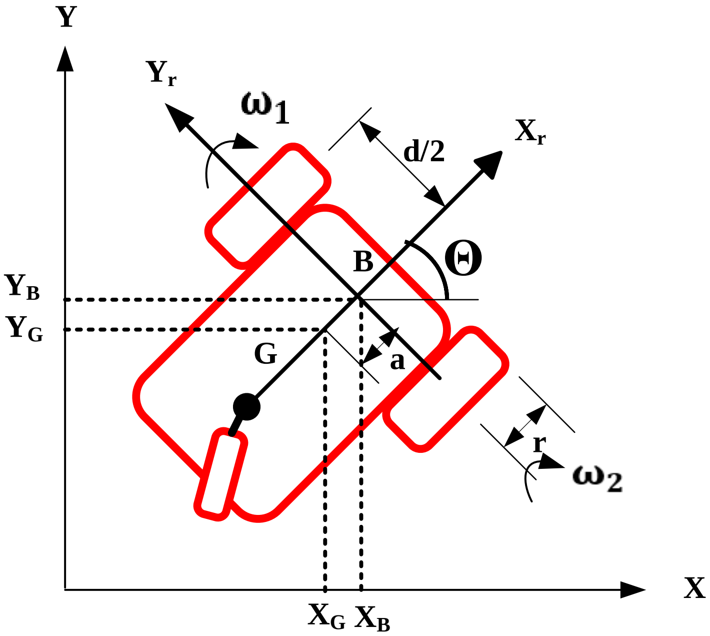


Figure 6.13: Kinematic model of the differential-drive mobile robot.

Our mobile robot belongs to the class of differential-drive ground robots. Hence, the linear and angular velocities can be described using the left $V_l$ and right $V_r$ wheels as follows: [114, 300]:

$$v = \frac{V_l+V_r}{2} \tag{6.59}$$

$$w = \frac{V_r-V_l}{d} \tag{6.60}$$

where $d$ is the distance between wheels. The robot rotation radius, $r$, can be computed as

$$r = \frac{d(V_l+V_r)}{2(V_r-V_l)} = \frac{v}{w} \tag{6.61}$$

Lastly, the full kinematic model of the autonomous mobile robot can be represented as follows:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \frac{r(dcos\theta - asin\theta)}{2d} & \frac{r(dcos\theta + asin\theta)}{2d} \\ \frac{r(dsin\theta - acos\theta)}{2d} & \frac{r(dsin\theta - acos\theta)}{2d} \\ \frac{r}{2d} & \frac{-r}{2d} \end{bmatrix} \begin{bmatrix} w_r \\ w_l \end{bmatrix}, \tag{6.62}$$

where$[w_r, w_l]$ are the right and left wheels angular speed, respectively. The dynamic model of the mobile robot can be represented using the Euler-Lagrangian method in the following form [301]:

$$M(q)(\ddot{q}) + C(q, \dot{q})(\dot{q}) + F(\dot{q}) = B(q)u - A(Q)\nabla \tag{6.63}$$

where $M(q)$ is the inertia matrix; $\dot{q}$ represents the velocity vector of both position and orientation; $\ddot{q}$ denotes the acceleration vector of position and direction; $C(q, \dot{q})$ denotes the Centripetal/Coriolis matrix; $F(\dot{q})$ represents a friction vector; $A(q)$ is a constraint matrix; $B(q)$ is the input transformation matrix; $u$ is the control input vector; and $\nabla$ denotes a Lagrange multiplier vector.

### 6.4.4 T2-EFCs Results and Discussion

In the following section, the effectiveness of the proposed control system is investigated under three different scenarios. First, the proposed method is utilized to regulate the dynamics of a differential-drive mobile robot to follow the desired trajectory under nominal conditions. Second, band-limited white noise was injected into the feedback loop using a MATLAB/SIMULINK block as illustrated in Fig. 6.14. Third, external disturbance was added to the system's dynamics as $d_x = 2\ cos(t)$ for the *x-axis* and $d_y = 2\ sin(t)$ for the *y-axis*. In addition, the efficacy of the proposed method is compared with three other controllers, namely, type-1 fuzzy logic control (T1FLC), type-2 fuzzy logic control (T2FLC), and a type-1 evolving fuzzy control system (T1-EFCS). The root mean square error (RMSE) criterion is utilized to evaluate the performance of the proposed control system. The prior threshold values are chosen as $T_{add} = 0.11$ and $T_{del} = 0.07$. The desired/reference trajectory of the mobile robot can be represented as $p_r = [x_r,\ y_r,\ \theta_r]^T$, where we use the velocity reference model to obtain the expected velocity as $q_r(t) =$

$[v_r,\ w_r]$. Therefore, the velocity error can be defined by:

$$e_{desired} = q_r - q = [e_v,\ e_w]^T. \tag{6.64}$$

In the simulation model, the following parameters are set as: $m = 10kg$; $r = 0.05m$; $d = 0.4m$; $F(\dot{q}) = 0$. The desired trajectory is set to follow a sine wave reference for the *x-axis*, and a cosine wave reference for the *y-axis*.

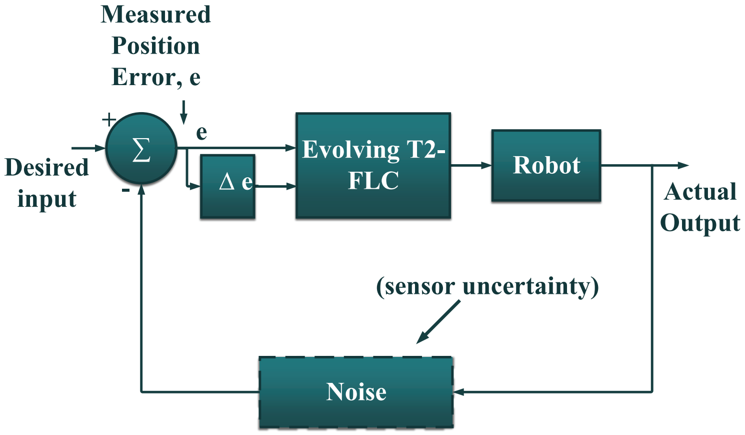


Figure 6.14: Overall closed-loop control system with added noise as sensor uncertainties.

#### 6.4.4.1 Performance Under Nominal Condition

As can be depicted in Figs. 6.15-6.16, the simulation results for different controllers for position control in the $xy$-axes were presented, where the proposed T2-EFCS achieved a favorable tracking performance compared to other benchmark controllers demonstrated by their RMSE values in Table. 6.2. Moreover, the distance error between the desired and the actual trajectory for the four controllers was illustrated in Fig. 6.17, where the error decreased from around $0.19m$ to almost zero value in a very short time using our proposed method. Since the proposed T2-EFCS evolves with time, it is important to report the evolving rules with respect to simulation time. It can be seen in Fig. 6.18 that only two rules were required to track the desired trajectory using T2-EFCS. This clearly indicates the simplicity of the proposed control algorithm, making it suitable for small systems with limited computational payloads.

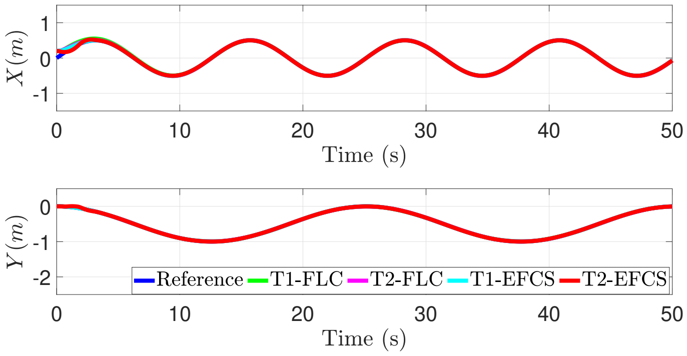


Figure 6.15: Desired vs. actual positions of the $xy$-axes.

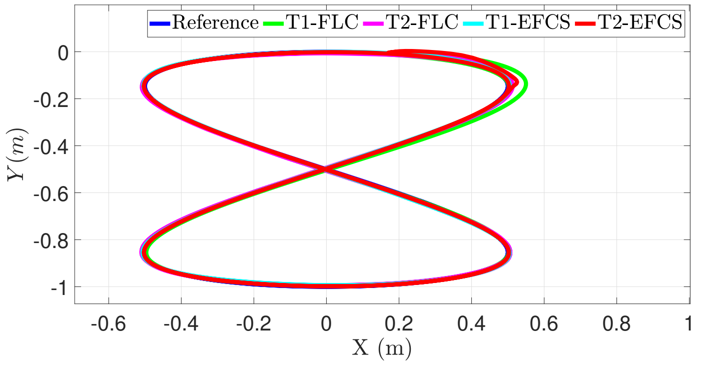


Figure 6.16: Desired vs. actual positions of 8-shape.

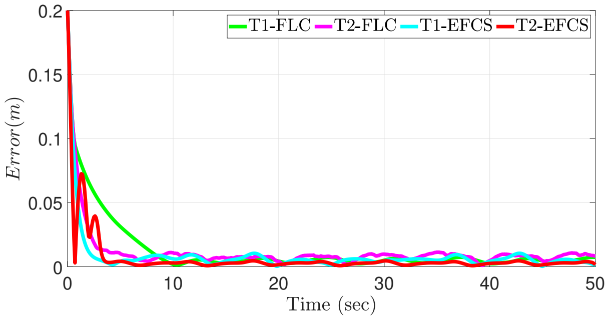


Figure 6.17: Distance error evolution for different control systems.

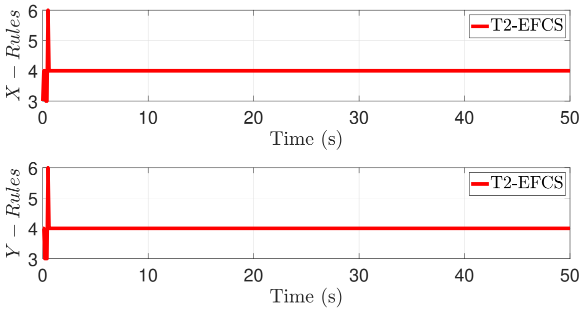


Figure 6.18: Evolution of the fuzzy rules for the proposed T2-EFCS.

