Systematic Design of Type-2 Fuzzy Logic Systems for Modeling and Control with Applications to Modular and Reconfigurable Robots

by

Mohammad Biglar Begian

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosohpy
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2010

Abstract

Fuzzy logic systems (FLSs) are well known in the literature for their ability to model linguistics and system uncertainties. Due to this ability, FLSs have been successfully used in modeling and control applications such as medicine, finance, communications, and operations research. Moreover, the ability of higher order fuzzy systems to handle system uncertainty has become an interesting topic of research in the field. In particular, type-2 FLSs (T2 FLSs), systems consisting of fuzzy sets with fuzzy grades of membership, a feature that type-1 (T1) does not offer, are most well-known for this capability. The structure of T2 FLSs allows for the incorporation of uncertainty in the input membership grades, a common situation in reasoning with physical systems. General T2 FLSs have a complex structure, thus making them difficult to adopt on a large scale. As a result, interval T2 FLSs (IT2 FLSs), a special class of T2 FLSs, have recently shown great potential in various applications with input-output (I/O) system uncertainties.

Due to the sophisticated mathematical structure of IT2 FLSs, little to no systematic analysis has been reported in the literature to use such systems in control design. Moreover, to date, designers have distanced themselves from adopting such systems on a wide scale because of their design complexity. Furthermore, the very few existing control methods utilizing IT2 fuzzy logic control systems (IT2 FLCSs) do not guarantee the stability of their system. Therefore, this thesis presents a systematic method for designing stable IT2 Takagi-Sugeno-Kang (IT2 TSK) fuzzy systems when antecedents are T2 fuzzy sets and consequents are crisp numbers (A2-C0). Five new inference mechanisms are proposed that have closed-form I/O mappings, making them more feasible for FLCS stability analysis. The thesis focuses on control applications for when (a) both plant and controller use A2-C0 TSK models, and (b) the plant uses T1 Takagi-Sugeno (T1 TS) and the controller uses IT2 TS models. In both cases, sufficient stability conditions for the stability of the closed-loop system are derived. Furthermore, novel linear matrix inequality-based algorithms are developed for satisfying the stability conditions. Numerical analyses are included to validate the effectiveness of the new inference methods. Case studies reveal that a well-tuned IT2 TS FLCS using the proposed inference engine can potentially outperform its T1 TSK counterpart, a result of IT2 having greater structural flexibility than T1. Moreover, due to the simple nature of the proposed inference engine, it is easy to implement in real-time control systems.

In addition, a novel design methodology is proposed for IT2 TSK FLC for modular and reconfigurable robot (MRR) manipulators with uncertain dynamic parameters. A mathematical framework for the design of IT2 TSK FLCs is developed for tracking purposes that can be effectively used in real-time applications. To verify the effectiveness of the proposed controller, experiments are performed on an MRR with two degrees of freedom which exhibits dynamic coupling behavior. Results show that the developed controller can outperform some well-known linear and nonlinear controllers for different configurations. Therefore, the proposed structure can be adopted for the position control of MRRs with unknown dynamic parameters in trajectory-tracking applications.

Finally, a rigorous mathematical analysis of the robustness of FLSs (both T1 and IT2) is presented in the thesis and entails a formulation of the robustness of FLSs as a constraint multi-objective optimization problem. Consequently, a procedure is proposed for the design of robust IT2 FLSs. Several examples are presented to demonstrate the effectiveness of the proposed methodologies. It was concluded that both T1 and IT2 FLSs can be designed to achieve robust behavior in various applications. IT2 FLSs, having a more flexible structure than T1 FLSs, exhibited relatively small approximation errors in the several examples investigated.

The rigorous methodologies presented in this thesis lay the mathematical foundations for analyzing the stability and facilitating the design of stabilizing IT2 FLCSs. In addition, the proposed control technique for tracking purposes of MRRs will provide control engineers with tools to control dynamic systems with uncertainty and changing parameters. Finally, the systematic approach developed for the analysis and design of robust T1 and IT2 FLSs is of great practical value in various modeling and control applications.

Acknowledgements

The completion of this thesis would not have been possible without the help and encouragement of many individuals to whom I would like to express my gratitude.

First and foremost, I would like to thank my supervisors, Professor William Melek and Professor Jerry Mendel, for their constant support and interest in my work. William has been an outstanding advisor, who through our close relationship at work, has become a very good friend. His positive attitude toward research is unique and made me believe that I can aim for the highest goals. He provided me with everything I asked for and prepared a very comfortable working environment. I was fortunate to find him when I was at Waterloo. As well, I would like to thank Professor Mendel for taking a major responsibility in advising me from far away. During our collaboration, I have learnt to a great extent from him. He is an exemplary researcher, whose meticulousness and thought-provoking questions enriched the quality of my own research work. It has been an honor working with him.

I would like to thank the government of Canada and the Ministry of Education for Ontario for providing me with scholarships, NSERC and OGS, which further facilitated conducting my PhD research.

Thanks also go to my friends at the “Computational Intelligence and Robotics” lab, in particular, I would like to thank Zai Li for helping me to get started with the robot tests. I would also like to thank Professor Daniel Davison from the Electrical Engineering Department for answering my control questions. In addition, I would like to express my gratitude to my external examiner, Professor Ying, and my committee members, especially Professor Karray for his counsel when I wanted to move onto a new research topic.

I appreciate the feedbacks and comments that I received from the “UW Grad Writing Center” on the quality of my writing. Specifically, I would like to express my gratitude to Mary McPherson for her help. Thanks also go to all my wonderful friends especially Mike, Alex, and David for being supportive in the last few years.

I am indebted to my father who nurtured the love of math in me. My passion in research and learning has been greatly inspired by him. My mother has had an equal influence on me. She has been with me all these years. Her love and compassionate have always been a motivating force. I do not think I can thank her enough. Where I stand today is entirely because of my parents. I would also like to thank my brothers for their constant encouragement, especially Behzad, who has joined me in Canada and been very encouraging.

Finally, I would like to thank my great teachers and professors in Iran for their dedication and excellence in teaching and education.

Contents

List of Tables

List of Figures

Nomenclature

α, β real numbers

$\mu_{\widetilde{F}_j^i}$ jth type-1 membership function of rule i

$\underline{\mu}_{\widetilde{F}_j^i}$ ith lower type-2 membership function of rule i

$\overline{\mu}_{\widetilde{F}_j^i}$ ith upper type-2 membership function of rule i

a_j^i, b_j^i, c_j^i jth coefficient of TSK output for rule i

$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, $\boldsymbol{E}$, $\boldsymbol{A}_i$,$\boldsymbol{B}_{i,l}$ $n \times n$ matrices

$\boldsymbol{e}$ output tracking error

$\dot{\boldsymbol{e}}$ rate of change of output tracking error

$F_B, G_B, \epsilon_n, D_B, Kp, Kd$ positive constants

f^i type-1 firing strength of rule i

$\underline{f}^i, \overline{f}^i$ type-2 lower and upper rule firing strengths

F_j^i type-1 fuzzy set of input state j in rule i

$\widetilde{F}_j^i$ type-2 fuzzy set of input state j in rule i

$\boldsymbol{F}(\dot{\boldsymbol{q}})$ friction terms

g_1, g_2 functions of upper and lower firing levels

$\boldsymbol{G}(\boldsymbol{q})$ gravity vector

$\boldsymbol{K}_{PD}$ design gain matrix of a PD controller

m, n, m', n', q tuning parameters

M number of rules

$\boldsymbol{M}(\boldsymbol{q})$ inertia matrix

$\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{\Lambda}, \boldsymbol{F}$ $n \times n$ positive definite matrices

$\boldsymbol{q}$ joint variable (trajectory)

$\dot{\boldsymbol{q}}$ rate of change of joint trajectory

$\boldsymbol{q}_d$ desired trajectory

$\dot{\boldsymbol{q}}_d$ rate of change of desired trajectory

$\boldsymbol{r}$ filtered tracking error

$\boldsymbol{r}_1, \boldsymbol{r}_2$ $n \times 1$ vector

R_i robustness index

u control input

$\boldsymbol{u}_e$ compensation torque

$\boldsymbol{u}_{Fuzzy}$ fuzzy controller effort

$\boldsymbol{u}_{PD}$ control effort of a PD controller

$\widehat{\boldsymbol{u}_e}$ difference between the function and its estimated value

$\underline{v}^i, \overline{v}^i$ lower and upper rule firing strengths of controller

$\boldsymbol{V}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})$ coriolis matrix

$V(\boldsymbol{x}(k))$, $\Delta V(\boldsymbol{x}(k))$ Lyapunov function, gradient of Lyapunov function

$\underline{w}^i, \overline{w}^i$ lower and upper rule firing strengths of plant

$\boldsymbol{x}(k)$ $n \times 1$ state vector

$\boldsymbol{x}(k+1)$ $n \times 1$ output vector

$x(k-i)$ ith state variable

y_i ith rule output

y_l, y_r left and right uncertainty bounds

$\underline{y}_l$,$\overline{y}_l$,$\underline{y}_r$,$\overline{y}_r$ Wu-Mendel Uncertainty Bounds

Y_{MTSK} output of a modified type-1 TSK fuzzy logic system

Y_{output} final crisp output of an interval type-2 fuzzy system

$Y_{\text{TSK/A2-C0}}$ type-1 output of an interval type-2 TSK A2-C0 fuzzy system

Y_{WM} crisp output of the interval type-2 fuzzy system using Wu-Mendel Uncertainty Bounds

$\boldsymbol{Z}, \boldsymbol{Z}_1$ $n \times n$ negative definite matrices

$\boldsymbol{\tau}$ torque vector or control input

$\boldsymbol{\tau}_d$ disturbances

Δx_{max} maximum allowable input deviation of a fuzzy logic system

$\Delta Y, \Delta Y_{desired}$ output deviation and desired output deviation of a fuzzy logic system

List of Abbreviations

A2-C0 antecedents type-2 fuzzy sets and consequents crisp numbers

EKM enhanced Karnik-Mendel

EPI error performance index

ET1S equivalent type-1 set

FLC fuzzy logic control

FLS fuzzy logic system

FOU footprint of uncertainty

FS fuzzy set

I/O input-output

IT2 FS interval type-2 fuzzy set

IT2 FLS interval type-2 fuzzy logic system

IT2 FLC interval type-2 fuzzy logic controller

IT2 TSK FLC interval type-2 Takagi-Sugeno-Kang fuzzy logic controller

IT2 TSK FLCS interval type-2 Takagi-Sugeno-Kang fuzzy logic control system

IT2 TSK FLS interval type-2 Takagi-Sugeno-Kang fuzzy logic system

KM Karnik Mendel

LMI linear matrix inequality

MF membership function

MRR modular and reconfigurable robot

MSE mean squared error

PD proportional-derivative

PI proportional-integral

PID proportional-integral-derivative

PIm percentage improvement

RM robot manipulator

RPI robustness performance improvement

T2 FLS type-2 fuzzy logic system

T2 FLC type-2 fuzzy logic control

T1 FLS type-1 fuzzy logic system

T1 FLC type-1 fuzzy logic controller

T2 FLCS type-2 fuzzy logic controller

T2 FS type-2 fuzzy set

TR type-reduction

TS Takagi-Sugeno

TSK Takagi-Sugeno-Kang

WM UBs Wu-Mendel uncertainty bounds

Chapter 1

Introduction

1.1 Motivation

The theory of fuzzy sets (FSs) was introduced by Zadeh [2] in 1965, which led to the advent of fuzzy logic systems (FLSs). In general, FLSs are well known in the literature for their ability to model linguistics and uncertainties in systems [3], [1], [4]. Because of this ability, FLSs have found a variety of applications in modeling, control, and computing with words, to name a few. Systems that use fuzzy logic reasonings have been used in fields such as medicine, finance, control, communications, operations research.

Higher-order FLSs are referred to as fuzzy systems whose membership grades are themselves fuzzy. The analysis of these systems has become a favorite topic for investigation due to such FLS's ability to handle system uncertainties. In the literature, type-2 FLSs (T2 FLSs) are particularly well-known for this capability. More specifically, interval T2 FLSs (IT2 FLSs)[1] have become very popular recently and have shown great promise to be used in various modeling and control applications with input-output (I/O) system uncertainties. Therefore, this dissertation deals with the design of IT2 FLSs for modeling and control applications.

Even though fuzzy logic was originally developed to model linguistic terms, interpretations, and human perceptions, the most-frequent implementation of fuzzy systems has been in control applications. To date, fuzzy logic control has been implemented with great success in many real-world applications and was also shown in some cases to outperform traditional control systems [5]. More specifically, IT2 fuzzy logic control systems (IT2 FLCSs) have of late also been applied in various applications [6], [7]. In particular, the advantages of using FLSs and specifically IT2 FLSs are as follows:

[1] Interval T2 FLSs are a special class of T2 FLSs whose secondary membership grades of their fuzzy sets are one.

- When dealing with human knowledge, the applicability of fuzzy logic becomes apparent; particularly for systems solely dependent on expert knowledge. Moreover, for control applications, if the model of the plant is not known or is highly uncertain, IT2 FLSs have a great potential to model the plant more accurately than type-1 FLSs (T1 FLSs). Classical control systems require explicit knowledge about the plant model and any associated uncertain parameters. Therefore, for systems with uncertainty, classical nonlinear control may not perform as desired if explicit knowledge about the models are not well defined. Hence, a fuzzy control system might be an attractive alternative.