Table 6.2: Summary of the experimental comparison of the performance of different controllers in nominal condition.

| **RMSE Values (Nominal Condition)** | | | |
|---|---|---|---|
| **Metrics** | $\boldsymbol{Error_x}[m]$ | $\boldsymbol{Error_y}[m]$ | $\boldsymbol{Error_{dis}}[m]$ |
| **T1FLC** | 0.025 | 0.006 | 0.026 |
| **T2FLC** | 0.021 | 0.005 | 0.022 |
| **ET1FLC** | 0.018 | **0.004** | 0.019 |
| **ET2FLC** | **0.017** | 0.005 | **0.018** |

#### 6.4.4.2 Performance in the face of measurement noise

In this section, the robustness analysis is presented by injecting a band-limited white noise to the feedback loop using MATLAB/SIMULINK block. The position tracking in the $xy$-axes was shown in Fig. 6.19. The simulation results illustrated that the proposed evolving method can handle the disturbance within a reasonable period of time. Also, the distance error between the desired and the actual trajectory for the four controllers was plotted in Fig. 6.20. Lower RMSE values were obtained using the proposed method compared to other benchmark control systems as tabulated in Table. 6.3. Lastly, the rules were pruned according to plant's dynamics to accommodate measurement noise as shown in Fig. 6.21.

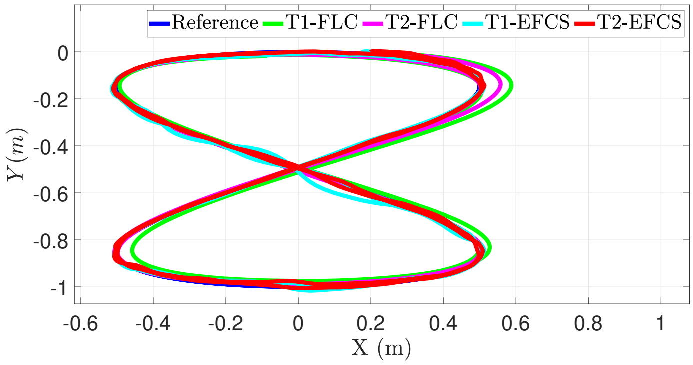


Figure 6.19: Desired vs. actual positions of 8-shape in the face of sensor noise.

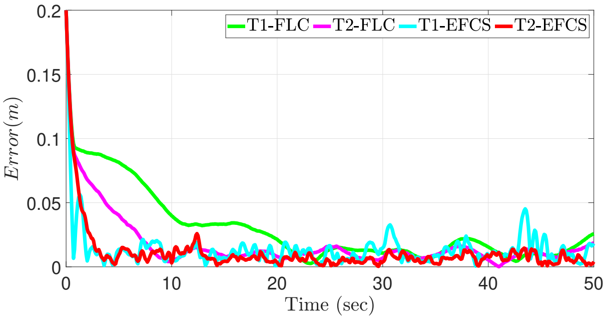


Figure 6.20: Distance error evolution for different control systems in the face of measurement noise.

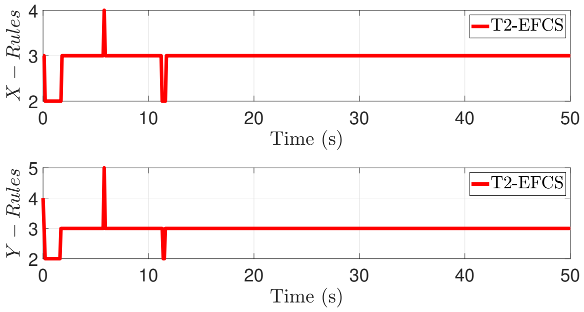


Figure 6.21: Evolution of the fuzzy rules for T2-EFCS in the face of measurement noise.

Table 6.3: Summary of the experimental comparison of the performance of different controllers in the face of measurement noise.

| **RMSE values (Uncertain Condition-Noisy Sensor data)** | | | |
|---|---|---|---|
| **Metrics** | $\boldsymbol{Error_x}[m]$ | $\boldsymbol{Error_y}[m]$ | $\boldsymbol{Error_{dis}}[m]$ |
| **T1FLC** | 0.0391 | 0.012 | 0.041 |
| **T2FLC** | 0.0269 | 0.008 | 0.028 |
| **ET1FLC** | **0.019** | 0.010 | 0.021 |
| **ET2FLC** | 0.020 | **0.006** | **0.021** |

#### 6.4.4.3 Performance Under Unknown Disturbance

Additionally, the controller's performance has been assessed by observing the tracking performance in the face of external disturbance. The position tracking in the $xy$-axes under external disturbance was shown in Fig. 6.22, where the simulation results illustrated the efficacy of the proposed method to handle the external disturbances. Similarly, the RMSE criterion was used for comparing the tracking performance with benchmark controllers as presented in Table. 6.4. The distance error between the desired and the actual trajectory for the four controllers was plotted in Fig. 6.23. Lastly, more rules were evolved to face uncertainties as shown in Fig. 6.24.

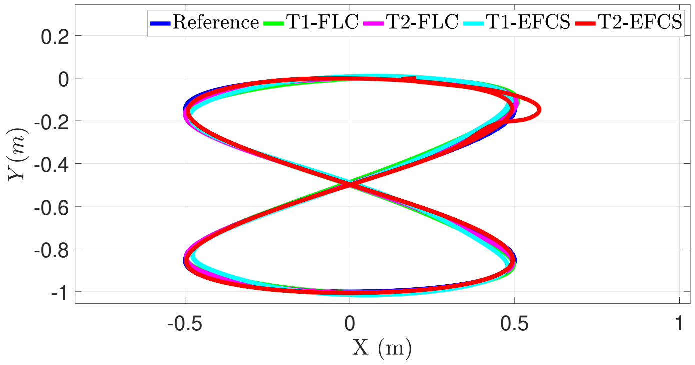


Figure 6.22: Desired vs. actual positions of 8-shape in the face of external disturbance.

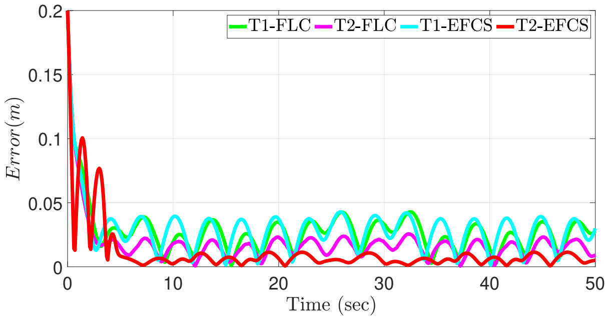


Figure 6.23: Distance error evolution for different control systems in the face of external disturbance.

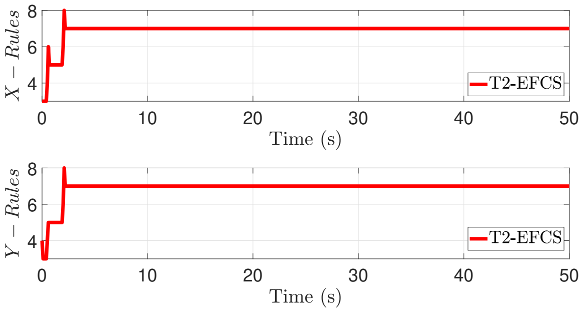


Figure 6.24: Evolution of the fuzzy rules for the proposed T2-EFCS in the face of external disturbance.

Table 6.4: Summary of the experimental comparison of the performance of different controllers in the face of external disturbance.

| **RMSE Values (Uncertain Condition-External disturbance)** | | | |
|---|---|---|---|
| **Metrics** | $\boldsymbol{Error_x}[m]$ | $\boldsymbol{Error_y}[m]$ | $\boldsymbol{Error_{dis}}[m]$ |
| **T1FLC** | 0.0307 | 0.0140 | 0.0337 |
| **T2FLC** | 0.0232 | 0.0112 | 0.0258 |
| **ET1FLC** | 0.0306 | 0.0164 | 0.0347 |
| **ET2FLC** | **0.0224** | **0.0049** | **0.0229** |

## 6.5 Summary

In the first part of this chapter, a simplified model-free type-1 fuzzy evolving controller was designed for uncertain nonlinear dynamic systems. Simulation results illustrated a better trajectory tracking performance using the proposed self-evolving fuzzy control system compared to the adaptive fuzzy controller. The main benefit of the proposed controller is due to its ability to accommodate uncertainties and variations in plant dynamics.

The second part of this chapter proposed a novel type-2 evolving fuzzy control system, named the T2-EFCS, for uncertain dynamic systems. The proposed method was implemented to regulate the position of a mobile robot system. The self-evolving framework can perform self-learning which adds and prunes its fuzzy rules efficiently. To investigate the robustness of T2-EFCS, an external disturbance was added to the nonlinear model, where simulation results illustrated that T2-EFCS can handle uncertainties in systems' dynamics. Moreover, lower RMSE values were reported from the proposed T2-EFCS compared to other benchmark controllers. The newly developed method leverages the advantages of evolving fuzzy systems for controlling uncertain nonlinear systems.

# Chapter 7

# Conclusions

## 7.1 Research and Outcomes

As universal approximators, fuzzy logic systems have proved to be successful computational tools for representing the behavior of complex nonlinear dynamical systems. They also can provide a large avenue for classification, modeling, and controlling various engineering applications. Nevertheless, traditional techniques for deriving fuzzy rules are often tedious and time-consuming. They require expert knowledge and their performance is less than optimum. Addressing the shortcomings, the main motivation of this thesis is to design self-learning type-2 fuzzy systems for system identification and control of dynamic systems. Besides, since the footprint-of-uncertainty is not incorporated in the membership function of type-1 fuzzy sets, such systems have limited ability to handle uncertainties. As such, the development of type-2 fuzzy systems in this thesis has been presented to mitigate the limited capability of type-1 fuzzy systems to accommodate the footprint-of-uncertainty, in particular for autonomous systems, where uncertainty is unavoidable.

In chapter 3, a novel online system identification approach, named the IT2-TS-FC technique, was proposed, by employing a recursive interval type-2 fuzzy C-means clustering technique. The algorithm of IT2-TS-FC utilized the Lagrange method to minimize the objective function. As with its predecessors, IT2-TS-FC was equipped with two fuzzifers to represent the upper and lower membership functions while employing the WLS method to determine the fuzzy consequent parameters. The algorithm was intended to model a Mackey-Glass time series as well as describe the dynamics of a quadcopter. The online prediction accuracy of the IT2-TS-FC identifier was better than the accuracy of the other benchmark techniques. The proposed method leverages the benefits of IT2-FLSs for modeling dynamic systems, especially in the presence of a noisy dataset, thanks to the *FOU* in IT2FLSs.

In Chapter 4, a new self-adaptive interval type-2 fuzzy controller, named SAF2C, was presented to control MIMO dynamic systems. SAF2C was based on the Takagi-Sugeno fuzzy model and it accommodated the 'Enhanced Iterative Algorithm with Stop Condition' type-reducer, which is more computationally efficient than the 'Karnik-Mendel' type-reduction algorithm. The proposed method was designed to expedite the learning process of the control algorithm by 80% compared to separate SISO controllers. The proposed control system was deployed to control various nonlinear systems, including a hexacopter UAV, where the outcomes of this chapter suggest a 20% improvement in transient response, in addition to achieving a better noise rejection capability with respect to its T1-FLC counterpart. Moreover, SAF2C demonstrated better robustness against uncertainties (e.g., external disturbance and wind/gust), where superior disturbance rejection was shown through extensive computer simulations compared to the benchmark type-1 fuzzy and PID controller. Besides, the proposed closed-loop control system was applied to control other benchmark dynamic systems (e.g., a simulated autonomous underwater vehicle and inverted pendulum on a cart system) demonstrating high accuracy and robustness to variations in system parameters and external disturbance.

Another novel stand-alone enhanced self-adaptive interval type-2 fuzzy controller named the ESAF2C algorithm was proposed in chapter 5. ESAF2C was developed as an enhanced version of SAF2C and for mitigating the challenges for real-time deployment. The parameters of the ESAF2C algorithm were tuned in an online fashion using the SMC theory making the system robust to variations in system parameters and external disturbances. The proposed technique was applied on a quadcopter UAV, where extensive simulations and real-time flight tests for a hovering QUAV under wind disturbances were also conducted. Specifically, the controller's performance has been assessed by studying the tracking performance in the face of external wind/gust disturbances, generated using an industrial fan. The promising outcomes of the proposed method open the door for utilizing the ESAF2C in various real-world systems.

Additionally, the development of a type-2 self-evolving fuzzy control system, named the T2-EFCS, was presented in chapter 6 to facilitate self-learning (either from scratch or from a certain predefined rule). T2-EFCS was proposed to mitigate the limitation of the static structure of fuzzy systems. T2-EFCS has two phases, namely, structure learning and parameters learning. The structure of T2-EFCS did not require previous information about the fuzzy structure, and it can start the construction of its rules from scratch with only one rule. The rules were then added and pruned in an online fashion to achieve the desired set-point. The self-evolving closed-loop control mechanism has been implemented to control an unmanned ground vehicle in the presence of multiple external disturbances demonstrating promising tracking performance. The T2-EFCS parameters have been tuned using the SMC theory making the system robust to variations in system parameters and external disturbance. The integration of self-evolving approaches with the SMC-based adaptive law resulted in fast learning and compact structure. The proposed T2-EFCS was computationally efficient while maintaining superior control performance.

Prior to the development of the T2-EFCS, a simplified type-1 evolving fuzzy control system, named the T1-EFCS, was introduced in the first part of chapter 6. The structure of T1-EFCS had no dependency on previous information about the fuzzy structure and it started its construction from scratch with only one rule. The rules were then added and deleted in an online fashion. Achievements have been compared with an adaptive T1-FLC.

In the end of this thesis, stability analysis of the proposed control frameworks in this thesis was conducted using the Lyapunov theory to ensure the convergence of fuzzy parameters. Moreover, throughout this thesis, the performance of the proposed control systems was tested in the face of various uncertainties (e.g. disturbance, sensor noise, and wind/gust).

## 7.2 Recommendations for Future Research

Although this thesis has identified numerous limitations of the existing type-2 fuzzy systems for system identification and control of dynamic systems and presented new approaches to mitigate these limitations, there are still several potential research problems that can be pursued to enhance the developed methodology. The following are some potential research areas in adaptive and evolving type-2 fuzzy systems:

- The learning rates $\mathit{\Gamma}$ and $\lambda$ of the SAF2C and ESAF2C control systems were selected based on the input frequency and the system's rate-of-change, which decide the learning speed of the type-2 fuzzy system. As such, choosing high learning rates values will result in an unstable response. On the other hand, choosing small learning rates leads to slow convergence of type-2 fuzzy parameters. Although it is recommended to commence with low values and increase gradually until the desired performance is observed, for future work, an automatic learning algorithm can be investigated to select the learning rate parameters in an online fashion.
- The proposed T2-EFCS requires the initialization of the predefined threshold parameters when adding and pruning fuzzy rules. This can be improved in future studies to find these optimal values without the need for user-defined parameters. In addition, the stability of the proposed T2-EFCS has been proven using the Lyapunov stability theory and assessed by numerical simulations to control an autonomous ground vehicle, for future studies, this promising method can be implemented for MIMO ground robots in real-time to reduce the execution time.
- Evolving type-2 fuzzy systems can be implemented in fault-tolerant control problems. For instance, actuator failures are one of the most common problems that may occur in dynamic systems, resulting in poor system behaviour or even catastrophic accidents. To handle these failures, the proposed T2-EFCS approach can be

implemented to reconfigure the plant such that safe and secure operations can be achieved in the face of faults.

- The efficacy of the proposed IT2-TS-FC can be further compared with other universal approximator system identification techniques such as NNs-based approaches.
- Lastly, reinforcement learning is becoming one of the most prominent AI-based approaches as it resembles the way humans learn. For example, in literature, the fuzzy Q-learning (FQL) method has been implemented in various applications, however, the fuzzy structure is still determined by a *priori* knowledge. To tackle this problem, the integration of reinforcement learning and self-evolving fuzzy systems can be deployed in various possible control applications such as in an unknown environment for mobile robots, allowing an elastic structure of FQL approaches.

## 7.3 Concluding Remarks

Benefiting from the footprint-of-uncertainties provided in T2-FSs, self-learning type-2 fuzzy systems have many advantages over type-1 fuzzy systems to handle uncertainties in system identification and control problems. This is demonstrated by their promising tracking performance, robustness against external disturbance and noise rejection. Besides, T2EFSs has an additional feature, where their structure evolves autonomously making them able to cope with sudden changes to systems' dynamics. There are still issues to be tackled when selecting fuzzy learning parameters but there is doubtlessly much hope that this can be resolved.

# References

[1] G. Ferri, A. Munafo, and K. D. LePage, "An autonomous underwater vehicle data-driven control strategy for target tracking," *IEEE Journal of Oceanic Engineering*, vol. 43, no. 2, pp. 323–343, 2018.

[2] Q. Zhang and M. Mahfouf, "Fuzzy modelling using a new compact fuzzy system: A special application to the prediction of the mechanical properties of alloy steels," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*. IEEE, 2011, pp. 1041–1048.

[3] M. M. Ferdaus, S. G. Anavatti, M. A. Garratt, and M. Pratama, "Fuzzy clustering based nonlinear system identification and controller development of pixhawk based quadcopter," in *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*. IEEE, 2017, pp. 223–230.

[4] F. Santoso, M. Garratt, and S. Anavatti, "Fuzzy system identification for the dynamics of the AR. Drone quadcopter," in *Australasian Conference on Robotics and Automation*, vol. 16, 2016, pp. 69–74.

[5] J. d. J. Rubio, "SOFMLS: Online self-organizing fuzzy modified least-squares network," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 6, pp. 1296–1309, Dec 2009.

[6] M. M. Ferdaus, S. G. Anavatti, M. A. Garratt, and M. Pratama, "Fuzzy clustering based modelling and adaptive controlling of a flapping wing micro air vehicle," in

*2017 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE, Nov 2017, pp. 1–6.

[7] A. J. Al-Mahasneh, S. Anavatti, and M. Garratt, "Nonlinear multi-input multi-output system identification using neuro-evolutionary methods for a quadcopter," in *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI).* IEEE, 2017, pp. 217–222.

[8] A. J. Al-Mahasneh, S. G. Anavatti, and M. A. Garratt, "Altitude identification and intelligent control of a flapping wing micro aerial vehicle using modified generalized regression neural networks," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE, 2017, pp. 1–6.

[9] C. N. Giap, L. H. Son, and F. Chiclana, "Dynamic structural neural network," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 4, pp. 2479–2490, 2018.

[10] P. Angelov, "A fuzzy controller with evolving structure," *Information Sciences*, vol. 161, no. 1-2, pp. 21–35, 2004.

[11] A. Al-Mahturi, F. Santoso, M. A. Garratt, S. G. Anavatti, and M. M. Ferdaus, "Online Takagi-Sugeno fuzzy identification of a quadcopter using experimental input-output data," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2019, pp. 527–533.

[12] R. L. Galveza, E. P. Dadiosa, and A. Bandala, "Fuzzy logic based path planning for quadrotor," in *Proceedings of the 2015 Annual PAASE Meeting and Symposium De La Salle University*, 2015.

[13] H.-J. Rong, Z.-X. Yang, P. K. Wong, C. M. Vong, and G.-S. Zhao, "Self-evolving fuzzy model-based controller with online structure and parameter learning for hypersonic vehicle," *Aerospace Science and Technology*, vol. 64, pp. 1–15, 2017.

[14] F. Santoso, M. A. Garratt, and S. G. Anavatti, "State-of-the-art intelligent flight control systems in unmanned aerial vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. PP, no. 99, pp. 1–15, 2017.

[15] J. M. Mendel, R. I. John, and F. Liu, “Interval type-2 fuzzy logic systems made simple,” *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 6, pp. 808–821, 2006.

[16] N. N. Karnik, J. M. Mendel, and Q. Liang, “Type-2 fuzzy logic systems,” *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.

[17] A. D. Torshizi, M. H. F. Zarandi, and H. Zakeri, “On type-reduction of type-2 fuzzy sets: A review,” *Applied Soft Computing*, vol. 27, pp. 614–627, 2015.

[18] 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.

[19] D. Wu and J. M. Mendel, “Recommendations on designing practical interval type-2 fuzzy systems,” *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 182–193, 2019.

[20] J. Mendel, H. Hagras, W.-W. Tan, W. W. Melek, and H. Ying, *Introduction to type-2 fuzzy logic control: theory and applications.* John Wiley & Sons, 2014.

[21] E. Kayacan, A. Sarabakha, S. Coupland, R. John, and M. A. Khanesar, “Type-2 fuzzy elliptic membership functions for modeling uncertainty,” *Engineering Applications of Artificial Intelligence*, vol. 70, pp. 170–183, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197618300253

[22] J. M. Mendel, *Uncertain rule-based fuzzy systems.* Springer, 2017.