- Apart from the uncertainty of a plant model, in the real world, sensor data is noisy and hence uncertain. Therefore, it is desirable to design and implement a system that can also effectively process information in an uncertain environment.

- It is possible to design IT2 FLSs that act as adaptive or robust controllers. Those FLCs can be designed with the characteristics of robustness or adaptability, according to the application or process they are intended to control.

Recently, because of their ability to model uncertainties, IT2 FLSs are now being considered for use in many applications as well as in control processes. Since IT2 FLSs have more parameters to be characterized, they provide more flexibility in modeling or control of physical systems with uncertain parameters. Hence, the possibility of achieving enhanced results over T1 FLSs can be expected.

Because of the sophisticated mathematical structure of IT2 FLSs very few analyses have been reported in the literature for the control design and, hence, designers have, up-to-date, distanced themselves from adopting those systems on a wider scale. Furthermore, most of the existing control methods do not provide insight into the design process and do not even guarantee the stability of their system. Instead, control design is accomplished through simulations or ad hoc parametric design. In an attempt to address the stability issue, new inference mechanisms are introduced that are utilized for systematically designing IT2 FLCSs. Hence, this thesis presents a rigorous mathematical analysis which also entails the design of stable IT2 FLCSs. The methods proposed herein lay the necessary foundations to effectively analyze the stability of IT2 FLCSs as well as designing stable IT2 controllers. To address the issue of control design, a practical approach is presented that enables control engineers to easily implement IT2 FLCs. More specifically, a rigorous control design methodology for MRR manipulators in tracking applications is presented. Finally, the robustness of FLSs (both T1 and IT2) is analyzed in depth and several performance measures are proposed. This analysis provides guidelines for the design of robust systems that have merit for various modeling, identification, and control problems.

1.2 Thesis contributions

This thesis addresses the design of IT2 FLSs for modeling and control applications. The contributions of this thesis are as follows:

- Novel closed-form inference engines are proposed for IT2 FLSs. The proposed inference mechanisms facilitate the analytical design of IT2 FLSs for modeling and control applications. More importantly, an easier implementation of IT2 FLCs in real-time is made possible through the employment/utilization of the closed-form engines. Numerous examples are shown to demonstrate the effectiveness of the newly developed engines.
- For a proposed inference engine, sufficient stability conditions of IT2 TSK FLCs in terms of linear matrix inequalities (LMIs) are derived and novel algorithms are presented to assess the feasibility of those LMIs to analyze the stability of IT2 FLCSs.
- A new methodology is presented for the design of IT2 FLCs with applications for MRRs. The effectiveness of the proposed approach has been validated experimentally.
- A novel methodology for the analysis of FLS robustness (both T1 and IT2) is proposed. The proposed method facilitates the design of robust FLSs for modeling and identification applications.

1.3 Literature Review

This section presents a review of the literature on T2 and IT2 FLSs. First, general and IT2 FLSs and their applications are reviewed. In subsequent sections, IT2 FLSs literature pertinent to control, robotics, and robust systems is presented.

T2 FSs were first proposed by Zadeh in 1975 as an extension of T1 FSs [8]. Some research had indicated that representation of fuzziness in systems using T1 membership functions (MF) is paradoxical because the membership grades are precise real numbers [9]. Hence, T2 FSs were introduced in the literature to model the uncertainties in systems [1], [7]. In T2 FSs, the grades of MFs are themselves fuzzy and therefore eliminate the paradox pointed out by researchers in earlier work.

Mizumoto and Tanaka [10] studied the operations of T2 FSs and examined the properties of membership grades of such sets as well as operations of algebraic products and sums. Dubois and Prade [11] presented a minimum t-norm formula for the composition of T2 relations as an extension of the T1. Karnik and Mendel [12] studied the previous research and presented a general formula for the extended composition of T2 relations [13, 14, 15] that established the theory of T2 FLSs.

Since then, T2 FLSs have been used in various modeling applications [16, 17]. Mitchel [18] utilized T2 FLSs to formulate classification problems for pattern recognition. Zeng and Liu in [16] integrated T2 FSs with traditional classifiers for pattern classification when both feature and hypothesis spaces have uncertainties. In this application, T2 FLSs showed improved performance in terms of robustness and classification rates. Recently, Liao et al. [19] used a data clustering technique to model T2 Takagi-Sugeno FLSs (T2 TS FLSs) and proposed approaches for the design of T2 TS model-based predictive FLCs.

In general, T2 FLSs are computationally involved due to the intensity of type-reduction (TR) [7]. However, when interval sets are utilized as secondary MFs, i.e., they are either zero or one, the computational complexity decreases [1]. Because of interval fuzzy sets simplicity, several studies have been carried out to better analyze their structures [13, 14]. Karnik and Mendel [12], in their seminal work, developed the TR method for general T2 FSs. This method is an iterative algorithm that requires extensive calculations even for interval fuzzy sets. On the other hand, as it is known that interval sets are characterized by left and right boundaries (end-points), this algorithm is capable of computing the end points accurately.

Owing to the effectiveness of Karnik-Mendel (KM) algorithms, they were extensively adopted [7]. In an interesting application, Zarandi et al. [20] developed a T2 fuzzy rule-based expert system for stock price analysis. The authors tested the effectiveness of their approach by predicting the stock price of an automotive factory. Results were promising and, hence, it was further suggested that the proposed methodology could be implemented in real-time stock price predictions.

Recently, it was proven in [21] that the KM algorithms converge monotonically and super-exponentially fast. It should be mentioned that when all sources of uncertainty disappear, KM's TR methods reduce to T1 FLSs. However, as Mendel [6] notes, KM's TR methods is not the only way for this design requirement to be met. In fact, other TR methods have been proposed. For example, Niewiadomski et. al. [22] defined four other kinds of TR, namely, optimistic, pessimistic, realistic, and realistic-weighted. Quite recently, Coupland and John [23] proposed a geometric representation and operation-based approach for general T2 FSs. The results of a real world example reported in their work show the considerably fast response of this system compared to the KM type-reduction method.

In order to reduce the computational complexity of KM algorithms, Wu and Mendel [24] developed uncertainty bounds for IT2 FSs that estimates the TR. They showed that the Wu-Mendel uncertainty bounds (WM UBs) can be computed without having to perform TR while achieving similar results to those of KM algorithms. Hence, it was proposed that this method be adopted for the design of IT2 FLSs.

Later, Wu and Mendel [25] developed an enhanced method called the 'Enhanced Karnik-Mendel (EKM)' algorithm and showed that it can save about two iterations, corresponding to a more than %39 reduction in computation time. Melgarejo [26] proposed a recursive algorithm to compute the generalized centroid of an IT2 FS. The new algorithm re-expresses the limits of the generalized centroid of an IT2 FS. In this work, numerical analyses were performed to investigate the performance of the proposed algorithm. Results show that this method for finding the symmetric footprint of uncertainty (FOU) is considerably faster than the KM algorithm and produces similar results. However, this method is recursive and can not be expressed as an I/O mapping with a closed-format.

While KM algorithms have been widely adopted, some new inference engines have been proposed that bypass TR. For example, Wu and Tan [27] also introduced a method that eliminates TR by defining equivalent T1 sets (ET1S) as "the collection of T1 sets that can be used in place of the footprint of uncertainties (FOU) in a T2 FLS". They considered a T2 set equivalent to a collection of E1TS. It was shown that ET1S method provides better performances than KM iterative algorithms. However, they have demonstrated this ET1S methodology for a two-input PI controller, and it is unknown this methodology generalizes to more complicated systems [6].

In recent years, researchers have been rigorously investigating the properties as well as the potentials of IT2 FLSs for numerous applications. Recently, Ying in [28, 29] has shown that IT2 FLSs are universal approximators, hence proving the capability of such systems to be used on a larger scale of modeling and control applications. In another work, Mendel et al. in [30, 31] presented the α-plane for T2 FSs which is useful for both theoretical and computational studies of these systems. Zhou et al. [32] proposed a new operator for linguistic terms in human decision-making modeled by T2 FSs. In a tutorial paper, Mendel [33] explains how to start solving problems involving IT2 FSs. Most recently, Mendel and Wu in [34] have explained how IT2 FSs are used for perceptual computing and computing with words. The increasingly ongoing research on T2 and more specifically IT2 FLSs is facilitating the utilization of these FLSs in different applications.

1.3.1 IT2 FLCSs

Even though fuzzy logic was originally developed to model linguistic terms, interpretations, and human perception, the most widely seen application of fuzzy systems has been in fuzzy control [1]. FLC has its earliest root in 1974 [35], when T1 FSs were used for both premise and consequent as part of the structure of the controller. T1 FLSs have been utilized in the control of nonlinear systems, e.g., [36, 37, 38]. To date, FLCs have been used with great success in many real-world applications [39, 40] and were also shown in some cases to outperform traditional control systems [41, 5, 42, 43].

Takagi-Sugeno-Kang (TSK) models were introduced in [44] and [45] to develop a systematic design of fuzzy systems in I/O mappings. Takagi-Sugeno [44] presented their method to design consequent parameters of a general fuzzy TSK model based on least-square methods. Sugeno and Kang [45] presented a strategy for structure identification of fuzzy systems. Since then, TSK models have been widely used in control applications [46, 37, 36, 47, 48, 49].

Proving the stability of FLCSs that use Mamdani inference engine is still a challenging problem [50]. Using TSK fuzzy systems, stability has been extensively investigated and solid mathematical foundations have been established; very compatible with conventional works in control theory. Tanaka and Sugeno were the first researchers to study the stability of T1 TSK FLCSs [51]. They modeled a general T1 FLCS as a fuzzy block diagram and developed the connections between the blocks. Next, they presented a mathematical framework to obtain sufficient stability conditions for these systems using Lyapunov's direct method. The stability criteria derived in this paper were presented as linear matrix inequalities (LMIs). Subsequently, utilizing the developed stability analysis methodology, a new technique for designing stable fuzzy controllers was presented. This work laid the foundation for the design of stable T1 TSK FLCs. However, no systematic method was introduced to ensure that the LMIs derived in the presented work are satisfied.

Wang et al. [52] considered satisfying the LMIs derived in [51] using numerical methods. They converted the inequalities into standard LMIs format solvable using conventional techniques [53]. Satisfying the derived LMIs in [51] using numerical techniques has also been addressed in [54]. In addition, research was conducted to systematically design stable T1 TSK FLSs. Joh et al. [55] attempted to tackle this by presenting a systematic method to satisfy the LMIs for stability of T1 TSK FLCSs. They showed that for a TSK system consisting of N subsystems that are pairwise commutative, the proposed algorithm is able to find a common positive definite matrix that satisfies the inequalities. Furthermore, it was shown that the common positive definite matrix can tolerate perturbations. They further generalized their method and investigated the robustness issue under uncertainty in each subsystem. Moreover, some researchers have investigated the stability of T1 TSK FLC of a nonlinear plant subject to parameter uncertainties [56]. They presented relaxed stability conditions for this class of fuzzy control systems, guaranteeing the system's stability.

With the development of T2 FLSs and their ability to handle uncertainty, utilizing T2 FLCSs has attracted a lot of interest in recent years [57, 58, 59, 60, 61]. Although to-date mostly IT2 FLSs have been applied to control applications, promising results have been reported, e.g., Wu and Tan [62] designed an IT2 FLCS for a coupled-tank liquid-level system and showed that, when the level of uncertainty increases, the IT2 FLCS outperforms its T1 counterpart. Additionally, Hagras [63] applied IT2 FLC to mobile robot navigation in dynamic unstructured indoor and outdoor environments. All the IT2 FLCSs implemented in [63] used much smaller rule bases than their T1 counterparts, and it was concluded that IT2 FLCSs provide a faster computation platform as well as enhanced performance results than T1 FLCSs.

The use of intelligent controllers, especially FLCs, for mobile robot applications has been attempted by numerous researchers. T1 FLCSs have been implemented successfully for these applications, and promising results achieved [64, 65, 66]. As well, the application of IT2 FLCs in mobile robots is increasing in the literature [67, 42, 63, 68, 69, 70]. Hagras [63] showed that “The T2-based control system when dealing with the uncertainties facing mobile robots in unstructured environments results in good performance”. The use of hybrid intelligent systems was attempted by Wagner and Hagras [68] who used a genetic algorithm to design T2 FLCs. The developed T2 FLC in this work showed improved performance over a T1 FLC. However, neither work [63] and [68] presents the design steps of the tested IT2 FLCSs. Furthermore, the stability of the developed systems has not been considered in the design process, and nor has system robustness.

The stability of IT2 FLCSs has been a topic of interest to investigators in the last few years. Castillo et al. [70] presented a methodology for the design of stable IT2 FLCs for robot manipulators. Even though the proposed approach in this work is interesting, it is limited to a one-degree-of freedom robot manipulator, and it is therefore, unknown how the proposed approach can be adopted for other nonlinear systems to guarantee stability. In another attempt, Lam and Seneviratne in [71] investigated stability analysis of IT2 TS FLCSs. Their approach requires several assumptions to be made about the MFs in order to enable the derivation of stability conditions, which makes the approach applicable only in specific situations. Additionally, no systematic method is introduced to identify the MF parameters required to satisfy the inequalities defined by those assumptions. Moreover, their model structure produces LMIs that can not be easily simplified or evaluated by anyone wishing to examine the existence of stability criteria.