[23] O. Castillo and P. Melin, “A review on interval type-2 fuzzy logic applications in intelligent control,” *Information Sciences*, vol. 279, pp. 615–631, Sep. 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0020025514004629

[24] R. Shahnazi, “Observer-based adaptive interval Type-2 fuzzy control of uncertain MIMO nonlinear systems with unknown asymmetric saturation actuators,” *Neurocomputing*, vol. 171, pp. 1053–1065, 2016.

[25] H. Hagras, “Type-2 FLCs: A new generation of fuzzy controllers,” *IEEE Computational Intelligence Magazine*, vol. 2, no. 1, pp. 30–43, 2007.

[26] S. Aich, C. Ahuja, T. Gupta, and P. Arulmozhivarman, "Analysis of ground effect on multi-rotors," in *Electronics, Communication and Computational Engineering (ICECCE), 2014 International Conference on.* IEEE, 2014, pp. 236–241.

[27] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 535–550, Oct. 2000.

[28] T.-T. Huynh, C.-M. Lin, T.-L. Le, H.-Y. Cho, T.-T. T. Pham, F. Chao *et al.*, "A new self-organizing fuzzy cerebellar model articulation controller for uncertain nonlinear systems using overlapped gaussian membership functions," *IEEE Transactions on Industrial Electronics*, 2019.

[29] E. Kayacan and R. Maslim, "Type-2 fuzzy logic trajectory tracking control of quadrotor VTOL aircraft with elliptic membership functions," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 339–348, 2017.

[30] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "An intelligent control of an inverted pendulum based on an adaptive interval type-2 fuzzy inference system," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–6.

[31] O. Castillo and P. Melin, "A review on the design and optimization of interval type-2 fuzzy controllers," *Applied Soft Computing*, vol. 12, no. 4, pp. 1267–1278, Apr. 2012. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1568494611004947

[32] A. Sarabakha and E. Kayacan, "Online deep fuzzy learning for control of nonlinear systems using expert knowledge," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 7, pp. 1492–1503, 2020.

[33] E. Kayacan, M. A. Khanesar, J. Rubio-Hervas, and M. Reyhanoglu, "Learning control of fixed-wing unmanned aerial vehicles using fuzzy neural networks," *International Journal of Aerospace Engineering*, vol. 2017, 2017.

[34] M. Biglarbegian, W. W. Melek, and J. M. Mendel, “On the stability of interval type-2 TSK fuzzy logic control systems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 798–818, 2010.

[35] Y. Shi, R. Eberhart, and Y. Chen, “Implementation of evolutionary fuzzy systems,” *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 2, pp. 109–119, 1999.

[36] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, “A robust adaptive interval type-2 fuzzy control for autonomous underwater vehicles,” in *2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, 2019, pp. 19–24.

[37] F. Santoso, M. A. Garratt, and S. G. Anavatti, “T2-ETS-IE: Type-2 evolutionary Takagi-Sugeno fuzzy inference systems with information entropy-based pruning technique,” *IEEE Transactions on Fuzzy Systems*, 2019.

[38] 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, 2017.

[39] E. Hüllermeier and K. Brinker, “Learning valued preference structures for solving classification problems,” *Fuzzy Sets and Systems*, vol. 159, no. 18, pp. 2337–2352, 2008.

[40] D. Leite, R. M. Palhares, V. C. 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, 2014.

[41] M. M. Ferdaus, M. Pratama, S. Anavatti, M. A. Garratt, and Y. Pan, “Generic evolving self-organizing neuro-fuzzy control of bio-inspired unmanned aerial vehicles,” *IEEE Transactions on Fuzzy Systems*, 2019.

[42] H.-G. Han, Z.-Y. Chen, H.-X. Liu, and J.-F. Qiao, “A self-organizing interval type-2 fuzzy-neural-network for modeling nonlinear systems,” *Neurocomputing*, vol. 290, pp. 196–207, 2018.

[43] Y. Y. Lin, J. Y. Chang, and C. T. 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, Jan. 2014.

[44] C.-M. Lin, Y.-M. Chen, and C.-S. Hsueh, "A Self-Organizing Interval Type-2 Fuzzy Neural Network for Radar Emitter Identification," *International Journal of Fuzzy Systems*, vol. 16, no. 1, p. 11, 2014.

[45] 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, Dec. 2014. [Online]. Available: https://link.springer.com/article/10.1007/s12530-013-9102-9

[46] A. Bouchachia and C. Vanaret, "GT2FC: An Online Growing Interval Type-2 Self-Learning Fuzzy Classifier," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 999–1018, Aug. 2014.

[47] M. Pratama, S. G. Anavatti, P. P. Angelov, and E. Lughofer, "PANFIS: A Novel Incremental Learning Machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 55–68, Jan. 2014.

[48] Y. Y. Lin, J. Y. Chang, N. R. Pal, and C. T. 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, Jun. 2013.

[49] A. J. Al-Mahasneh, S. Anavatti, and M. Garratt, "Self-evolving neural control for a class of nonlinear discrete-time dynamic systems with unknown dynamics and unknown disturbances," *IEEE Transactions on Industrial Informatics*, 2019.

[50] E. Lughofer, *Evolving fuzzy systems-methodologies, advanced concepts and applications.* Springer, 2011, vol. 53.

[51] J. Tavoosi, A. A. Suratgar, and M. B. Menhaj, "Nonlinear system identification based on a self-organizing type-2 fuzzy RBFN," *Engineering Applications of*

*Artificial Intelligence*, vol. 54, pp. 26–38, Sep. 2016. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0952197616300835

[52] C. Li, W. Zou, N. Zhang, and X. Lai, “An evolving T–S fuzzy model identification approach based on a special membership function and its application on pump-turbine governing system,” *Engineering Applications of Artificial Intelligence*, vol. 69, pp. 93 – 103, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197617303007

[53] H. Han, X. Wu, and J. Qiao, “Nonlinear systems modeling based on self-organizing fuzzy-neural-network with adaptive computation algorithm,” *IEEE Transactions on Cybernetics*, vol. 44, no. 4, pp. 554–564, 2014.

[54] H. Han and J. Qiao, “A self-organizing fuzzy neural network based on a growing-and-pruning algorithm,” *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 6, pp. 1129–1143, 2010.

[55] 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.

[56] L. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S001999586590241X

[57] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning—II,” *Information sciences*, vol. 8, no. 4, pp. 301–357, 1975.

[58] L. Zadeh, “Outline of a new approach to the analysis of complex systems and decision processes,” *IEEE Transactions on Systems, Man, and Cybernetics*, no. 1, pp. 28–44, 1973.

[59] M. F. Hamza, H. J. Yap, I. A. Choudhury, H. Chiroma, and T. Kumbasar, “A survey on advancement of hybrid Type-2 fuzzy sliding mode control,” *Neural Computing and Applications*, vol. 30, no. 2, pp. 331–353, 2018.

[60] M. M. Ferdaus, S. G. Anavatti, M. Pratama, and M. A. Garratt, "Towards the use of fuzzy logic systems in rotary wing unmanned aerial vehicle: a review," *Artificial Intelligence Review*, pp. 1–34, 2018.

[61] A. Beke and T. Kumbasar, "Learning with type-2 fuzzy activation functions to improve the performance of deep neural networks," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 372–384, 2019.

[62] M. H. Khooban, D. N. M. Abadi, A. Alfi, and M. Siahi, "Optimal type-2 fuzzy controller for hvac systems," *Automatika*, vol. 55, no. 1, pp. 69–78, 2014.

[63] O. Castillo, P. Melin, J. Kacprzyk, and W. Pedrycz, "Type-2 fuzzy logic: theory and applications," in *IEEE International Conference on Granular Computing, 2007,*. IEEE, 2007, pp. 145–145.

[64] R. Antão, *Type-2 Fuzzy Logic: Uncertain Systems' Modeling and Control.* Springer, 2017.

[65] J. M. Mendel, "Uncertainty, fuzzy logic, and signal processing," *Signal Processing*, vol. 80, no. 6, pp. 913–933, 2000.

[66] D. Sánchez and P. Melin, "Modular neural network with fuzzy integration and its optimization using genetic algorithms for human recognition based on iris, ear and voice biometrics," in *Soft computing for recognition based on biometrics.* Springer, 2010, pp. 85–102.

[67] D. Wu and J. M. Mendel, "Designing practical interval type-2 fuzzy logic systems made simple," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* IEEE, 2014, pp. 800–807.

[68] M. E. Yuksel and A. Basturk, "Application of type-2 fuzzy logic filtering to reduce noise in color images," *IEEE Computational intelligence magazine*, vol. 7, no. 3, pp. 25–35, 2012.

[69] D. Sun, Q. Liao, and H. Ren, “Type-2 fuzzy modeling and control for bilateral teleoperation system with dynamic uncertainties and time-varying delays,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 447–459, 2017.

[70] C.-H. Hsu and C.-F. Juang, “Evolutionary robot wall-following control using type-2 fuzzy controller with species-de-activated continuous aco,” *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 100–112, 2013.

[71] H. Li, Y. Pan, and Q. Zhou, “Filter design for interval type-2 fuzzy systems with D stability constraints under a unified frame,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 719–725, 2015.

[72] R. John and S. Coupland, “Type-2 fuzzy logic: challenges and misconceptions [discussion forum],” *IEEE Computational Intelligence Magazine*, vol. 7, no. 3, pp. 48–52, 2012.

[73] J. M. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new directions.* Prentice Hall PTR Upper Saddle River, 2001.

[74] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, “Nonlinear altitude control of a quadcopter drone using interval Type-2 fuzzy logic,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE, 2018, pp. 236–241.

[75] J. M. Mendel, “Advances in type-2 fuzzy sets and systems,” *Information sciences*, vol. 177, no. 1, pp. 84–110, 2007.

[76] S. Sivanandam, S. Sumathi, S. Deepa *et al.*, *Introduction to fuzzy logic using MATLAB.* Springer, 2007, vol. 1.

[77] K. Tai, A.-R. El-Sayed, M. Biglarbegian, C. I. Gonzalez, O. Castillo, and S. Mahmud, “Review of Recent Type-2 Fuzzy Controller Applications,” *Algorithms*, vol. 9, no. 2, p. 39, Jun. 2016. [Online]. Available: http://www.mdpi.com/1999-4893/9/2/39

[78] N. N. Karnik and J. M. Mendel, “Centroid of a type-2 fuzzy set,” *Information Sciences*, vol. 132, no. 1-4, pp. 195–220, 2001.

[79] D. Wu and J. M. Mendel, "Enhanced Karnik-Mendel algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923–934, 2008.

[80] D. Wu and M. Nie, "Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, June 2011, pp. 2131–2138.

[81] M. Nie and W. W. Tan, "Towards an efficient type-reduction method for interval type-2 fuzzy logic systems," in *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 1425–1432.

[82] D. Wu, "Twelve considerations in choosing between gaussian and trapezoidal membership functions in interval type-2 fuzzy logic controllers," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* IEEE, 2012, pp. 1–8.

[83] D. Wu and M. Nie, "Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems." in *FUZZ-IEEE*, 2011, pp. 2131–2138.

[84] C. Chen, D. Wu, J. M. Garibaldi, R. John, J. Twycross, and J. M. Mendel, "A comment on a direct approach for determining the switch points in the karnik-mendel algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3905–3907, 2018.

[85] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Adaptive neuro-fuzzy inference system identification for the dynamics of the AR. Drone quadcopter," in *International Conference on Sustainable Energy Engineering and Application (ICSEEA).* IEEE, 2016, pp. 55–60.

[86] R. Isermann and M. Münchhof, *Identification of dynamic systems: an introduction with applications.* Springer Science & Business Media, 2010.

[87] E. Capello, H. Park, B. Tavora, G. Guglieri, and M. Romano, "Modeling and experimental parameter identification of a multicopter via a compound pendulum test

rig," in *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE, 2015, pp. 308–317.

[88] A. Kumar, S. Saderla, and A. K. Ghosh, "Aerodynamic parameter estimation using neuro-fuzzy model based method," in *2017 First International Conference on Recent Advances in Aerospace Engineering (ICRAAE)*. IEEE, 2017, pp. 1–5.

[89] S. Samanta, S. Suresh, J. Senthilnath, and N. Sundararajan, "A new neuro-fuzzy inference system with dynamic neurons (NFIS-DN) for system identification and time series forecasting," *Applied Soft Computing*, vol. 82, p. 105567, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494619303473

[90] W. Zou, C. Li, and N. Zhang, "A T-S fuzzy model identification approach based on a modified inter type-2 FRCM algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 3, pp. 1104–1113, 2017.

[91] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "Online system identification for nonlinear autonomous systems using recursive interval type-2 TS fuzzy C-means clustering," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2020.

[92] T. J. Ross *et al.*, *Fuzzy logic with engineering applications*. Wiley Online Library, 2004, vol. 2.

[93] C. Li, J. Zhou, J. Xiao, and H. Xiao, "Hydraulic turbine governing system identification using T-S fuzzy model optimized by chaotic gravitational search algorithm," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 9, pp. 2073–2082, 2013.

[94] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy C-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[95] J. Abonyi, R. Babuska, and F. Szeifert, "Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 5, pp. 612–621, 2002.

[96] C. Li, J. Zhou, X. Xiang, Q. Li, and X. An, "T-S fuzzy model identification based on a novel fuzzy C-regression model clustering algorithm," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 4-5, pp. 646–653, 2009.

[97] M. F. Zarandi, I. Türkşen, and O. T. Kasbi, "Type-2 fuzzy modeling for desulphurization of steel process," *Expert Systems with Applications*, vol. 32, no. 1, pp. 157–171, 2007.

[98] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "A robust self-adaptive interval type-2 TS fuzzy logic for controlling multi-input multi-output nonlinear uncertain dynamical systems." *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[99] C. Hwang and F. C.-H. Rhee, "Uncertain fuzzy clustering: Interval type-2 fuzzy approach to *c*-means," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 107–120, 2007.

[100] R. A. Aliev, W. Pedrycz, B. G. Guirimov, R. R. Aliev, U. Ilhan, M. Babagil, and S. Mammadli, "Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization," *Information Sciences*, vol. 181, no. 9, pp. 1591–1608, May 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0020025510006201

[101] T. Dam and A. K. Deb, "Interval type-2 modified fuzzy c-regression model clustering algorithm in TS fuzzy model identification," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* IEEE, 2016, pp. 1671–1676.

[102] K. M. Passino, S. Yurkovich, and M. Reinfrank, *Fuzzy control.* Citeseer, 1998, vol. 20.

[103] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "A novel learning-from-scratch evolving type-2 fuzzy system for robotic control," *Preparing to be submitted to IEEE Transactions on Intelligent Transportation Systems*, 2021.

[104] W. S. Levine, *Control system fundamentals.* CRC press, 1999.

[105] S. P. Bhattacharyya, A. Datta, and L. H. Keel, *Linear control theory: structure, robustness, and optimization.* CRC press, 2018.

[106] A. Kumar and V. Kumar, “Evolving an interval type-2 fuzzy PID controller for the redundant robotic manipulator,” *Expert Systems with Applications*, vol. 73, pp. 161–177, 2017.

[107] S. Zeghlache, K. Kara, and D. Saigaa, “Fault tolerant control based on interval type-2 fuzzy sliding mode controller for coaxial trirotor aircraft,” *ISA Transactions*, vol. 59, pp. 215–231, Nov. 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0019057815002207

[108] A. J. Al-Mahasneh, S. G. Anavatti, and M. A. Garratt, “Review of applications of generalized regression neural networks in identification and control of dynamic systems,” *arXiv preprint arXiv:1805.11236*, 2018.

[109] Y.-Y. Lin, S.-H. Liao, J.-Y. Chang, C.-T. Lin *et al.*, “Simplified interval type-2 fuzzy neural networks.” *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 5, pp. 959–969, 2014.

[110] J. A. Suykens, J. P. Vandewalle, and B. L. de Moor, *Artificial neural networks for modelling and control of nonlinear systems.* Springer Science & Business Media, 1995.

[111] N. Siddique, *Intelligent control: a hybrid approach based on fuzzy logic, neural networks and genetic algorithms.* Springer, 2013, vol. 517.

[112] C. Grosan and A. Abraham, *Intelligent systems.* Springer, 2011.

[113] R. A. Aliev and B. G. Guirimov, *Type-2 fuzzy neural networks and their applications.* Springer, 2014.

[114] C. J. Kim and D. Chwa, “Obstacle Avoidance Method for Wheeled Mobile Robots Using Interval Type-2 Fuzzy Neural Network,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 677–687, Jun. 2015.

[115] H. Kandath, M. A. Hady, M. Pratama, and N. B. Feng, "Robust evolving neuro-fuzzy control of a novel tilt-rotor vertical takeoff and landing aircraft," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2019, pp. 1–6.

[116] A. L. Salih, M. Moghavvemi, H. A. Mohamed, and K. S. Gaeid, "Flight PID controller design for a UAV quadrotor," *Scientific Research and Essays*, vol. 5, no. 23, pp. 3660–3667, 2010.

[117] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2004, pp. 2451–2456.

[118] K. Alexis, G. Nikolakopoulos, and A. Tzes, "Model predictive quadrotor control: attitude, altitude and position experimental studies," *IET Control Theory & Applications*, vol. 6, no. 12, pp. 1812–1827, 2012.

[119] A. Sarabakha, C. Fu, E. Kayacan, and T. Kumbasar, "Type-2 fuzzy logic controllers made even simpler: From design to deployment for UAVs," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 5069–5077, 2018.

[120] L. Xu, H. Ma, D. Guo, a. xie, and d. song, "Backstepping sliding-mode and cascade active disturbance rejection control for a quadrotor UAV," *IEEE/ASME Transactions on Mechatronics*, pp. 1–1, 2020.

[121] A. J. Al-Mahasneh, S. G. Anavatti, M. A. Garratt, and M. Pratama, "Stable adaptive controller based on generalized regression neural networks and sliding mode control for a class of nonlinear time-varying systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2019.

[122] M. M. Ferdaus, M. Pratama, S. G. Anavatti, M. A. Garratt, and Y. Pan, "Generic evolving self-organizing neuro-fuzzy control of bio-inspired unmanned aerial vehicles," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 8, pp. 1542–1556, 2020.

[123] M.-K. Chang, "An adaptive self-organizing fuzzy sliding mode controller for a 2-dof rehabilitation robot actuated by pneumatic muscle actuators," *Control*

*Engineering Practice*, vol. 18, no. 1, pp. 13–22, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066109001518

[124] M.-T. Yan, "An adaptive control system with self-organizing fuzzy sliding mode control strategy for micro wire-EDM machines," *The International Journal of Advanced Manufacturing Technology*, vol. 50, no. 1-4, pp. 315–328, 2010.

[125] Y. Zhao, J. Wang, F. Yan, and Y. Shen, "Adaptive sliding mode fault-tolerant control for type-2 fuzzy systems with distributed delays," *Information Sciences*, vol. 473, pp. 227 – 238, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025518306911

[126] Q. Lu, P. Shi, H. Lam, and Y. Zhao, "Interval type-2 fuzzy model predictive control of nonlinear networked control systems," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2317–2328, 2015.

[127] H. Hagras, F. Doctor, V. Callaghan, and A. Lopez, "An Incremental Adaptive Life Long Learning Approach for Type-2 Fuzzy Embedded Agents in Ambient Intelligent Environments," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 1, pp. 41–55, Feb. 2007.

[128] L. Shanmugam and Y. H. Joo, "Design of interval Type-2 fuzzy-based sampled-data controller for nonlinear systems using novel fuzzy lyapunov functional and its application to PMSM," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, 2018.

[129] F. Chao, D. Zhou, C.-M. Lin, L. Yang, C. Zhou, and C. Shang, "Type-2 fuzzy hybrid controller network for robotic systems," *IEEE Transactions on Cybernetics*, 2019.

[130] X. Lu, Y. Zhao, and M. Liu, "Self-learning interval type-2 fuzzy neural network controllers for trajectory control of a Delta parallel robot," *Neurocomputing*, vol. 283, pp. 107–119, Mar. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217319057

[131] H. Li, J. Wang, H. Lam, Q. Zhou, and H. Du, "Adaptive sliding mode control for interval Type-2 fuzzy systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 12, pp. 1654–1663, Dec 2016.

[132] A. Sarabakha, C. Fu, E. Kayacan, and T. Kumbasar, "Type-2 fuzzy logic controllers made even simpler: From design to deployment for UAVs," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 6, pp. 5069–5077, 2018.

[133] P. K. Ray, S. R. Paital, A. Mohanty, Y. E. Foo, A. Krishnan, H. B. Gooi, and G. A. Amaratunga, "A hybrid firefly-swarm optimized fractional order interval type-2 fuzzy PID-PSS for transient stability improvement," *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 6486–6498, 2019.