In recent years, interest has grown in the analysis and implication of IT2 FLCSs because of the increased popularity of such systems [72, 73, 74, 75, 76, 77, 78, 79, 40, 80]. Zhou et al. [78] developed a novel IT2 adaptive FLC for the control of uncertain systems. Simulation results on an inverted pendulum demonstrated that the proposed controller is stable and achieves satisfactory tracking performances. Jammeh et al. [40] developed an IT2 FLC for encoded video streaming of films and demonstrated the enhanced improvement of the IT2 FLC over T1 and other traditional congestion controllers in the presence of uncertainties. The IT2 FLC also reduced packet loss. The authors find the proposed IT2 FLC an efficient controller for congestion control of video streaming. In the most recent work, Du and Ying [81] derived and analyzed the analytical structures of proportional-derivative (PD) and proportional-integral (PI)-like IT2 FLCs.

Because of the complicated nature of the TR method and nonlinearities associated with the controller structure, very limited research has been conducted on systematic T2 fuzzy control designs and, hence, control designers have so far not generally adopted these systems widely. Moreover, most of the existing T2 FLC design methodologies do not even guarantee the stability of their closed-loop system. Instead, control design is accomplished through simulations or trial and error parameters tuning. As a result, this thesis focuses on addressing the stability concern that will eventually facilitate the utility of IT2 FLCs.

T1 TSK models were extensively used for control applications, and demonstrated the usefulness of their structures: consequently the author is motivated to exploit the extended structure of T1 for T2 FLSs and derive the stability criteria needed for IT2 FLC design.

Since type-reduction algorithms are all iterative and have no closed-form formula for I/O relationships, it is extremely difficult to prove the stability or at least obtain sufficient stability conditions. The only available method that has a closed-form to infer the output rules from T2 FLS is WM UBs that estimates the left and right boundaries of IT2 FSs [24]. Therefore, in this research, WM UBs are exploited to develop stability analysis for IT2 FLCSs.

In summary, the majority of the implemented IT2 FLCSs in the literature lack stability analysis in their design. Moreover, the aforementioned T2 FLCs used iterative KM algorithms for TR, and hence, for fast control processes, the applicability of the developed systems is in doubt.

1.3.2 Modular and Reconfigurable Robot (MRR)

Robot manipulators are being increasingly used in various industrial applications such as pick-and-place, assembly, power-assisting, and welding tasks [82, 83, 84]. Although fixed-configuration robots are still common in industry, they are capable of performing only specific tasks, hence limiting their utility. These limitations are, but not restricted to, workspace limits, singularities, etc. In order to perform multiple tasks to address a growing demand from the automation industry, flexible manipulators have been proposed. Modular and reconfigurable robots (MRR), considered to be flexible manipulators, are a viable solution for performing a multitude of desired tasks that cannot be handled using fixed-configuration robots [85, 86, 87]. An MRR system is defined as a collection of individual link and joint components that can be easily assembled into a variety of configurations and geometries [86]. More specifically, MRR not only refers to the entire manipulator, including the modular mechanical hardware, but also to modular electrical hardware, control algorithms, and software [88]. Furthermore, short product life cycles and increasing demand for highly diversified products require innovative design of adaptable manufacturing systems. Today's manufacturers must flexibly adapt production plants to both changing products and methods of production. The use of MRRs for such systems will lead to the significant facilitation of repair and maintenance. Moreover, MRRs will benefit the automation and manufacturing industries by reducing labor and increasing throughput.

MRRs are categorized into three groups: first, modular and reconfigurable serial robots, second, modular and reconfigurable parallel robots, and finally modular self-reconfigurable robots. The most common types of robots in industry are serial manipulators (sometimes called serial robots), which is the category that this paper focuses on.

In the literature, several controllers with different structures have been proposed for robotic systems [89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]. In [97] authors proposed a computationally efficient robust fuzzy control scheme based on the Takagi-Sugeno model. However, the effectiveness of the developed controller was only verified through simulations on a two-link robot manipulator. Most recently, Mostefai et al. [99] used fuzzy modeling approach to compensate for the nonlinear friction of a robot joint structure. The linearity of the local model facilitates the design and the effectiveness of the approach was shown through experiments for trajectory tracking. However, it is an open question if the proposed method can be applied to other complex systems driven by friction. Moreover, this work was applied to a robot with single arm which limits the utility of the method for serial industrial robots or MRRs with multiple degrees of freedom that exhibit dynamic coupling.

In addition, extensive research has been conducted on applying neural networks as well as fuzzy neural networks to control these systems [100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 94, 112, 113]. Due to the universal approximation property of neural networks, Lewis and his team developed a hybrid controller for robotic arms [104]. The neural network in the control structure is used to handle the uncertainties and unknown dynamics of the system. However, only simulations were carried out to demonstrate the performance of the hybrid controller. In recent years, due to the advancements in the implementation of fast algorithms, neural networks have been applied to control robotic systems [106] and [109]. However, to obtain the optimum number of neurons in the input and hidden layers of neural networks, several time consuming experiments must be conducted.

The challenging aspects of controlling MRRs lie in uncertainties in dynamic parameters due to arm reconfigurability. Friction, varying payloads, and dynamic couplings among the joints also contribute to these challenges. There exists very limited research dedicated to control of MRRs. In [114] a control strategy was proposed to handle MRRs as a group of one degree-of-freedom defective joints. This approach does not consider dynamic interactions amongst joints and their variability under reconfigurability. In another attempt, Paredis et al. [115] developed a control software based on the assignment of configuration-dependent parameters. This approach requires a large number of configurations that have to be realized for MRRs. Liu et al. [116] developed a configuration-independent position controller for MRRs based on a joint torque sensing approach. In this method, an MRR is stabilized joint by joint, and modules can be added or removed without the need to adjust the control parameters. However, this approach requires expensive torque sensors to be installed at every joint. Moreover, the approach was not validated experimentally for varying MRR configurations.

Because researchers have to deal with robotic arms that have dynamic parameter uncertainties, there has been an increasing interest in using T1 FLCs for such systems [117, 118]. For example, a stable adaptive fuzzy-based tracking controller with parameter uncertainties and external disturbances was developed for serial robots in [117]. The proposed controller results in a stable system with guaranteed trajectory tracking performances; however, only simulations were performed to validate the performance of the control scheme for a manipulator with two links. In an another study, Chatterjee et al. [118] proposed a stable state-feedback fuzzy control strategy for a flexible robotic arm. The controller structure is based on a neuro-fuzzy model that requires training using experimental data. The authors proved the stability of the system mathematically and verified the success of the design methodology through experiments with several payload conditions. Nevertheless, the system considered had only one flexible arm and dealt with a simple dynamic that exhibited no motion coupling – a common challenge for most serial robots. Furthermore, hybrid fuzzy logic systems (FLS) and neural networks have shown promising results in handling unmodeled dynamics/disturbances. Melek and Goldenberg [119] proposed a hybrid neuro-fuzzy controller that can be used in the presence of parameter uncertainties and unmodeled disturbances. The proposed structure uses fuzzy logic for tuning the PID parameters of the controller. The developed controller was implemented on an MRR with four joints, and its performance was compared to a saturated-type controller. Although results showed significant improvements due to the existence of neural networks in the control architecture, implementing this control methodology is challenging due to the larger number of control parameters that need to be defined a priori.

Interval IT2 FLCs have been shown in several case studies to handle uncertainties better than their T1 counterparts [42, 120, 1, 6]. Furthermore, they have successfully been implemented in real-time control applications, and notable results have been reported in [6] and [121]. However, one of the main issues in adopting such systems on a larger scale is lack of a systematic design methodology, largely because of the complex structure of IT2 fuzzy logic control systems (FLCSs), for which several parameters need to be designed. Implementation of IT2 FLC for control of robotic systems is very limited in the literature, and the existing handful of approaches lacks rigorous mathematical proof of closed-loop system stability. As an example, an IT2 FLC based on the sliding mode control strategy was proposed in [76]. The controller was implemented on a single arm and was proven to outperform its T1 counterpart; however, due to the simplicity of the considered system, the dynamic coupling effect, a major challenge in handling MRRs, does not exist in the authors' work. It is an open question whether this controller can perform better than traditional controllers such as Proportional-Derivative (PD). Moreover, no systematic design methodology was provided to demonstrate how this work can be extended to robots with more than one degree of freedom while still ensuring that stability is guaranteed.

To the best of the author's knowledge, no prior work has been published that provides a systematic method for the design of IT2 TSK FLCs for MRRs. Therefore, in this thesis, a rigorous design paradigm of novel IT2 TSK FLCs for MRRs is studied, the performance of the proposed control strategy is experimentally validated, and finally the performance of the proposed control methodology is compared with some well-known controllers.

1.3.3 Robustness

A great deal of research has been dedicated with success to effectively design FLSs that can accurately capture the model/dynamics of systems [33], [1], [122]. To accurately determine the dynamics of the considered system, it is desired to identify the system as precisely as possible with maximum possible robustness to the given data point.

In most of the existing FLSs, robustness is not considered in the design process. Therefore, it is of great importance to determine the sensitivity or robustness of a FLS to its parameter variations [123, 124]. Robustness analysis in this sense is more appealing for practical applications in which uncertainty, noise, disturbance, etc, not always considered in the design, are present.

Furthermore, robustness of fuzzy mechanisms is one of the major topics in the design of IT2 FLSs. Robustness is defined as the maximum deviation of the output as a result of the deviation of the inputs [119]. In [119], a parameterized formulation of the fuzzy reasoning process was introduced. This parameterized formulation has a closed form and it can be exploited to investigate the robustness characteristics of the fuzzy inference mechanism. Melek and Goldenberg [119] formulated the robustness problem by introducing several parameters into the fuzzy reasoning. By defining the bounds on inference parameters, they obtained maximum input deviation without reaching the maximum desired output. Although authors have mathematically investigated the problem of robustness, certain assumptions have been made on the MFs and, hence, it is uncertain whether the results can be applied in more general cases. Furthermore, the proposed approach is only applicable to FLSs that use Mamdani as their inference engine. Moreover, the approach presented in this paper lacks a systematic methodology for the design of robust T1 FLSs.

Robustness of FLSs with respect to fuzzy operators has been studied thoroughly. Nguyen et al. [125] introduced the robust properties of various fuzzy connectors. He also showed that min and max are the most robust operators. Ying [126] proposed the concepts of maximum and average perturbations of fuzzy sets that led to estimation of perturbation parameters of a fuzzy reasoning. Cai [127] investigated robustness of various operators and inference rules in fuzzy reasoning and discussed how errors in premises affect conclusions. In another study, Li et al. [128] as well as in [129] introduced some measures of robustness of fuzzy operators and discussed their relationships to perturbation. They showed that the robustness of fuzzy reasoning is directly dependent on the fuzzy connectives and implication operators. The measures defined in this work are useful in the context of robustness (or sensitivity) analysis; however, the focus of this work is only on the operators. Most recently, Zheng et al. [130] investigated the robustness of fuzzy system operators for small random perturbations. Their work is based on algebraic operators and proposes two methods to analyzing robustness for random deviations. Similar to the previous publications in the literature, the focus of this paper is also on fuzzy operators. Furthermore, no methodology has been proposed for the design of robust FLS.

Research on the robustness of T1 FLSs is mostly limited to fuzzy operators and, hence, more in-depth analyses must be conducted which will ultimately lead to design of robust systems. Furthermore, this thesis is concerned with a systematic methodology for robustness analysis as well as the design of robust systems that can be practical for modeling and control applications. More importantly, robustness of IT2 FLSs is an important concept that to the best of the author's knowledge has not been studied in the literature.

In most of the existing FLSs, robustness is not considered in the design process. Therefore, it is of great importance to determine the sensitivity of an FLS to input deviations. Specifically, to the best of our knowledge, no research has been conducted to investigate the robustness of T1 TSK and also IT2 FLSs. Therefore, in this work, the robustness of T1 and IT2 TSK FLSs as a function of input data variations is mathematically formulated. We also present algorithms for the design of robust FLSs. Robustness in this sense is an important concept for applications involving I/O mappings such as function approximation, forecasting, modeling, and identification. It is believed that an in-depth analysis of IT2 FLS robustness will help researchers design IT2 FLS structures with maximum robustness for modeling and control applications.

1.4 Organization

The organization of this thesis is as follows:

Chapter 2 establishes preliminaries and backgrounds on T1 TSK and T2 FLSs. It also presents some preliminaries on Linear Algebra used in the control design. First, T1 TSK fuzzy structures are reviewed. Next, T2 FLS structures are visited and existing inference engines are reviewed. Finally, important Linear Algebra results, needed for the control design of MRRs, are presented.

Chapter 3 discusses the stability analysis of IT2 TSK FLSs. We propose new inference engines for the design of IT2 FLSs and present detailed derivation of stability conditions for these systems. We also present algorithms to systematically satisfy the stability criteria followed by several numerical analyses.

Chapter 4 introduces a novel design of IT2 TSK FLC with applications to MRR. An adaptive control scheme is presented for tracking applications of MRRs. Experimental results are provided to demonstrate the effectiveness of the proposed control technique.

Chapter 5 presents a systematic methodology for the robustness analysis and design of robust FLSs (both IT2 and T1). We formulate robustness of FLSs in the context of constraint multi-objective optimization and illustrate the implementation of the proposed approach through several case studies.

Chapter 6 discusses the conclusions and future work of the dissertation.