[134] S. Zeghlache, D. Saigaa, and K. Kara, "Fault tolerant control based on neural network interval Type-2 fuzzy sliding mode controller for octorotor UAV," *Frontiers of Computer Science*, vol. 10, no. 4, pp. 657–672, Aug. 2016. [Online]. Available: http://link.springer.com/10.1007/s11704-015-4448-8

[135] M. Ghaemi and M.-R. Akbarzadeh-Totonchi, "Indirect adaptive interval Type-2 fuzzy PI sliding mode control for a class of uncertain nonlinear systems," *Iranian Journal of Fuzzy Systems*, vol. 11, no. 5, pp. 1–21, 2014.

[136] M.-Y. Hsiao, T.-H. S. Li, J.-Z. Lee, C.-H. Chao, and S.-H. Tsai, "Design of interval Type-2 fuzzy sliding-mode controller," *Information Sciences*, vol. 178, no. 6, pp. 1696–1716, 2008.

[137] M. E. Abdelaal, H. M. Emara, and A. Bahgat, "Interval Type-2 fuzzy sliding mode control with application to inverted pendulum on a cart," in *2013 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2013, pp. 100–105.

[138] F. Baghbani, M.-R. Akbarzadeh-T., A. Akbarzadeh, and M. Ghaemi, "Robust adaptive mixed $\mathrm{H}_2/\mathrm{H}\infty$ interval Type-2 fuzzy control of nonlinear uncertain systems with minimal control effort," *Engineering Applications of Artificial Intelligence*, vol. 49, pp. 88 – 102, 2016.

[139] Q. Lu, P. Shi, H.-K. Lam, and Y. Zhao, "Interval Type-2 fuzzy model predictive control of nonlinear networked control systems," *IEEE Transactions on Fuzzy systems*, vol. 23, no. 6, pp. 2317–2328, 2015.

[140] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Robust hybrid feedback linearization and interval type-2 fuzzy control systems for the flapping angle dynamics of a biomimetic aircraft," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, 2019.

[141] P. P. Angelov, *Evolving rule-based models: a tool for design of flexible adaptive systems.* Physica, 2013, vol. 92.

[142] 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.

[143] 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.

[144] 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.*, 2005, pp. 1068–1073.

[145] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165011405006020

[146] P. Angelov, "Evolving takagi-sugeno fuzzy systems from streaming data (eTS+)," in *Evolving intelligent systems: methodology and applications.* Wiley Online Library, 2010, vol. 12, p. 21.

[147] 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, Jan. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016401667

[148] C. F. Juang and Y. W. Tsao, "A Type-2 Self-Organizing Neural Fuzzy System and Its FPGA Implementation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 6, pp. 1537–1548, Dec. 2008.

[149] O. Hassanein, S. G. Anavatti, H. Shim, and S. A. Salman, "Auto-generating fuzzy system modelling of physical systems," in *2015 IEEE Conference on Control Applications (CCA)*. IEEE, 2015, pp. 1142–1147.

[150] C.-H. Chen, C.-J. Lin, and C.-T. Lin, "A functional-link-based neurofuzzy network for nonlinear system control," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 5, pp. 1362–1378, 2008.

[151] 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.

[152] M. Pratama, S. G. Anavatti, and E. Lughofer, "GENEFIS: toward an effective localist network," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 3, pp. 547–562, Jun. 2014.

[153] C. Li, W. Zou, N. Zhang, and X. Lai, "An evolving T-S fuzzy model identification approach based on a special membership function and its application on pump-turbine governing system," *Engineering Applications of Artificial Intelligence*, vol. 69, pp. 93–103, 2018.

[154] 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)*, Jul. 2016, pp. 1841–1848.

[155] N. K. Kasabov, *Evolving connectionist systems: the knowledge engineering approach.* Springer Science & Business Media, 2007.

[156] 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, Jan. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092523121731723X

[157] M. Asif, P. Angelov, and H. Ahmed, "An approach to real-time colorbased object tracking," in *International Symposium on Evolving Fuzzy Systems*, 2006, pp. 86–91.

[158] 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.

[159] S. Wu, M. J. Er, and Y. Gao, "A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 578–594, 2001.

[160] 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.

[161] M. Sayed-Mouchaweh and E. Lughofer, *Learning in non-stationary environments: methods and applications.* Springer Science & Business Media, 2012.

[162] 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.

[163] 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.

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

[165] D. Leite, R. Ballini, P. Costa, and F. Gomide, "Evolving fuzzy granular modeling from nonstationary fuzzy data streams," *Evolving Systems*, vol. 3, no. 2, pp. 65–79, 2012.

[166] 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, 2014.

[167] M. Pratama, S. G. Anavatti, M. Garratt, and E. Lughofer, "Online identification of complex multi-input-multi-output system based on generic evolving neuro-fuzzy inference system," in *2013 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*. IEEE, 2013, pp. 106–113.

[168] M. M. Ferdaus, S. G. Anavatti, M. A. Garratt, and M. Pratama, "Evolving fuzzy inference system based online identification and control of a quadcopter unmanned aerial vehicle," in *2017 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)*. IEEE, 2017, pp. 223–228.

[169] C. F. Juang and K. J. Juang, "Circuit Implementation of Data-Driven TSK-Type Interval Type-2 Neural Fuzzy System With Online Parameter Tuning Ability," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4266–4275, May 2017.

[170] P. P. Angelov and X. Zhou, "Evolving Fuzzy-Rule-Based Classifiers From Data Streams," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1462–1475, Dec. 2008.

[171] C.-F. Hsu and K.-Y. Wong, "On-line constructive fuzzy sliding-mode control for voice coil motors," *Applied Soft Computing*, vol. 47, pp. 415–423, 2016.

[172] X. Yu, Y. Fu, P. Li, and Y. Zhang, "Fault-tolerant aircraft control based on self-constructing fuzzy neural networks and multivariable SMC under actuator faults," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2324–2335, 2017.

[173] C.-S. Chen and W.-C. Lin, "Self-adaptive interval Type-2 neural fuzzy network control for PMLSM drives," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14 679–14 689, 2011.

[174] C.-M. Lin and T.-Y. Chen, “Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems,” *IEEE Transactions on Neural Networks*, vol. 20, no. 9, pp. 1377–1384, 2009.

[175] T. Huynh, C. Lin, T. Le, H. Cho, T. T. Pham, N. Le, and F. Chao, “A new self-organizing fuzzy cerebellar model articulation controller for uncertain nonlinear systems using overlapped Gaussian membership functions,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, pp. 9671–9682, 2020.

[176] T.-L. Le, T.-T. Huynh, and C.-M. Lin, “Self-evolving interval type-2 wavelet cerebellar model articulation control design for uncertain nonlinear systems using PSO,” *International Journal of Fuzzy Systems*, vol. 21, no. 8, pp. 2524–2541, 2019.

[177] T.-L. Le, C.-M. Lin, and T.-T. Huynh, “Self-evolving type-2 fuzzy brain emotional learning control design for chaotic systems using PSO,” *Applied Soft Computing*, vol. 73, pp. 418–433, 2018.

[178] M. Perhinschi, M. Napolitano, and S. Tamayo, “Integrated simulation environment for unmanned autonomous systems—towards a conceptual framework,” *Modelling and Simulation in Engineering*, vol. 2010, 2010.

[179] C. Ding and L. Lu, “A tilting-rotor unmanned aerial vehicle for enhanced aerial locomotion and manipulation capabilities: Design, control, and applications,” *IEEE/ASME Transactions on Mechatronics*, 2020.

[180] D. Floreano and R. J. Wood, “Science, technology and the future of small autonomous drones,” *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.

[181] Y. Zhang, X. Yuan, W. Li, and S. Chen, “Automatic power line inspection using UAV images,” *Remote Sensing*, vol. 9, no. 8, p. 824, 2017.

[182] P. Doherty and P. Rudol, “A UAV search and rescue scenario with human body detection and geolocalization,” in *Australasian Joint Conference on Artificial Intelligence.* Springer, 2007, pp. 1–13.

[183] A. Puri, “A survey of unmanned aerial vehicles UAV for traffic surveillance,” *Department of computer science and engineering, University of South Florida*, pp. 1–29, 2005.

[184] X. Chen, D. Li, Z. Xu, and Y. Bai, “Robust control of quadrotor mav using self-organizing interval type-ii fuzzy neural networks (SOIT-IIFNNs) controller,” *International Journal of Intelligent Computing and Cybernetics*, vol. 4, no. 3, pp. 397–412, 2011.

[185] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles,” *IEEE Robotics and Automation magazine*, vol. 20, no. 32, 2012.

[186] F. Santoso, M. A. Garratt, and S. G. Anavatti, “Fuzzy logic-based self-tuning autopilots for trajectory tracking of a low-cost quadcopter: A comparative study,” in *2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA).* IEEE, Oct 2015, pp. 64–69.

[187] V. P. Tran, F. Santoso, M. A. Garratt, and I. R. Petersen, “Adaptive second-order strictly negative imaginary controllers based on the interval Type-2 fuzzy self-tuning systems for a hovering quadrotor with uncertainties,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 1, pp. 11–20, 2020.

[188] B. Martin, D. C. Tarraf, T. C. Whitmore, J. DeWeese, C. Kenney, J. Schmid, and P. DeLuca, *Advancing Autonomous Systems: An Analysis of Current and Future Technology for Unmanned Maritime Vehicles.* Santa Monica, CA: RAND Corporation, 2019.

[189] F. Kendoul, “Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems,” *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, 2012.

[190] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, “A survey of quadrotor unmanned aerial vehicles,” in *2012 Proceedings of IEEE Southeastcon.* IEEE, 2012, pp. 1–6.

[191] M. Jacquet and A. Franchi, "Motor and perception constrained nmpc for torque-controlled generic aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 518–525, 2020.

[192] K. Boudjit and C. Larbes, "Detection and target tracking with a quadrotor using fuzzy logic," in *2016 8th International Conference on Modelling, Identification and Control (ICMIC)*. IEEE, 2016, pp. 127–132.

[193] M. Liu, G. K. Egan, and F. Santoso, "Modeling, autopilot design, and field tuning of a UAV with minimum control surfaces," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2353–2360, 2015.