Chapter 2

Background and Preliminaries

This chapter provides background on T1 TSK FLSs and general T2 FLSs. Preliminaries on Linear Algebra, necessary for control development, are also provided. The organization of this chapter is as follows: First, T1 TSK FLSs are described and necessary facts are established. Next, T2 FLSs are introduced and their properties are presented. The third section is devoted to IT2 TSK FLSs which are extensively used in the subsequent chapters. Finally, in the last section, important Linear Algebra concepts are reviewed.

2.1 T1 TSK FLSs

Consider n inputs $x_1 \epsilon X_1, \cdots, x_n \epsilon X_n$ where $X_1, \cdots, X_n$ are T1 FSs and M outputs y^i, $i = 1, ..., M$ that are crisp numbers. A T1 TSK FLS is described by fuzzy If-Then rules representing I/O relations of the system. In a T1 TSK system with M rules, the ith rule is expressed as

$$i\text{th rule : If } x_1 \text{ is } F_1^i \text{ and } \cdots \text{ and } x_n \text{ is } F_n^i \text{ Then } y^i = a_0^i + a_1^i x_1 + \cdots + a_n^i x_n \tag{2.1}$$

where $i = 1, ..., M$, F_j^i represents the T1 FS of input state j in rule i, $a_0^i, a_1^i, \cdots, a_n^i$ are the coefficients of consequent parameters for rule i. Note that in (2.1) MFs are only associated with rule antecedents. Next, define the input vector as

$$\boldsymbol{x} = [x_1, \cdots, x_n]^T \tag{2.2}$$

The output $y_{\text{T1-TSK}}$ is then obtained by combining the outputs of the M rules as follows:

$$y_{\text{T1-TSK}} = \frac{\sum_{i=1}^{M} f^i(\boldsymbol{x}) y^i(\boldsymbol{x})}{\sum_{i=1}^{M} f^i(\boldsymbol{x})} \tag{2.3}$$

where $f^i(\boldsymbol{x})$ is the firing strength of rule i and is given by

$$f^i(\boldsymbol{x}) = \mu_{F_1^i}(x_1) * \cdots * \mu_{F_n^i}(x_n) \tag{2.4}$$

"$*$" represents a t-norm and $\mu_{F_j^i}$ represents the membership of input x_j in FS F_j^i of rule i. Note that if a discrete system is considered for the purpose of modeling, then the structure of the fuzzy system remains the same but state variables replace the inputs in the above equations.

2.2 T2 Fuzzy Models

This section reviews the main characteristics of T2 FLSs and revisits some important concepts associated with them.

2.2.1 T2 MFs

T2 FLSs are characterized by the shape of their MFs. Figure 2.1 shows two different MFs: (a) a typical T1 MF, (b) a blurred T1 MF that can represent a T2 MF.

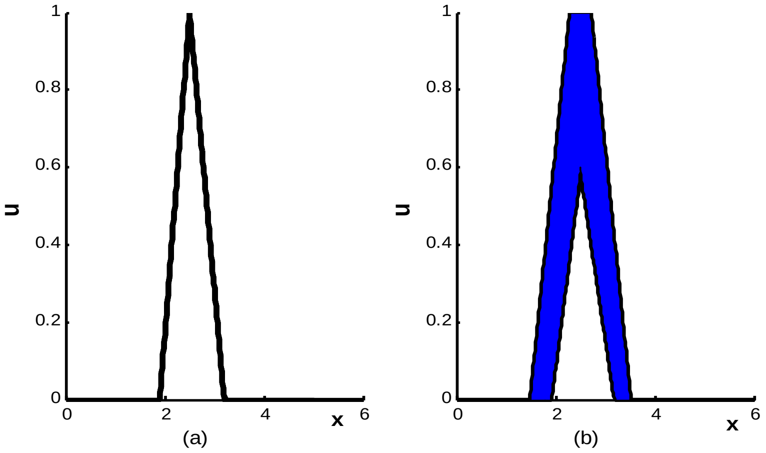


Figure 2.1: (a) T1 MF, (b) Footprint of Uncertainty.

Definition 1 *[1], A T2 FS, denoted as $\widetilde{A}$, is characterized by T2 MF $\mu_{\widetilde{A}}(x,u)$, where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e.,*

$$\widetilde{A} = \left\{ \left((x,u)\,, \mu_{\widetilde{A}}(x,u) \right) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1] \right\} \tag{2.5}$$

which simply shows that a T2 MF is a function of two variables x, and u. It should be mentioned that $\mu_{\widetilde{A}}(x,u)$ is a number between 0 and 1.

$$\widetilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\widetilde{A}}(x,u)/(x,u) \qquad\qquad J_x \subseteq [0,1] \tag{2.6}$$

where $\int\int$denotes union over all x and u.

Definition 2 *[13] When all $\mu_{\widetilde{A}}(x,u) = 1$ then $\widetilde{A}$ is an IT2 FS.*

Definition 3 *[1] The domain of a secondary MF is called the primary membership of x. Therefore, J_x is the primary membership of x. Using this notation, (2.5) can be re-written as*

$$\widetilde{A} = \left\{ \left((x,u)\,, \mu_{\widetilde{A}}(x) \right) \mid \forall x \in X \right\} \tag{2.7}$$

2.2.2 FOU

FOU is one of the major parameters in T2 FLSs, which is frequently used in this thesis. This terminology denotes the uncertainty in the system and also enables us to have a convenient method of description of the entire domain for the secondary MFs.

Definition 4 *[13]: Uncertainty in the primary memberships of a T2 FS consists of a bounded region that is called the FOU, which is the union of all primary memberships, i.e.,*

$$FOU(\widetilde{A}) = \cup_{x \in X} J_x \tag{2.8}$$

Upper and Lower MFs

Definition 5 *[14]: If we bound the FOU of a T2 FS by two T1 MFs, the upper MFs is associated with the upper bound denoted by $\overline{\mu}_{\widetilde{A}}(x), \forall x \in X$ and the lower MF is related to the lower bound denoted by $\underline{\mu}_{\widetilde{A}}(x), \forall x \in X$. In other words,*

$$\begin{aligned} \overline{\mu}_{\widetilde{A}}(x) &= \overline{FOU(\widetilde{A})}, \forall x \in X \\ \underline{\mu}_{\widetilde{A}}(x) &= \underline{FOU(\widetilde{A})}, \forall x \in X \end{aligned} \tag{2.9}$$

Embedded T2 sets

Definition 6 *[1]: For continuous universe of discourse X and U, an embedded T2 set $\widetilde{A}_e$ is:*

$$\widetilde{A}_e = \int_{x \in X} \left[f_x(\theta)/\theta \right] /x \ldots . \theta \in J_x \subseteq [0,1] \tag{2.10}$$

Set $\widetilde{A}_e$ is embedded in set $\widetilde{A}$, and there are infinite number of embedded T2 sets.

2.2.3 T2 Fuzzy Structure

Figure 2.2 shows the structure of a general T2 FLS. This structure is similar to a T1 FLS except that the output processor consists of two operations: type-reducer and defuzzifier. In the remainder of this section, each block of Figure 2.2 is explained.

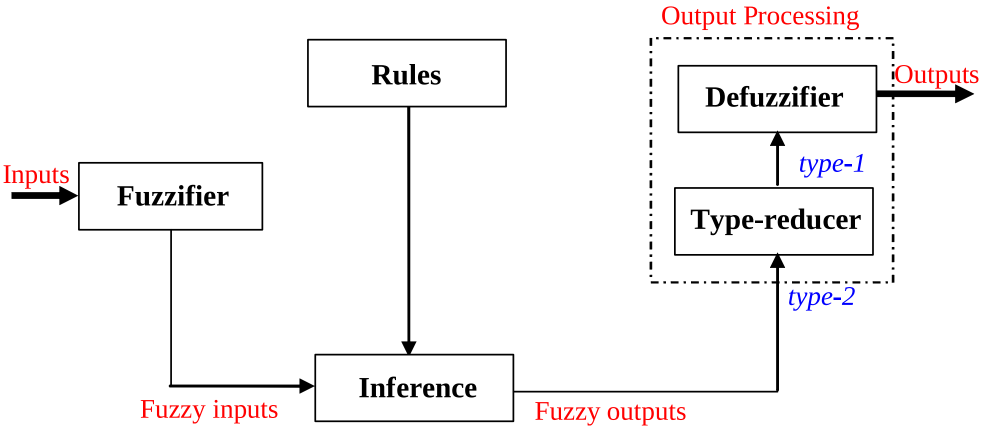


Figure 2.2: Structure of a T2 FLS [1].

Fuzzifier

The first block in Fig. 2.2 is the 'Fuzzifier', where the crisp inputs are fuzzified. The fuzzifier maps the crisp input vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ to a T2 FS $\widetilde{A}_x$, very similar to the procedure performed in a T1 FLS.

Rules

The general form of the ith rule of a T2 FLS can be written as

$$\text{If } x_1 \text{ is } \widetilde{F}_1^i \text{ and } x_2 \text{ is } \widetilde{F}_2^i \text{ and } \cdots \text{ and } x_n \text{ is } \widetilde{F}_n^i, \text{ Then } y^i = \widetilde{G}^i \tag{2.11}$$

where $i = 1, ..., M$, $\widetilde{F}_j^i$ represents the T2 FS of input state j of the ith rule, x_1, x_2,. . ., x_n are inputs, $\widetilde{G}^i$ is the output of T2 FS for rule i, and M is the number of rules. As can be seen, the rule structure of a T2 FLS is very similar to T1 except that T1 MFs are replaced with their T2 counterparts.

Inference Engine

The next block is the inference where reasoning is applied. During this process, using fuzzy logic principles, rules are combined and a mapping is performed from fuzzy input sets to T2 fuzzy output sets.

Output Processing

The output of the inference engine is a T2 FS and must be processed by output processor. Thereafter, the outputs are defuzzified and the corresponding crisp values are found. This process is accomplished through using TR followed by defuzzification.

TR

After the rules are fired and inference is executed, the centroid of the obtained T2 FS resulting in a T1 FS is computed. In this part, the available methods to compute the centroid of a T2 FS using the Extension Principle [1] are discussed. The centroid of a T1 FS, A, is given by

$$C_A = \frac{\sum_{i=1}^{n} z_i w_i}{\sum_{i=1}^{n} w_i} \tag{2.12}$$

where n represents the number of discretized domain of A, $z_i \in R$ and $w_i \in [0, 1]$. If each z_i and w_i are replaced with a T1 FS, Z_i, and W_i, with associated MFs of $\mu_Z(z_i)$ and $\mu_W(w_i)$, respectively, by using the Extension Principle, the generalized centroid for a T2 FS $\widetilde{A}$ is given by

$$GC_{\widetilde{A}} = \int_{z_1 \in Z_1} \cdots \int_{z_n \in Z_n} \int_{w_1 \in W_1} \cdots \int_{w_n \in W_n} \left[T_{i=1}^{n} \mu_Z(z_i) * T_{i=1}^{n} \mu_W(w_i)\right] / \frac{\sum_{i=1}^{n} z_i w_i}{\sum_{i=1}^{n} w_i} \tag{2.13}$$

where T is a t-norm. Note that $GC_{\tilde{A}}$ is a T1 FS. For an IT2 FS

$$GC_{\tilde{A}} = \int_{z_1 \in Z_1} \cdots \int_{z_n \in Z_n} \int_{w_1 \in W_1} \cdots \int_{w_n \in W_n} 1/\frac{\sum_{i=1}^{n} z_i w_i}{\sum_{i=1}^{n} w_i} = [y_l, y_r] \qquad (2.14)$$

KM Algorithms

Since IT2 FLSs are the most common T2 FLSs, the well-known KM algorithms are now presented that are used to find their center. First, express (2.14) as

$$Y_{TR} = [y_l(\boldsymbol{x}), y_r(\boldsymbol{x})] \equiv [y_l, y_r] = \int_{y^1 \in [y_l^1, y_r^1]} \cdots \int_{y^M \in [y_l^M, y_r^M]} \int_{f^1 \in [\underline{f}^1, \overline{f}^1]} \cdots \int_{f^M \in [\underline{f}^M, \overline{f}^M]} 1/\frac{\sum_{i=1}^{M} f^i y^i}{\sum_{i=1}^{M} f^i} \qquad (2.15)$$

The KM algorithms presents iterative procedures to compute y_l, y_r in (2.15) as follows [131]:

To compute y_r:

1. Without loss of generality, assume that y_r^i are arranged in ascending order; i.e., $y_r^1 \leq y_r^2 \leq \cdots \leq y_r^M$.
2. Compute y_r as $y_r = \frac{\sum_{i=1}^{M} f_r^i y_r^i}{\sum_{i=1}^{M} f_r^i}$ by initially setting $f_r^i = \frac{\underline{f}^i + \overline{f}^i}{2}$for $i = 1, \cdots, M$ and let $y_r^{'} \equiv y_r$.
3. Find R $(1 \leq R \leq M - 1)$ such that $y_r^R \leq y_r^{'} \leq y_r^{R+1}$.
4. Compute $y_r = \frac{\sum_{i=1}^{M} f_r^i y_r^i}{\sum_{i=1}^{M} f_r^i}$ with $f_r^i = \underline{f}^i$ for $i \leq R$ and $f_r^i = \overline{f}^i$ for $i > R$ and let $y_r^{''} \equiv y_r$.
5. If $y_r^{''} \neq y_r^{'}$ then go to Step 6. If $y_r^{''} = y_r^{'}$, then stop and set $y_r^{''} \equiv y_r$.
6. Set $y_r^{'}$ equal to $y_r^{''}$, and return to Step 3.