[194] F. Santoso, M. Liu, and G. Egan, "$H_2$ and $H_\infty$ robust autopilot synthesis for longitudinal flight of a special unmanned aerial vehicle: a comparative study," *IET Control Theory & Applications*, vol. 2, no. 7, pp. 583–594(11), Jul. 2008. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-cta_20070415

[195] R. Babuška and H. B. Verbruggen, "An overview of fuzzy modeling for control," *Control Engineering Practice*, vol. 4, no. 11, pp. 1593–1606, 1996.

[196] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," in *Readings in Fuzzy Sets for Intelligent Systems*. Elsevier, 1993, pp. 387–403.

[197] S. Shan and Z. Hou, "Neural network NARMAX model based unmanned aircraft control surface reconfiguration," in *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1. IEEE, 2016, pp. 154–157.

[198] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," *MSc Theses*, 2008.

[199] E.-H. Zheng, J.-J. Xiong, and J.-L. Luo, "Second order sliding mode control for a quadrotor UAV," *ISA Transactions*, vol. 53, no. 4, pp. 1350–1356, 2014.

[200] B. Ahmed, H. R. Pota, and M. Garratt, "Flight control of a rotary wing UAV using backstepping," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 20, no. 6, pp. 639–658, 2010.

[201] F. Santoso, M. A. Garratt, S. G. Anavatti, and I. Petersen, "Robust hybrid nonlinear control systems for the dynamics of a quadcopter drone," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1–13, 2018.

[202] A. J. Al-Mahasneh, S. G. Anavatti, M. A. Garratt, and M. Pratama, "Stable adaptive controller based on generalized regression neural networks and sliding mode control for a class of nonlinear time-varying systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2019.

[203] H. Yu, T. Xie, S. Paszczynski, and B. M. Wilamowski, "Advantages of radial basis function networks for dynamic system design," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 12, pp. 5438–5450, 2011.

[204] L. Cervantes and O. Castillo, "Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control," *Information Sciences*, vol. 324, pp. 247–256, 2015.

[205] P. Ponce, A. Molina, I. Cayetano, J. Gallardo, H. Salcedo, J. Rodriguez, and I. Carrera, "Fuzzy logic sugeno controller type-2 for quadrotors based on anfis," in *Nature-Inspired Computing for Control Systems.* Springer, 2016, pp. 195–230.

[206] O. Hassanein, S. G. Anavatti, and T. Ray, "On-line adaptive fuzzy modeling and control for autonomous underwater vehicle," in *Recent advances in robotics and automation.* Springer, 2013, pp. 57–70.

[207] G. C. Karras, P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Unsupervised online system identification for underwater robotic vehicles," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 3, pp. 642–663, 2019.

[208] E. Y. Hong, T. K. Meng, and M. Chitre, “Online system identification of the dynamics of an autonomous underwater vehicle,” in *2013 IEEE International Underwater Technology Symposium (UT)*, 2013, pp. 1–10.

[209] Y. H. Eng, K. M. Teo, M. Chitre, and K. M. Ng, “Online system identification of an autonomous underwater vehicle via in-field experiments,” *IEEE Journal of Oceanic Engineering*, vol. 41, no. 1, pp. 5–17, 2016.

[210] L. Zhang, D.-p. Jiang, J.-x. Zhao, and S. Ma, “An AUV for ocean exploring and its motion control system architecture,” *The open mechanical engineering journal*, vol. 7, no. 1, 2013.

[211] J. Farrell, S. Pang, W. Li, and R. Arrieta, “Biologically inspired chemical plume tracing demonstrated on an autonomous underwater vehicle,” *at IEEE Systems, Man, and Cybernetics Conference*, 2004.

[212] M. H. Khodayari and S. Balochian, “Modeling and control of autonomous underwater vehicle (AUV) in heading and depth attitude via self-adaptive fuzzy PID controller,” *Journal of Marine Science and Technology*, vol. 20, no. 3, pp. 559–578, 2015.

[213] S. Soylu, B. J. Buckham, and R. P. Podhorodeski, “A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation,” *Ocean Engineering*, vol. 35, no. 16, pp. 1647–1659, 2008.

[214] P. Maurya, E. Desa, A. Pascoal, E. Barros, G. Navelkar, R. Madhan, A. Mascarenhas, S. Prabhudesai, S. Afzulpurkar, A. Gouveia *et al.*, “Control of the maya AUV in the vertical and horizontal planes: Theory and practical results,” in *Proceedings of the 7th IFAC Conference on Manoeuvring and Control of Marine Craft*, 2006, pp. 20–22.

[215] A. Al-Mahturi and H. Wahid, “Optimal tuning of linear quadratic regulator controller using a particle swarm optimization for two-rotor aerodynamical system,” *Int. J. Electr. Comput. Energ. Electron. Commun. Eng*, vol. 11, no. 2, pp. 184–190, 2017.

[216] C. Shen and Y. Shi, "Distributed implementation of nonlinear model predictive control for AUV trajectory tracking," *Automatica*, vol. 115, p. 108863, 2020.

[217] C. Shen, Y. Shi, and B. Buckham, "Path-following control of an AUV: A multiobjective model predictive control approach," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1334–1342, 2018.

[218] ——, "Trajectory tracking control of an autonomous underwater vehicle using lyapunov-based model predictive control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5796–5805, 2017.

[219] Q. Ha, L. Yen, and C. Balaguer, "Robotic autonomous systems for earthmoving in military applications," *Automation in Construction*, vol. 107, p. 102934, 2019.

[220] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, "Autonomous ground vehicles—concepts and a path to the future," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1831–1839, 2012.

[221] S. Bonadies and S. A. Gadsden, "An overview of autonomous crop row navigation strategies for unmanned ground vehicles," *Engineering in Agriculture, Environment and Food*, vol. 12, no. 1, pp. 24–31, 2019.

[222] M.-O. Sonneberg, M. Leyerer, A. Kleinschmidt, F. Knigge, and M. H. Breitner, "Autonomous unmanned ground vehicles for urban logistics: Optimization of last mile delivery operations," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[223] D. Chwa, "Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 6, pp. 1285–1295, 2010.

[224] ——, "Fuzzy adaptive tracking control of wheeled mobile robots with state-dependent kinematic and dynamic disturbances," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 3, pp. 587–593, 2011.

[225] A. Zhu and S. X. Yang, "Neurofuzzy-based approach to mobile robot navigation in unknown environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 4, pp. 610–621, 2007.

[226] I. Baturone, F. J. Moreno-Velo, V. Blanco, and J. Ferruz, "Design of embedded dsp-based fuzzy controllers for autonomous mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 2, pp. 928–936, 2008.

[227] D. Dovžan and I. Škrjanc, "Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes," *ISA Transactions*, vol. 50, no. 2, pp. 159–169, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001905781100005X

[228] Zhaolin Yang, Feng Lin, and B. M. Chen, "Survey of autopilot for multi-rotor unmanned aerial vehicles," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6122–6127.

[229] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2992–2997.

[230] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1, pp. 21–39, 2012.

[231] A. autopilot, "Ardupilot development guide," http://ardupilot.org/ardupilot/, accessed: 2021.

[232] P. Autopilot, "Px4 development guide," http://https://px4.io/, accessed: 2021.

[233] R. G. Beall, "Engineering of fast and robust adaptive control for fixed-wing unmanned aircraft," Naval Postgraduate School Monterey United States, Tech. Rep., 2017.

[234] V. motion systems limited, "Vicon tracker user guide," https://www.vicon.com/software/tracker/, 2016.

[235] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro air vehicle link (mavlink) in a nutshell: A survey," *IEEE Access*, vol. 7, pp. 87 658–87 680, 2019.

[236] QGroundControl, "Open source mav ground control station," *MAVLink Micro Air Vehicle Communication Protocol. Available online: http://qgroundcontrol.org (accessed on 2021)*, 2018.

[237] J. M. O'Kane, *A Gentle Introduction to ROS.* Independently published, Oct. 2013, available at http://www.cse.sc.edu/~jokane/agitr/.

[238] A. Martinez and E. Fernández, *Learning ROS for robotics programming.* Packt Publishing Ltd, 2013.

[239] O. I. H. Hassan, "Black-box identification and control for autonomous underwater vehicles," Ph.D. dissertation, PhD. Dissertation, School of Engineering and Information Technology, UNSW Canberra, 2013.

[240] F. Santoso, M. A. Garratt, S. G. Anavatti, and O. Hassanein, "Entropy fuzzy system identification for the dynamics of the dragonfly-like flapping wing aircraft," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).* IEEE, 2018, pp. 1–8.

[241] M. M. Ferdaus, M. Pratama, S. G. Anavatti, and M. A. Garratt, "Online identification of a rotary wing unmanned aerial vehicle from data streams," *Applied Soft Computing*, vol. 76, pp. 313–325, 2019.

[242] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "A simplified model-free self-evolving TS fuzzy controller for nonlinear systems with uncertainties," in *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, May 2020, pp. 1–6.

[243] M. Ashrafi, D. K. Prasad, and C. Quek, "IT2-GSETSK: An evolving interval type-II TSK fuzzy neural system for online modeling of noisy

data," *Neurocomputing*, vol. 407, pp. 1–11, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220304380

[244] Q. Ren, L. Baron, and M. Balazinski, "Type-2 Takagi-Sugeno-Kang fuzzy logic modeling using subtractive clustering," in *NAFIPS 2006 - 2006 Annual Meeting of the North American Fuzzy Information Processing Society*, 2006, pp. 120–125.

[245] 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.

[246] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "A robust self-adaptive interval type-2 ts fuzzy logic for controlling multi-input–multi-output nonlinear uncertain dynamical systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 655–666, 2022.

[247] T. Zhang, S. S. Ge, and C. C. Hang, "Adaptive neural network control for strict-feedback nonlinear systems using backstepping design," *Automatica*, vol. 36, no. 12, pp. 1835–1846, 2000.

[248] W. Sun, Y. Liu, and H. Gao, "Constrained sampled-data ARC for a class of cascaded nonlinear systems with applications to motor-servo systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 766–776, 2018.

[249] S. Lin and A. A. Goldenberg, "Neural-network control of mobile manipulators," *IEEE Transactions on Neural networks*, vol. 12, no. 5, pp. 1121–1133, 2001.

[250] W. Sun, J. Zhang, and Z. Liu, "Two-time-scale redesign for antilock braking systems of ground vehicles," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4577–4586, 2018.

[251] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Hybrid PD-fuzzy and PD controllers for trajectory tracking of a quadrotor unmanned aerial vehicle: Autopilot designs and real-time flight tests," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2019.