The procedure to compute y_l is very similar to y_r. Just replace y_r^i by y_l^i and in Step 3, find L $(1 \leq L \leq M - 1)$ such that $y_l^L \leq y_l^{'} \leq y_l^{L+1}$. In Step 2, compute y_l as $y_l = \frac{\sum_{i=1}^{M} f_l^i y_l^i}{\sum_{i=1}^{M} f_l^i}$ by initially setting $f_l^i = \frac{\underline{f}^i + \overline{f}^i}{2}$for $i = 1, \cdots, M$ and in Step 4, compute $y_l = \frac{\sum_{i=1}^{M} f_l^i y_l^i}{\sum_{i=1}^{M} f_l^i}$ with $f_l^i = \overline{f}^i$ for $i \leq L$ and $f_l^i = \underline{f}^i$ for $i > L$.

Defuzzifier

To get a crisp output from a T1 FLS, the type-reduced set must be defuzzified. The most common method to do this is to find the centroid of the type-reduced set. If the type-reduced set Y is discretized to m points, then the following expression gives the centroid of the type-reduced set as

$$y_{\text{output}}(\boldsymbol{x}) = \frac{\sum_{i=1}^{m} y^i \mu(y^i)}{\sum_{i=1}^{m} \mu(y^i)} \tag{2.16}$$

2.3 IT2 TSK FLSs

This section deals with IT2 TSK FLSs. The general structure of an interval A2-C0 TSK model for a discrete system[1] is as follows [1]:

If $x(k)$ is $\widetilde{F}_1^i$ and $x(k-1)$ is $\widetilde{F}_2^i$ and $\cdots$ and $x(k-n+1)$ is $\widetilde{F}_n^i$, Then $y_i = a_1^i x(k) + \cdots + a_n^i x(n-k+1)$ (2.17)

where $i = 1, ..., M$, $\widetilde{F}_j^i$ represents the IT2 FS of input state j in rule i, $x(k)$,. . ., $x(k-n+1)$ are states, $a_1^i, \cdots, a_n^i$ are the coefficients of the output function for rule i (and hence are crisp numbers, i.e., type-0 FSs), y_i is the output of the ith rule, and M is the number of rules. The above rules allow us to model the uncertainties encountered in the antecedents. In an IT2 TSK A2-C0 model, $\underline{f}^i$ and $\overline{f}^i$, lower and upper firing strengths of the ith rule, respectively, are given by

$$\underline{f}^i(\boldsymbol{x}) = \underline{\mu}_{\widetilde{F}_1^i}(x(k)) * \cdots * \underline{\mu}_{\widetilde{F}_n^i}(x(k-n+1)) \tag{2.18}$$

$$\overline{f}^i(\boldsymbol{x}) = \overline{\mu}_{\widetilde{F}_1^i}(x(k)) * \cdots * \overline{\mu}_{\widetilde{F}_n^i}(x(k-n+1)) \tag{2.19}$$

where $\underline{\mu}_{\widetilde{F}_i^j}$ and $\overline{\mu}_{\widetilde{F}_j^i}$ represent the jth ($j = 1, \cdots, M$) lower and upper MFs of rule i, and "$*$" is a t-norm operator. State vector is defined as

$$\boldsymbol{x} = [x(k), x(k-1), \cdots, x(k-n+1)]^T \tag{2.20}$$

The final output of the IT2 TSK A2-C0 is given as

$$Y_{\text{TSK/A2-C0}} = [y_l, y_r] = \int_{f^1 \in \left[\underline{f}^1, \overline{f}^1\right]} \cdots \int_{f^M \in \left[\underline{f}^M, \overline{f}^M\right]} 1 \Big/ \frac{\sum_{i=1}^{M} f^i(\boldsymbol{x}) y_i}{\sum_{i=1}^{M} f^i(\boldsymbol{x})} \tag{2.21}$$

where y_i is given by the consequent part of (2.17). $Y_{\text{TSK/A2-C0}}$ is an interval T1 set and only depends on its left and right end-points y_l,y_r, which can be computed using the iterative KM algorithms explained in the previous section. Therefore, the final output is given as

$$Y_{\text{output}}(\boldsymbol{x}) = \frac{y_l(\boldsymbol{x}) + y_r(\boldsymbol{x})}{2} \tag{2.22}$$

[1] This discrete model is used for the stability analysis of IT2 FLCs in Chapter 3.

2.4 Linear Algebra Preliminaries

This section provides preliminaries on the Linear Algebra used to develop IT2 TSK FLCs.

Suppose $\boldsymbol{x}$, $\boldsymbol{r}_1$, $\boldsymbol{r}_2 \in \mathbb{R}^n$, and $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ is a positive semi-definite matrix, then the following properties hold:

P1. $\boldsymbol{r}_1^T \boldsymbol{r}_2 \leq \|\boldsymbol{r}_1\| \, \|\boldsymbol{r}_2\|$.

P2. $\|\boldsymbol{x}\|^2 \min(eig(\boldsymbol{P})) \leq \boldsymbol{x}^T \boldsymbol{P} \boldsymbol{x} \leq \|\boldsymbol{x}\|^2 \max(eig(\boldsymbol{P}))$.

P3. If $\boldsymbol{P}$ is also symmetric, $\|\boldsymbol{P}\|_2 = \max(eig(\boldsymbol{P}))$.

P4. Given two matrices $\boldsymbol{P}_1 \in \mathbb{R}^{n \times m}$ and $\boldsymbol{P}_2 \in \mathbb{R}^{m \times p}$, $\|\boldsymbol{P}_1 \boldsymbol{P}_2\| \leq \|\boldsymbol{P}_1\| \, \|\boldsymbol{P}_2\|$.

P5. For any symmetric matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$, $\boldsymbol{r}_1^T \boldsymbol{Q} \boldsymbol{r}_2 = \boldsymbol{r}_2^T \boldsymbol{Q} \boldsymbol{r}_1$.

P6. For any skew-symmetric matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$, $\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} = 0$.

Chapter 3

Design of IT2 TSK FLCSs for Modeling and Control Applications

This chapter presents the design of IT2 TSK FLSs for modeling and control applications. First, WM UBs are introduced and necessary background are established. Second, new inference engines for IT2 TSK FLSs are introduced. Third, using the most general proposed inference engine, the stability analysis of IT2 FLCSs is presented. Finally, the last section is devoted to the examples.

3.1 WM UBs

Since the iterative KM algorithms can not be expressed in a closed mathematical form, WM UBs [24] are used to obtain the output of an IT2 TSK FLS. For development of IT2 FLCs, the following are key requirements:

1. An analytical methodology is preferred, so as to guarantee a stable control design.
2. The control structure must be suited for real-time implementation.

Therefore, a closed-form I/O inference engine relationship is preferred especially for Lyapunov-based control design. Unfortunately, (2.21) does not provide such a closed-form relationship. Moreover, to satisfy the second requirement, iterative KM inference algorithms may not be suitable. Hence, an alternative approach is considered.

As an alternative to computing $Y_{\text{output}}(\boldsymbol{x})$ using (2.18)-(2.22), WM UBs [24] are used. First, background on WM UBs and their general form as stated in [1] [6] are provided.

[1] In [6], a Mamdani rule is used in which the consequent is an IT2 FS.

3.1.1 Background on WM UBs

WM UBs use the following four centroids (also called *boundary T1 FLSs*):

$$\{\text{LMFs, left}\} \quad : \quad y_l^{(0)}(\boldsymbol{x}) = \frac{\sum_{i=1}^{M} \underline{f}^i\,(\boldsymbol{x})\, y_l^i}{\sum_{i=1}^{M} \underline{f}^i\,(\boldsymbol{x})} \tag{3.1}$$

$$\{\text{LMFs, right}\} \quad : \quad y_r^{(M)}(\boldsymbol{x}) = \frac{\sum_{i=1}^{M} \underline{f}^i\,(\boldsymbol{x})\, y_r^i}{\sum_{i=1}^{M} \underline{f}^i\,(\boldsymbol{x})} \tag{3.2}$$

$$\{\text{UMFs, left}\} \quad : \quad y_l^{(M)}(\boldsymbol{x}) = \frac{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) y_l^i}{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})} \tag{3.3}$$

$$\{\text{UMFs, right}\} \quad : \quad y_r^{(0)}(\boldsymbol{x}) = \frac{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) y_r^i}{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})} \tag{3.4}$$

where y_l^i and y_r^i are the left and right end points of the centroid of the ith consequent IT2 FS and $\underline{f}^i\,(\boldsymbol{x})$ and $\overline{f}^i(\boldsymbol{x})$ are computed using (2.18) and (2.19). The WM UBs are lower and upper bounds for y_l, y_r, and are

$$\begin{aligned} \overline{y}_l(\boldsymbol{x}) &= \min\left\{ y_l^{(0)}(\boldsymbol{x}), y_l^{(M)}(\boldsymbol{x}) \right\} && (3.5) \\ \underline{y}_r(\boldsymbol{x}) &= \max\{ y_r^{(0)}(\boldsymbol{x}), y_r^{(M)}(\boldsymbol{x}) \} && (3.6) \end{aligned}$$

$$\underline{y}_l(\boldsymbol{x}) = \overline{y}_l(\boldsymbol{x}) - \left[\frac{\sum_{i=1}^{M} \left(\overline{f}^i(\boldsymbol{x}) - \underline{f}^i(\boldsymbol{x}) \right)}{\sum_{i=1}^{M} \underline{f}^i\,(\boldsymbol{x}) \,.\, \sum_{i=1}^{M} \overline{f}^i\,(\boldsymbol{x})} \times \frac{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x})\,(y_l^i - y_l^1) \sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \left(y_l^M - y_l^i \right)}{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x})\,(y_l^i - y_l^1) + \sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \left(y_l^M - y_l^i \right)} \right] \tag{3.7}$$

$$\overline{y}_r(\boldsymbol{x}) = \underline{y}_r(\boldsymbol{x}) + \left[\frac{\sum_{i=1}^{M} \left(\overline{f}^i(\boldsymbol{x}) - \underline{f}^i(\boldsymbol{x}) \right)}{\sum_{i=1}^{M} \underline{f}^i\,(\boldsymbol{x}) \,.\, \sum_{i=1}^{M} \overline{f}^i\,(\boldsymbol{x})} \times \frac{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})\,(y_r^i - y_r^1) \,.\, \sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \left(y_r^M - y_r^i \right)}{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})\,(y_r^i - y_r^1) + \sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \left(y_r^M - y_r^i \right)} \right] \tag{3.8}$$

Using the WM UBs the final output of an IT2 FLS, $Y_{\text{WM}}(\boldsymbol{x})$, is computed as [6]

$$Y_{\text{WM}}(\boldsymbol{x}) = \frac{1}{2} \left[\frac{\underline{y}_l(\boldsymbol{x}) + \overline{y}_l(\boldsymbol{x})}{2} + \frac{\underline{y}_r(\boldsymbol{x}) + \overline{y}_r(\boldsymbol{x})}{2} \right] \tag{3.9}$$

Subsequently, the general form of WM UBs is applied to (2.17)-(2.22). Since this chapter deals with IT2 A2-C0 TSK models, $y_l^i = y_r^i = y_i$, the boundary T1 FLSs defined by (3.1)-(3.4) reduce to the following *two* equations:

$$\begin{aligned} y^{(0)}(\boldsymbol{x}) &= \frac{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right) y_i}{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right)} && (3.10) \\ y^{(M)}(\boldsymbol{x}) &= \frac{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x}) y_i}{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})} && (3.11) \end{aligned}$$

Without loss of generality, assume $y^{(M)}(\boldsymbol{x}) > y^{(0)}(\boldsymbol{x})$ ($Y_{\mathrm{WM}}(\boldsymbol{x})$ in (3.9) is invariant to $y^{(M)}(\boldsymbol{x}) > y^{(0)}(\boldsymbol{x})$); therefore, (3.5)-(3.8) can be written as

$$\begin{aligned} \overline{y}_l(\boldsymbol{x}) &= y^{(0)}(\boldsymbol{x}) = \frac{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right) y_i}{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right)} && (3.12) \\ \underline{y}_r(\boldsymbol{x}) &= y^{(M)}(\boldsymbol{x}) = \frac{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x}) y_i}{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})} && (3.13) \end{aligned}$$

$$\begin{aligned} \underline{y}_l(\boldsymbol{x}) = {} & \frac{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right) y_i}{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right)} \\ & - \left[\frac{\sum_{i=1}^{M} \left(\overline{f}^{i}(\boldsymbol{x}) - \underline{f}^{i}(\boldsymbol{x})\right)}{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right) . \sum_{i=1}^{M} \overline{f}^{i}\left(\boldsymbol{x}\right)} \times \frac{\sum_{i=1}^{M} \underline{f}^{i}(\boldsymbol{x})\left(y_i - y_1\right) . \sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})\left(y_M - y_i\right)}{\sum_{i=1}^{M} \underline{f}^{i}(\boldsymbol{x})\left(y_i - y_1\right) + \sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})\left(y_M - y_i\right)} \right] \quad (3.14) \end{aligned}$$

$$\begin{aligned} \overline{y}_r(\boldsymbol{x}) = {} & \frac{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x}) y_i}{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})} \\ & + \left[\frac{\sum_{i=1}^{M} \left(\overline{f}^{i}(\boldsymbol{x}) - \underline{f}^{i}(\boldsymbol{x})\right)}{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right) . \sum_{i=1}^{M} \overline{f}^{i}\left(\boldsymbol{x}\right)} \times \frac{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})\left(y_i - y_1\right) . \sum_{i=1}^{M} \underline{f}^{i}(\boldsymbol{x})\left(y_M - y_i\right)}{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})\left(y_i - y_1\right) + \sum_{i=1}^{M} \underline{f}^{i}(\boldsymbol{x})\left(y_M - y_i\right)} \right] \quad (3.15) \end{aligned}$$

Using (3.12)-(3.15), it is straightforward to show that $Y_{\mathrm{WM}}(\boldsymbol{x})$ in (3.9) can be expressed as

$$\begin{aligned} Y_{\mathrm{WM}}(\boldsymbol{x}) = {} & \frac{1}{2} \left(\frac{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right) y_i}{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right)} + \frac{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x}) y_i}{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})} \right) \\ & - \frac{1}{4} \left[\frac{\sum_{i=1}^{M} \left(\overline{f}^{i}(\boldsymbol{x}) - \underline{f}^{i}(\boldsymbol{x})\right)}{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right) . \sum_{i=1}^{M} \overline{f}^{i}\left(\boldsymbol{x}\right)} \times \frac{\sum_{i=1}^{M} \underline{f}^{i}(\boldsymbol{x})\left(y_i - y_1\right) . \sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})\left(y_M - y_i\right)}{\sum_{i=1}^{M} \underline{f}^{i}(\boldsymbol{x})\left(y_i - y_1\right) + \sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})\left(y_M - y_i\right)} \right] \\ & + \frac{1}{4} \left[\frac{\sum_{i=1}^{M} \left(\overline{f}^{i}(\boldsymbol{x}) - \underline{f}^{i}(\boldsymbol{x})\right)}{\sum_{i=1}^{M} \underline{f}^{i}\left(\boldsymbol{x}\right) . \sum_{i=1}^{M} \overline{f}^{i}\left(\boldsymbol{x}\right)} \times \frac{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})\left(y_i - y_1\right) . \sum_{i=1}^{M} \underline{f}^{i}(\boldsymbol{x})\left(y_M - y_i\right)}{\sum_{i=1}^{M} \overline{f}^{i}(\boldsymbol{x})\left(y_i - y_1\right) + \sum_{i=1}^{M} \underline{f}^{i}(\boldsymbol{x})\left(y_M - y_i\right)} \right] \quad (3.16) \end{aligned}$$

$Y_{\mathrm{WM}}(\boldsymbol{x})$ given by (3.16) represents the final output of the IT2 TSK A2-C0 system (2.17). It is easy to see that $Y_{\mathrm{WM}}(\boldsymbol{x})$ can be computed without having to perform TR and therefore $Y_{\mathrm{WM}}(\boldsymbol{x})$ can be considered a viable alternative to using (2.21) and (2.22) for real-time control.

Now, $Y_{\mathrm{WM}}(\boldsymbol{x})$ is applied to $Y_{\mathrm{TSK/A2\text{-}C0}}(\boldsymbol{x})$ using the following discrete-time model that appears in the consequent of rule i in (2.17):

$$y_i = \sum_{p=1}^{n} a_p^i x(k-p+1) \tag{3.17}$$

It follows that

$$\begin{aligned} y_i - y_1 &= \sum_{p=1}^{n} (a_p^i - a_p^1) x(k-p+1) \equiv \sum_{p=1}^{n} v_{i,p} a_p^i x(k-p+1) && (3.18) \\ y_M - y_i &= \sum_{p=1}^{n} (a_p^M - a_p^i) x(k-p+1) \equiv \sum_{p=1}^{n} w_{i,p} a_p^i x(k-p+1) && (3.19) \end{aligned}$$

where

$$\begin{aligned} v_{i,p} &\equiv \frac{a_p^i - a_p^1}{a_p^i} && (3.20) \\ w_{i,p} &\equiv \frac{a_p^M - a_p^i}{a_p^i} && (3.21) \end{aligned}$$

Substituting (3.17)-(3.19) into (3.16), it is straightforward to show that $Y_{\mathrm{WM}}(\boldsymbol{x})$ can be expressed as

$$\begin{aligned} Y_{\mathrm{WM}}(\boldsymbol{x}) = \frac{1}{2} \frac{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \left(\sum_{p=1}^{n} a_p^i x(k-p+1) \right)}{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x})} + \frac{1}{2} \frac{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \left(\sum_{p=1}^{n} a_p^i x(k-p+1) \right)}{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})} \\ + \alpha(\boldsymbol{x}) + \beta(\boldsymbol{x}) \quad (3.22) \end{aligned}$$

where

$$\begin{aligned} \alpha(\boldsymbol{x}) = -\frac{1}{4} \frac{\sum_{i=1}^{M} (\overline{f}^i(\boldsymbol{x}) - \underline{f}^i(\boldsymbol{x}))}{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \cdot \sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})} \times \\ \frac{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \left(\sum_{p=1}^{n} v_{i,p} a_p^i x(k-p+1) \right) \cdot \sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \left(\sum_{p=1}^{n} w_{i,p} a_p^i x(k-p+1) \right)}{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \left(\sum_{p=1}^{n} v_{i,p} a_p^i x(k-p+1) \right) + \sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \left(\sum_{p=1}^{n} w_{i,p} a_p^i x(k-p+1) \right)} \quad (3.23) \end{aligned}$$

$$\beta(\boldsymbol{x}) = \frac{1}{4}\frac{\sum_{i=1}^{M}(\overline{f}^i(\boldsymbol{x}) - \underline{f}^i(\boldsymbol{x}))}{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x}).\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})}\times$$
$$\frac{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})\left(\sum_{p=1}^{n} v_{i,p}a_p^i x(k-p+1)\right).\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})\left(\sum_{p=1}^{n} w_{i,p}a_p^i x(k-p+1)\right)}{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})\left(\sum_{p=1}^{n} v_{i,p}a_p^i x(k-p+1)\right)+\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})\left(\sum_{p=1}^{n} w_{i,p}a_p^i x(k-p+1)\right)} \tag{3.24}$$

(3.22) has been used recently for control design [68]; however, no information is available to-date on how to systematically design IT2 FLCSs using $Y_{\mathrm{WM}}(\boldsymbol{x})$. Stability analysis for (3.22) was attempted, but no successful results were achieved (due to its nonlinear and complicated structure).

3.2 Proposed Inference Methods for IT2 TSK FLCSs

To obtain stability conditions for a FLCS using rigorous mathematical analyses, closed-form equations are required. However, due to the complexity of WM UBs, use of this method is limited. Hence, in this section, novel inference engines are proposed that are effectively used to replace TR. The new inference mechanisms have simple structures and are therefore more suited for implementation in real-time control applications. These inference engines are as follows:

$$Y_{\mathrm{TSK1}}(\boldsymbol{x}) = \frac{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})y_i}{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})+\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})} + \frac{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})y_i}{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})+\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})} \tag{3.25}$$

$$Y_{\mathrm{TSK2}}(\boldsymbol{x}) = q\frac{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})y_i}{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})} + (1-q)\frac{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})y_i}{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})} \tag{3.26}$$

$$Y_{\mathrm{TSK3}}(\boldsymbol{x}) = q\frac{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})y_i}{q\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})+(1-q)\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})} + (1-q)\frac{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})y_i}{q\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})+(1-q)\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})} \tag{3.27}$$

$$Y_{\mathrm{TSK4}}(\boldsymbol{x}) = \frac{m\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})y_i + n\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})y_i}{m\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x}) + n\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})} \tag{3.28}$$

$$Y_{\mathrm{TSK5}}(\boldsymbol{x}) = m\frac{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})y_i}{\sum_{i=1}^{M}\underline{f}^i(\boldsymbol{x})} + n\frac{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})y_i}{\sum_{i=1}^{M}\overline{f}^i(\boldsymbol{x})} \quad (\text{if } M=1: m+n=1) \tag{3.29}$$

To tune the model variables, parameters p, q, m, and n are introduced in the formulas. The parameters q in (3.26) and (3.27) and m and n in (3.28) and (3.29) are design factors that weight the share of firing level of each fired rule depending on the level of the system uncertainty. The parameter formulas enable designers to effectively optimize the I/O mapping represented by the selected inference methods.

Formula (3.29) is similar to the output of a T1 TSK FLS in which the fired rules are weighted-averaged. However, it differs from it in a sense that both upper and lower MFs contribute to the output of the fuzzy system. Throughout this thesis, (3.29) is adopted as the inference engine since it is more general than other proposed engines and similar to a T1 FLS output. Observe that m and n are design parameters that weight the sharing of lower and upper firing levels of each fired rule and can be tuned during the design of this new TSK system. Observe, also, that if all uncertainty disappears so that $\underline{f}^i(\boldsymbol{x}) = \overline{f}^i(\boldsymbol{x})$, then (3.29) reduces to a T1 TSK FLCS in which one can set $m + n = 1$. There is also a connection between $Y_{\mathrm{TSK5}}(\boldsymbol{x})$ and $Y_{\mathrm{WM}}(\boldsymbol{x})$.

Proposition 1 *If m and n are independent parameters that do not depend on the inference process, then $Y_{TSK5}(\boldsymbol{x})$ is derivable from $Y_{WM}(\boldsymbol{x})$ and is a simplified version of $Y_{WM}(\boldsymbol{x})$.*

Proof. Using (3.23) and (3.24), $\alpha(\boldsymbol{x})$ and $\beta(\boldsymbol{x})$ can be expressed as nonlinear functions of the upper and lower firing levels of each rule as well as the input states, i.e.,

$$\alpha(\boldsymbol{x}) = g_1\left(\underline{f}^i(\boldsymbol{x}), \overline{f}^i(\boldsymbol{x}), \boldsymbol{x}, v_{i,p}, w_{i,p}\right) \cdot \frac{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \left[\sum_{p=1}^{n} a_p^i x(k-p+1)\right]}{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x})} \tag{3.30}$$

$$\beta(\boldsymbol{x}) = g_2\left(\underline{f}^i(\boldsymbol{x}), \overline{f}^i(\boldsymbol{x}), \boldsymbol{x}, v_{i,p}, w_{i,p}\right) \cdot \frac{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \left[\sum_{p=1}^{n} a_p^i x(k-p+1)\right]}{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})} \tag{3.31}$$

where functions $g_1$and g_2 are given by[2]

$$\begin{aligned} g_1 = &-\frac{1}{4} \frac{\sum_{i=1}^{M} \left(\overline{f}^i(\boldsymbol{x}) - \underline{f}^i(\boldsymbol{x})\right)}{\left[\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} a_p^i x(k-p+1)\right] \sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})} \\ &\times \frac{\sum_{i=1}^{M} \left[\underline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} v_{i,p} a_p^i x(k-p+1)\right] \sum_{i=1}^{M} \left[\overline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} w_{i,p} a_p^i x(k-p+1)\right]}{\sum_{i=1}^{M} \left[\underline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} v_{i,p} a_p^i x(k-p+1)\right] + \sum_{i=1}^{M} \left[\overline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} w_{i,p} a_p^i x(k-p+1)\right]} \end{aligned} \tag{3.32}$$

[2] In order to simplify the notation, in the rest of the derivation g_1 and g_2 are short for $g_1\left(\underline{f}^i(\boldsymbol{x}), \overline{f}^i(\boldsymbol{x}), \boldsymbol{x}, v_{i,p}, w_{i,p}\right)$ and $g_2\left(\underline{f}^i(\boldsymbol{x}), \overline{f}^i(\boldsymbol{x}), \boldsymbol{x}, v_{i,p}, w_{i,p}\right)$.

$$
\begin{aligned}
g_2 = \frac{1}{4} & \frac{\sum_{i=1}^{M} \left(\overline{f}^i(\boldsymbol{x}) - \underline{f}^i(\boldsymbol{x}) \right)}{\left[\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} a_p^i x(k-p+1) \right] \sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x})} \\
\times & \frac{\sum_{i=1}^{M} \left[\overline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} v_{i,p} a_p^i x(k-p+1) \right] \sum_{i=1}^{M} \left[\underline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} w_{i,p} a_p^i x(k-p+1) \right]}{\sum_{i=1}^{M} \left[\overline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} v_{i,p} a_p^i x(k-p+1) \right] + \sum_{i=1}^{M} \left[\underline{f}^i(\boldsymbol{x}) \sum_{p=1}^{n} w_{i,p} a_p^i x(k-p+1) \right]}
\end{aligned}
\tag{3.33}
$$

Using (3.30)-(3.33), $Y_{\mathrm{WM}}(\boldsymbol{x})$ in (3.22) can be written as

$$
\begin{aligned}
Y_{\mathrm{WM}}(\boldsymbol{x}) = & \frac{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \left(\frac{1}{2} \sum_{p=1}^{n} a_p^i x(k-p+1) \right)}{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x})} + \frac{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \left(\frac{1}{2} \sum_{p=1}^{n} a_p^i x(k-p+1) \right)}{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})} \\
& + g_1 \times \frac{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \left[\sum_{p=1}^{n} a_p^i x(k-p+1) \right]}{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x})} + g_2 \times \frac{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \left[\sum_{p=1}^{n} a_p^i x(k-p+1) \right]}{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})}
\end{aligned}
\tag{3.34}
$$

Combining the first and the third terms and the second and the fourth terms of $Y_{\mathrm{WM}}(\boldsymbol{x})$, (3.34) can be rewritten as

$$
\begin{aligned}
Y_{\mathrm{WM}}(\boldsymbol{x}) = & \frac{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x}) \left(\sum_{p=1}^{n} \overbrace{\left(\frac{1}{2} + g_1 \right)}^{m} a_p^i x(k-p+1) \right)}{\sum_{i=1}^{M} \underline{f}^i(\boldsymbol{x})} \\
& + \frac{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x}) \left(\sum_{p=1}^{n} \overbrace{\left(\frac{1}{2} + g_2 \right)}^{n} a_p^i x(k-p+1) \right)}{\sum_{i=1}^{M} \overline{f}^i(\boldsymbol{x})}
\end{aligned}
\tag{3.35}
$$

Comparing (3.35) and (3.29), it can be seen that m and n correspond to $\left(\frac{1}{2} + g_1\right)$ and $\left(\frac{1}{2} + g_2\right)$, respectively, i.e.,

$$
m = \frac{1}{2} + g_1 \tag{3.36}
$$

$$
n = \frac{1}{2} + g_2 \tag{3.37}
$$

where g_1 and g_2 are given by (3.32) and (3.33), respectively. Under the assumption that m and n are adjustable parameters that do not depend on the inference process, $Y_{\mathrm{WM}}(\boldsymbol{x})$ simplifies to $Y_{\mathrm{TSK5}}(\boldsymbol{x})$.