[252] M. M. Ferdaus, S. G. Anavatti, M. A. Garratt, and M. Pratama, “Development of c-means clustering based adaptive fuzzy controller for a flapping wing micro air vehicle,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 2, pp. 99–109, 2019.

[253] O. Hassanein, S. G. Anavatti, H. Shim, and T. Ray, “Model-based adaptive control system for autonomous underwater vehicles,” *Ocean Engineering*, vol. 127, pp. 58–69, 2016.

[254] X. Lu, Y. Zhao, and M. Liu, “Self-learning interval Type-2 fuzzy neural network controllers for trajectory control of a delta parallel robot,” *Neurocomputing*, vol. 283, pp. 107–119, 2018.

[255] C.-T. Lin, N. R. Pal, S.-L. Wu, Y.-T. Liu, and Y.-Y. Lin, “An interval Type-2 neural fuzzy system for online system identification and feature elimination,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1442–1455, 2015.

[256] J. M. Mendel, *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions.* Springer, 2019.

[257] S.-J. Huang and W.-C. Lin, “Adaptive fuzzy controller with sliding surface for vehicle suspension control,” *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 4, pp. 550–559, 2003.

[258] K. S. Narendra and A. M. Annaswamy, “A new adaptive law for robust adaptation without persistent excitation,” in *1986 American Control Conference*, June 1986, pp. 1067–1072.

[259] F. Chen and H. K. Khalil, “Adaptive control of nonlinear systems using neural networks - a dead-zone approach,” in *1991 American Control Conference*, June 1991, pp. 667–672.

[260] F. Santoso, M. A. Garratt, and S. G. Anavatti, “Hybrid PD-fuzzy and PD controllers for trajectory tracking of a quadrotor unmanned aerial vehicle: Autopilot designs

and real-time flight tests," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2019.

[261] J. M. Seddon and S. Newman, *Basic helicopter aerodynamics.* John Wiley & Sons, 2011, vol. 40.

[262] H. Glauert, "The theory of the autogyro," *The Aeronautical Journal*, vol. 31, no. 198, pp. 483–508, 1927.

[263] R. C. Nelson *et al.*, *Flight stability and automatic control.* WCB/McGraw Hill New York, 1998, vol. 2.

[264] M. A. Garratt, "Biologically inspired vision and control for an autonomous flying vehicle," Ph.D. dissertation, The Australian National University, 2007.

[265] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation.* John Wiley and Sons, 2016.

[266] D. Wu, "Approaches for reducing the computational cost of interval Type-2 fuzzy logic systems: overview and comparisons," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 80–99, 2012.

[267] M. Tinkir, M. Kalyoncu, U. Onen, and F. M. Botsali, "PID and interval type-2 fuzzy logic control of double inverted pendulum system," in *The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010*, vol. 1. IEEE, 2010, pp. 117–121.

[268] L. B. Prasad, B. Tyagi, and H. O. Gupta, "Optimal control of nonlinear inverted pendulum system using PID controller and LQR: performance analysis without and with disturbance input," *International Journal of Automation and Computing*, vol. 11, no. 6, pp. 661–670, 2014.

[269] X. Yu and O. Kaynak, "Sliding-mode control with soft computing: A survey," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3275–3285, 2009.

[270] F. Santoso, M. A. Garratt, and S. G. Anavatti, "A self-learning ts-fuzzy system based on the c-means clustering technique for controlling the altitude of a hexacopter unmanned aerial vehicle," in *2017 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA).* IEEE, 2017, pp. 46–51.

[271] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "Nonlinear altitude control of a quadcopter drone using interval type-2 fuzzy logic," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE, 2018, pp. 1–6.

[272] E. Zakeri, S. Farahat, S. A. Moezi, and A. Zare, "Path planning for unmanned underwater vehicle in 3D space with obstacles using spline-imperialist competitive algorithm and optimal interval type-2 fuzzy logic controller," *Latin American Journal of Solids and Structures*, vol. 13, no. 6, pp. 1054–1085, 2016.

[273] F. Ali, E. K. Kim, and Y.-G. Kim, "Type-2 fuzzy ontology-based semantic knowledge for collision avoidance of autonomous underwater vehicles," *Information Sciences*, vol. 295, pp. 441–464, 2015.

[274] P. Londhe and B. Patre, "Adaptive fuzzy sliding mode control for robust trajectory tracking control of an autonomous underwater vehicle," *Intelligent Service Robotics*, pp. 1–16, 2018.

[275] T. I. Fossen, *Guidance and control of ocean vehicles.* John Wiley & Sons Inc, 1994.

[276] J. Javadi-Moghaddam and A. Bagheri, "An adaptive neuro-fuzzy sliding mode based genetic algorithm control system for under water remotely operated vehicle," *Expert Systems with Applications*, vol. 37, no. 1, pp. 647–660, 2010.

[277] E. Kayacan and O. Kaynak, "Sliding mode control theory-based algorithm for online learning in type-2 fuzzy neural networks: application to velocity control of an electro hydraulic servo system," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 7, pp. 645–659, 2012.

[278] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, “Self-learning in aerial robotics using type-2 fuzzy systems: Case study in hovering quadrotor flight control,” *IEEE Access*, vol. 9, pp. 119 520–119 532, 2021.

[279] A. Matus-Vargas, G. Rodriguez-Gomez, and J. Martinez-Carranza, “Ground effect on rotorcraft unmanned aerial vehicles: a review,” *Intelligent Service Robotics*, pp. 1–20, 2021.

[280] F. Santoso, M. A. Garratt, S. G. Anavatti, and J. Wang, “Chapter 1 - evolutionary aerial robotics: the human way of learning,” in *Unmanned Aerial Systems*, ser. Advances in Nonlinear Dynamics and Chaos (ANDC), A. Koubaa and A. T. Azar, Eds. Academic Press, 2021, pp. 1–23. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978012820276000008X

[281] A. Eltayeb, M. F. Rahmat, M. A. M. Basri, M. A. M. Eltoum, and S. El-Ferik, “An improved design of an adaptive sliding mode controller for chattering attenuation and trajectory tracking of the quadcopter uav,” *IEEE Access*, vol. 8, pp. 205 968–205 979, 2020.

[282] H. Li, J. Wang, H. Lam, Q. Zhou, and H. Du, “Adaptive sliding mode control for interval type-2 fuzzy systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 12, pp. 1654–1663, 2016.

[283] S. Bouabdallah, “Design and control of quadrotors with application to autonomous flying,” EPFL, Tech. Rep., 2007.

[284] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, “Chapter 2 - modeling and control of a quadrotor unmanned aerial vehicle using type-2 fuzzy systems,” in *Unmanned Aerial Systems*, ser. Advances in Nonlinear Dynamics and Chaos (ANDC), A. Koubaa and A. T. Azar, Eds. Academic Press, 2021, pp. 25–46. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128202760000091

[285] C. Li, G. Zhang, J. Yi, and M. Wang, “Uncertainty degree and modeling of interval type-2 fuzzy sets: definition, method and application,” *Computers & Mathematics with Applications*, vol. 66, no. 10, pp. 1822–1835, 2013.

[286] J. Lin, R.-J. Lian, C.-N. Huang, and W.-T. Sie, “Enhanced fuzzy sliding mode controller for active suspension systems,” *Mechatronics*, vol. 19, no. 7, pp. 1178–1190, 2009.

[287] A. Saghafinia, H. W. Ping, M. N. Uddin, and K. S. Gaeid, “Adaptive fuzzy sliding-mode control into chattering-free IM drive,” *IEEE Transactions on Industry Applications*, vol. 51, no. 1, pp. 692–701, 2014.

[288] A. J. Al-Mahasneh, S. G. Anavatti, and M. A. Garratt, “Self-evolving neural control for a class of nonlinear discrete-time dynamic systems with unknown dynamics and unknown disturbances,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6518–6529, 2020.

[289] R. Lian, “Enhanced adaptive self-organizing fuzzy sliding-mode controller for active suspension systems,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 3, pp. 958–968, March 2013.

[290] A. Al-Mahturi, F. Santoso, M. A. Garratt, S. G. Anavatti, and M. M. Ferdaus, “A simplified model-free self-evolving ts fuzzy controller for nonlinear systems with uncertainties,” in *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, May 2020, pp. 1–6.

[291] J. Zou and J. K. Schueller, “Adaptive backstepping control for parallel robot with uncertainties in dynamics and kinematics,” *Robotica*, vol. 32, no. 6, p. 2, 2014.

[292] C.-F. Juang and Y.-C. Chang, “Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments,” *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 379–392, 2011.

[293] J. Huang, M. Ri, D. Wu, and S. Ri, “Interval Type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum,” *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 4, pp. 2030–2038, 2017.

[294] O. Linda and M. Manic, “Uncertainty-robust design of interval Type-2 fuzzy logic controller for delta parallel robot,” *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 661–670, 2011.

[295] L. Astudillo, O. Castillo, P. Melin, A. Alanis, J. Soria, and L. T. Aguilar, “Intelligent control of an autonomous mobile robot using type-2 fuzzy logic.” *Engineering Letters*, vol. 13, no. 3, 2006.

[296] E. Lughofer, “Towards robust evolving fuzzy systems,” *Evolving Intelligent Systems: Methodology and Applications*, pp. 87–126, 2010.

[297] P. Angelov and X. Zhou, “Evolving fuzzy systems from data streams in real-time,” in *2006 International Symposium on Evolving Fuzzy Systems.* IEEE, 2006, pp. 29–35.

[298] H.-G. Han, Z.-Y. Chen, H.-X. Liu, and J.-F. Qiao, “A self-organizing interval Type-2 fuzzy-neural-network for modeling nonlinear systems,” *Neurocomputing*, vol. 290, pp. 196–207, May 2018.

[299] J. Liu, *Intelligent Control Design and MATLAB Simulation.* Springer, 2018.

[300] F. N. Martins, W. C. Celeste, R. Carelli, M. Sarcinelli-Filho, and T. F. Bastos-Filho, “An adaptive dynamic controller for autonomous mobile robot trajectory tracking,” *Control Engineering Practice*, vol. 16, no. 11, pp. 1354–1363, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066108000373

[301] J. Zhang, Q. Li, X. Chang, F. Chao, C.-M. Lin, L. Yang, T. T. Huynh, L. Zheng, C. Zhou, and C. Shang, “A novel self-organizing emotional cmac network for robotic control,” in *2020 International Joint Conference on Neural Networks (IJCNN).* IEEE, 2020, pp. 1–6.