When (3.29) is used to model the plant, a procedure to obtain the TSK consequent parameters as well as the tuning parameters, m and n, is given next. First, bounds for the tuning parameters of the plant are derived; then, for a given m and n, it is mathematically explained how to identify the IT2 TSK consequent parameters, and finally, an algorithm is presented to obtain suitable tuning parameters for the plant.

Since this chapter deals with stability analysis, it is assumed the parameters of the MFs are known [identifying the MFs is not within the scope of this thesis]. When IT2 TSK is used for practical control design, states $x(k), x(k-1), \cdots, x(k-n+1)$ are physical quantities, e.g., displacement, velocity, acceleration; therefore, for a specific problem, the lower and upper bounds of these states can be determined by the designer. Moreover, $y_i = \sum_{p=1}^{n} a_p^i x(k-p+1)$ corresponds to the output of rule i and represents a physical quantity [similar arguments can be made for the two terms $\sum_{p=1}^{n} v_{i,p} a_p^i x(k-p+1)$ and $\sum_{p=1}^{n} w_{i,p} a_p^i x(k-p+1)$ in g_1 and g_2]; hence, regardless of whether a_p^i is known or not, the designer can establish the lower and upper bounds on y_i as well as another similar terms in g_1 and g_2 (the range of variation is known). The lower and upper bounds on g_1 and g_2 for all rules can therefore be determined, i.e.,

$$g_1^{\min} \leq g_1 \leq g_1^{\max} \tag{3.38}$$

$$g_2^{\min} \leq g_2 \leq g_2^{\max} \tag{3.39}$$

Then, using (3.36) and (3.37) in (3.38) and (3.39), the bounds on m and n are given as

$$m^{\min} \equiv \frac{1}{2} + g_1^{\min} \leq m \leq \frac{1}{2} + g_1^{\max} \equiv m^{\max} \tag{3.40}$$

$$n^{\min} \equiv \frac{1}{2} + g_2^{\min} \leq n \leq \frac{1}{2} + g_2^{\max} \equiv n^{\max} \tag{3.41}$$

Recently, fuzzy clustering and subtractive clustering have been proposed to find the parameters of T2 FLSs. Subsequently, similar to [132], in the method presented in this chapter, it is assumed that the parameters of the input MFs are known by using a predefined clustering method, and the TSK consequent parameters are identified. Assume the plant is modeled as

$$\text{If } x_1 \text{ is } \widetilde{F}_1^i \text{ and } x_2 \text{ is } \widetilde{F}_2^i \text{ and } \cdots \text{ and } x_n \text{ is } \widetilde{F}_n^i, \text{ Then } y_i = a_0^i + a_1^i x_1 + a_2^i x_2 + \cdots + a_n^i x_n \tag{3.42}$$

where $i = 1, ..., M$. Suppose p input-output data (training data) for the plant are given as

$$\left\{ \left[x_1^i, x_2^i, \cdots, x_n^i\right], Y^i \right\}_{i=1}^{p} \tag{3.43}$$

where $[x_1^i, x_2^i, \cdots, x_n^i]$ is the ith input vector consisting of n inputs, and Y^i is the corresponding output. Define $\boldsymbol{Y} \in \mathbb{R}^p$ containing the training outputs

$$\boldsymbol{Y} \equiv \left[Y^1, Y^2, \cdots, Y^p\right]^T \tag{3.44}$$

Using (3.29) and applying the method described in [133], Y^i can be expressed as

$$Y^i = m\frac{\sum_{j=1}^{M} \underline{f}_i^j (a_0^j + a_1^j x_1^i + a_2^j x_2^i + ... + a_n^j x_n^i)}{\sum_{j=1}^{M} \underline{f}_i^j} \\ + n\frac{\sum_{i=1}^{M} \overline{f}_i^j (a_0^j + a_1^j x_1^i + a_2^j x_2^i + ... + a_n^j x_n^i)}{\sum_{j=1}^{M} \overline{f}_i^j} \tag{3.45}$$

where $i = 1, \cdots, p$. Let

$$\underline{v}_i^j = \frac{\underline{f}_i^j}{\sum_{j=1}^{M} \underline{f}_i^j} \tag{3.46}$$

$$\overline{v_i}^j = \frac{\overline{f_i}^j}{\sum_{j=1}^{M} \overline{f_i}^j} \tag{3.47}$$

Using (3.46) and (3.47), (3.45) can be rewritten as

$$Y^i = m\sum_{j=1}^{M} \underline{v}_i^j (a_0^j + a_1^j x_1^i + a_2^j x_2^i + ... + a_n^j x_n^i) \\ + n\sum_{j=1}^{M} \overline{v_i}^j (a_0^j + a_1^j x_1^i + a_2^j x_2^i + ... + a_n^j x_n^i) \tag{3.48}$$

Define $\underline{\boldsymbol{\phi}}$ as

$$\underline{\boldsymbol{\phi}} \equiv \left[\begin{array}{cccccccccc} \underline{v}_1^1 & \cdots & \underline{v}_1^M & \underline{v}_1^1 x_1^1 & \cdots & \underline{v}_1^M x_1^1 & & \underline{v}_1^1 x_n^1 & \cdots & \underline{v}_1^M x_n^1 \\ \underline{v}_2^1 & \cdots & \underline{v}_2^M & \underline{v}_2^1 x_1^2 & \cdots & \underline{v}_2^M x_1^2 & \cdots\cdots & \underline{v}_2^1 x_n^2 & \cdots & \underline{v}_2^M x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \underline{v}_p^1 & \cdots & \underline{v}_p^M & \underline{v}_p^1 x_1^p & \cdots & \underline{v}_p^M x_1^p & & \underline{v}_p^1 x_n^p & \cdots & \underline{v}_p^M x_n^p \end{array}\right] \tag{3.49}$$

Define $\overline{\boldsymbol{\phi}}$ and $\boldsymbol{\theta}$

$$\overline{\boldsymbol{\phi}} \equiv \left[\begin{array}{cccccccccc} \overline{v}_1^1 & \cdots & \overline{v}_1^M & \overline{v}_1^1 x_1^1 & \cdots & \overline{v}_1^M x_1^1 & & \overline{v}_1^1 x_n^1 & \cdots & \overline{v}_1^M x_n^1 \\ \overline{v}_2^1 & \cdots & \overline{v}_2^M & \overline{v}_2^1 x_1^2 & \cdots & \overline{v}_2^M x_1^2 & \cdots\cdots & \overline{v}_2^1 x_n^2 & \cdots & \overline{v}_2^M x_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \overline{v}_p^1 & \cdots & \overline{v}_p^M & \overline{v}_p^1 x_1^p & \cdots & \overline{v}_p^M x_1^p & & \overline{v}_p^1 x_n^p & \cdots & \overline{v}_p^M x_n^p \end{array}\right] \tag{3.50}$$

$$\boldsymbol{\theta} \equiv \left[\begin{array}{ccc} a_0^1, \cdots, a_0^M & , a_1^1, \cdots, a_1^M & \cdots & , a_n^1, \cdots, a_n^M \end{array}\right]^T \tag{3.51}$$

Using (3.49) and (3.50), $\boldsymbol{Y}$ can be expressed as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\theta} \tag{3.52}$$

where $\boldsymbol{A} = m\underline{\boldsymbol{\phi}} + n\overline{\boldsymbol{\phi}}$ is a known matrix consisting of the parameters of the input MFs.

Finally, the error vector is defined as $\boldsymbol{e} \equiv \boldsymbol{Y} - \boldsymbol{A}\boldsymbol{\theta}$, and the total error, e_t, which is the sum of squares of the components of $\boldsymbol{e}$ is defined by

$$e_t \equiv \sum_{i=1}^{p} \boldsymbol{e}_i^2 \tag{3.53}$$

To obtain the tuning parameters, m and n, of the plant inference engine, the following algorithm is proposed:

Algorithm 1 Finding the plant tuning parameters.

```
$m \leftarrow m^{min}$ and $n \leftarrow n^{min}$, and calculate the initial error using (3.44)-(3.53)
repeat
  repeat
    $n \leftarrow n^{min}$
    Solve for $\boldsymbol{\theta}$ from (3.52), and find the total error from (3.53)
    the new error is less than the error found in the previous step, save $m$, $n$, and $\boldsymbol{\theta}$
    Increment $n$, i.e., $n \leftarrow n + \Delta n$ (where $\Delta n = 0.05n$)
  until $n \leq n^{max}$
  $m \leftarrow m + \Delta m$ (where $\Delta m = 0.05m$)
until $m \leq m^{max}$
```

Note that in the proposed algorithm, the increments for $\frac{\Delta m}{m}$ and $\frac{\Delta n}{n}$ are 0.05. Depending on the required accuracy for identification of the plant or computational complexity in performing the algorithm, smaller or greater increments can be used.

3.3 Stability of SISO IT2 TSK FLCSs

In this section, a model for the stability analysis of single-input single-output is introduced (SISO)[3] IT2 TSK FLCS. SISO systems are considered because of the variety of applications in computing systems and bioengineering ([134], [135]). To begin, a controller structure is introduced; then, a model is introduced for a closed-loop control system, after which mathematical analyses are established for the design of stable IT2 TSK FLCSs.

[3] Referring to SISO systems, 'input' is considered the controller output signal and 'output' is the plant output (both input and output being scalars).

3.3.1 Controller

Figure 3.1 shows a controller in which the inputs are the states, $\boldsymbol{x}(k)$, and the output is $u(k)$.

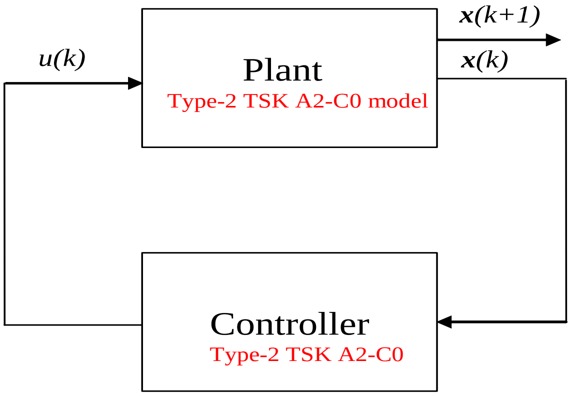


Figure 3.1: Closed-loop IT2 TSK A2-C0 FLCS.

For this system, the general ith rule has the following form:

$$\begin{aligned} i\text{th controller rule : If } x(k) \text{ is } \widetilde{C}_1^i \text{ and } x(k-1) \text{ is } \widetilde{C}_2^i \text{ and } \cdots \text{ and } x(k-n+1) \text{ is } \widetilde{C}_n^i \\ \text{Then } u^i(k+1) = c_1^i x(k) + c_2^i x(k-1) + \cdots + c_n^i x(k-n+1) \end{aligned} \tag{3.54}$$

where $i = 1, 2, ..., Q$, $\widetilde{C}_j^i$ represents the T2 FS of input state j of the ith rule, and c_j^i is the jth coefficient of the output function for rule i. Applying (3.29) to (3.54), the controller output, $u(k)$, can be expressed as

$$u(k) = m' \frac{\sum_{i=1}^{Q} \underline{v}^i(\boldsymbol{x}) u^i(k+1)}{\sum_{i=1}^{Q} \underline{v}^i(\boldsymbol{x})} + n' \frac{\sum_{i=1}^{Q} \overline{v}^i(\boldsymbol{x}) u^i(k+1)}{\sum_{i=1}^{Q} \overline{v}^i(\boldsymbol{x})} \tag{3.55}$$

where

$$\underline{v}^i(\boldsymbol{x}) = \underline{\mu}_{\widetilde{C}_1^i}(x(k)) * \cdots * \underline{\mu}_{\widetilde{C}_n^i}(x(k-n+1)) \tag{3.56}$$

$$\overline{v}^i(\boldsymbol{x}) = \overline{\mu}_{\widetilde{C}_1^i}(x(k)) * \cdots * \overline{\mu}_{\widetilde{C}_n^i}(x(k-n+1)) \tag{3.57}$$

and m' and n' are tuning parameters of the controller. Substituting the consequent part of (3.54) into (3.55), (3.55) can be written as

$$u(k) = m' \frac{\sum_{i=1}^{Q} \sum_{j=1}^{n} \underline{v}^i(\boldsymbol{x}) c_j^i x(k-j+1)}{k_1} + n' \frac{\sum_{i=1}^{Q} \sum_{j=1}^{n} \overline{v}^i(\boldsymbol{x}) c_j^i x(k-j+1)}{k_2} \tag{3.58}$$

where

$$\begin{cases} k_1 \equiv \sum_{i=1}^{Q} \underline{v}^i(\boldsymbol{x}) \\ k_2 \equiv \sum_{i=1}^{Q} \overline{v}^i(\boldsymbol{x}) \end{cases} \tag{3.59}$$

Note that parameters k_1 and k_2 are short for $k_1(\boldsymbol{x})$ and $k_2(\boldsymbol{x})$, respectively.

3.3.2 Closed-loop System

Consider the feedback control system shown in Figure 3.1, where the plant and the controller are each IT2 TSK A2-C0 models. For a closed-loop system, the controller signal, $u(k)$, is incorporated as an input to the plant. The general ith rule for the plant is

$$\begin{aligned} &\text{If } x(k) \text{ is } \widetilde{F}_1^i \text{ and } x(k-1) \text{ is } \widetilde{F}_2^i \text{ and } \cdots \text{ and } x(k-n+1) \text{ is } \widetilde{F}_n^i \text{ and } u(k) \text{ is } \widetilde{B}^i \\ &\qquad\qquad \text{Then } x^i(k+1) = a_1^i x(k) + \cdots + a_n^i x(n-k+1) + b^i u(k) \end{aligned} \tag{3.60}$$

where $i = 1, ..., M$, $x^i(k+1)$ is the output of the ith plant rule, $\widetilde{F}_j^i$ represents the T2 FS of input state j of the ith rule, $\widetilde{B}^i$ represents the T2 FS of the plant input, and a_j^i is the jth coefficient of the output function for rule i. The control rules are the same as (3.54) with Q being the number of rules, and $u(k)$ is given by (3.58). Substituting $u(k)$ from (3.58) into the consequent of (3.60), the output of the ith plant rule, $x^i(k+1)$, is given by

$$x^i(k+1) = \sum_{j=1}^{n} \left[a_j^i x(k-j+1)\right] + b^i m' \frac{\sum_{l=1}^{Q} \sum_{j=1}^{n} \underline{v}^l c_j^l x(k-j+1)}{k_1} + b^i n' \frac{\sum_{l=1}^{Q} \sum_{j=1}^{n} \overline{v}^l c_j^l x(k-j+1)}{k_2} \tag{3.61}$$

Using (3.29), the output of the closed-loop system, $x(k+1)$, can be expressed as

$$x(k+1) = \frac{m \sum_{i=1}^{M} \underline{f}^i x^i(k+1)}{\sum_{i=1}^{M} \underline{f}^i} + \frac{n \sum_{i=1}^{M} \overline{f}^i x^i(k+1)}{\sum_{i=1}^{M} \overline{f}^i} \tag{3.62}$$

where m and n are the tuning parameters of the plant and $\underline{f}^i$ and $\overline{f}^i$ [short for $\underline{f}^i(\boldsymbol{x})$ and $\overline{f}^i(\boldsymbol{x})$] are given by

$$\underline{f}^i(\boldsymbol{x}) = \underline{\mu}_{\widetilde{F}_1^i}(x(k)) * \cdots * \underline{\mu}_{\widetilde{F}_n^i}(x(k-n+1)) * \underline{\mu}_{\widetilde{B}^i}(u(k)) \tag{3.63}$$

$$\overline{f}^i(\boldsymbol{x}) = \overline{\mu}_{\widetilde{F}_1^i}(x(k)) * \cdots * \overline{\mu}_{\widetilde{F}_n^i}(x(k-n+1)) * \overline{\mu}_{\widetilde{B}^i}(u(k)) \tag{3.64}$$

Applying (3.61) to (3.62), $x(k+1)$ can be expressed as

$$\begin{aligned} x(k+1) = {} & \frac{m\sum_{i=1}^{M}\sum_{j=1}^{n}\underline{f}^i a_j^i x(k-j+1)}{\sum_{i=1}^{M}\underline{f}^i} + mm'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\sum_{j=1}^{n}\underline{f}^i\underline{v}^l b^i c_j^l x(k-j+1)}{k_1\sum_{i=1}^{M}\underline{f}^i} \\ & + mn'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\sum_{j=1}^{n}\underline{f}^i\overline{v}^l b^i c_j^l x(k-j+1)}{k_1\sum_{i=1}^{M}\underline{f}^i} + \frac{n\sum_{i=1}^{M}\sum_{j=1}^{n}\overline{f}^i a_j^i x(k-j+1)}{\sum_{i=1}^{M}\overline{f}^i} \\ & + nm'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\sum_{j=1}^{n}\overline{f}^i\underline{v}^l b^i c_j^l x(k-j+1)}{k_2\sum_{i=1}^{M}\overline{f}^i} + nn'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\sum_{j=1}^{n}\overline{f}^i\overline{v}^l b^i c_j^l x(k-j+1)}{k_2\sum_{i=1}^{M}\overline{f}^i} \end{aligned} \tag{3.65}$$

Next, define $n \times n$ matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_{i,l}$ as follows:

$$\boldsymbol{A}_i = \begin{pmatrix} a_1^i & a_2^i & \dots & a_{n-1}^i & a_n^i \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}, \boldsymbol{B}_{i,l} = \begin{pmatrix} b^i c_1^l & b^i c_2^l & \dots & b^i c_{n-1}^l & b^i c_n^l \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} \tag{3.66}$$

where $i = 1, 2, ..., M$ and $l = 1, 2, ..., Q$. Define the output vector as

$$\boldsymbol{x}(k+1) = [x(k+1), x(k), \cdots, x(k-n+2)]^T \tag{3.67}$$

Using (3.65) and (3.66), it is straightforward to show that $\boldsymbol{x}(k+1)$ in (3.67) can be written as

$$\boldsymbol{x}(k+1) = \boldsymbol{C}\boldsymbol{x}(k) \tag{3.68}$$

where

$$\begin{aligned} \boldsymbol{C} = {} & \frac{m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i}{\sum_{i=1}^{M}\underline{f}^i} + \frac{n\sum_{i=1}^{M}\overline{f}^i\boldsymbol{A}_i}{\sum_{i=1}^{M}\overline{f}^i} \\ & + mm'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}}{k_1\sum_{i=1}^{M}\underline{f}^i} + mn'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}}{k_1\sum_{i=1}^{M}\underline{f}^i} \\ & + nm'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\overline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}}{k_2\sum_{i=1}^{M}\overline{f}^i} + nn'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\overline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}}{k_2\sum_{i=1}^{M}\overline{f}^i} \end{aligned} \tag{3.69}$$

Although (3.68) may look like a linear system, it is not because $\boldsymbol{C}$ depends on $\boldsymbol{x}$ through the dependencies of $\underline{f}^i$ and $\overline{f}^i$ on $\boldsymbol{x}$.

3.3.3 Stability of closed-loop system

The stability of T1 TSK FLCSs using Fuzzy Lyapunov Function (FLF) has been addressed in several works ([136], [137]). Most notably, Tanaka et. al. in [136] proposed an FLF method composed of multiple Lyapunov functions to obtain the stability conditions for a T1 TSK FLCS. They also presented a new design control methodology using parallel distributed compensation. The proposed FLF methodology in [136] provided relaxed stability conditions for a T1 TSK FLCS; however, the design process required the time derivatives of premise MFs, and it is not always possible to derive such derivatives from the system states, which limits the use of this method. Because the same drawback will limit the use of this method for IT2 TSK FLCSs, in this thesis, a quadratic Lyapunov function is introduced to derive stability conditions for IT2 TSK FLCSs.

The proposed Lyapunov function is $V(\boldsymbol{x}(k)) = \boldsymbol{x}^T(k)\boldsymbol{P}\boldsymbol{x}(k)$, where $\boldsymbol{P}$ is a positive definite matrix [138]. $\Delta V(\boldsymbol{x}(k))$ is given by

$$\Delta V(\boldsymbol{x}(k)) = \boldsymbol{x}^T(k+1)\boldsymbol{P}\boldsymbol{x}(k+1) - \boldsymbol{x}^T(k)\boldsymbol{P}\boldsymbol{x}(k) \tag{3.70}$$

Using (3.68), $\Delta V(\boldsymbol{x}(k))$ can be expressed as

$$\Delta V(\boldsymbol{x}(k)) = \boldsymbol{x}^T(k)\boldsymbol{Z}\boldsymbol{x}(k) \tag{3.71}$$

where

$$\boldsymbol{Z} \equiv \boldsymbol{C}^T\boldsymbol{P}\boldsymbol{C} - \boldsymbol{P} \tag{3.72}$$

and $\boldsymbol{C}$ is given by (3.69). $\boldsymbol{Z}$ has 36 components, and can be expressed as

$$\boldsymbol{Z} \equiv \boldsymbol{Z}_1 + \boldsymbol{Z}_2 + \boldsymbol{Z}_3 \tag{3.73}$$

where $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, and $\boldsymbol{Z}_3$ are given in Appendix A.3.

Let the first bracketed term of $\boldsymbol{Z}_1$ in (A.1) be denoted $\boldsymbol{Z}_{1,1}$, i.e.,

$$\boldsymbol{Z}_{1,1} = \frac{m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{m\sum_{j=1}^{M}\underline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\underline{f}^j} - \frac{1}{36}\boldsymbol{P} \tag{3.74}$$

It can be expressed as

$$\boldsymbol{Z}_{1,1}\left(\sum_{i=1}^{M}\underline{f}^i\sum_{j=1}^{M}\underline{f}^j\right) = \sum_{i,j=1}^{M}\underline{f}^i\underline{f}^j\left[m^2\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_j - \frac{1}{36}\boldsymbol{P}\right] \tag{3.75}$$

Using the fact that $\underline{f}^i$ and $\underline{f}^j$ are positive, for $\boldsymbol{Z}_{1,1} < \boldsymbol{0}$ the expression inside the bracket in (3.75) must be negative definite; thus,

$$m^2\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_j - \frac{1}{36}\boldsymbol{P} < \boldsymbol{0} \tag{3.76}$$

where $i, j = 1, 2, ..., M$. It is straightforward to demonstrate that similar conditions are obtained for the remaining three terms of $\boldsymbol{Z}_1$, i.e.,

$$a\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_j - \frac{1}{36}\boldsymbol{P} < \boldsymbol{0} \tag{3.77}$$

where

$$a = \left\{m^2, mn, n^2\right\} \tag{3.78}$$

Applying the same method to $\boldsymbol{Z}_2$ and $\boldsymbol{Z}_3$, similar conditions can be obtained (some of the details are provided in Appendix A.4). The resulting conclusions are:

$$b\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{B}_{j,l} + b\boldsymbol{B}_{i,l}^T\boldsymbol{P}\boldsymbol{A}_j - \frac{1}{18}\boldsymbol{P} < \boldsymbol{0} \tag{3.79}$$

where

$$b = \{m^2m', mnm', m^2n', mnn', n^2m', n^2n'\} \tag{3.80}$$

$$c\boldsymbol{B}_{i,l}^T\boldsymbol{P}\boldsymbol{B}_{j,q} - \frac{1}{36}\boldsymbol{P} < \boldsymbol{0} \tag{3.81}$$

where

$$c = \left\{m^2m'^2, n^2n'^2, m^2m'n', mnm'n', m^2n'^2, mnn'^2, mnm'^2, n^2m'^2, n^2m'n'\right\} \tag{3.82}$$

In (3.77), (3.79), and (3.81), $i, j = 1, 2, ..., M$, and $l, q = 1, 2, ..., Q$. For each i, j, l and q combination, only 18 inequalities are given by (3.77)-(3.81), because half of the 36 inequality conditions for $\Delta V(\boldsymbol{x}(k)) < 0$ are repetitive. If there exists a common positive definite matrix $\boldsymbol{P}$ that satisfies the inequalities in (3.77)-(3.81), then the closed-loop system is globally asymptotically stable.

Note that combining the terms of $\boldsymbol{C}$ in (3.69) will not lead to simpler stability conditions, because when common terms are combined, the expressions inside the resulting multiple summations include *several* combinations of the $\boldsymbol{A}_i$ and $\boldsymbol{B}_{j,l}$ matrices, and hence require a larger number of inequalities to be satisfied.

Next, (3.77)-(3.81) are formulated into the LMI problems that can be solved using numerical techniques such as the interior point method [53]. Consider, e.g., one of the inequalities given by (3.77). By multiplying both sides of it by -36, it is straightforward to show that it can be rewritten as

$$\boldsymbol{P} - 36a\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_j > \boldsymbol{0} \tag{3.83}$$

Since (3.83) can be non-symmetric, it follows that its symmetric part must satisfy

$$\frac{1}{2}\left\{\left[\boldsymbol{P} - 36a\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_j\right]^T + \left[\boldsymbol{P} - 36a\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{A}_j\right]\right\} > \boldsymbol{0} \tag{3.84}$$

(3.84) can be re-expressed as ($i, j = 1, 2, ..., M$; a in (3.78))

$$\boldsymbol{P} - \frac{36a}{2}(\boldsymbol{A}_j^T \boldsymbol{P} \boldsymbol{A}_i + \boldsymbol{A}_i^T \boldsymbol{P} \boldsymbol{A}_j) > \boldsymbol{0} \tag{3.85}$$

It can be similarly shown that for (3.79) and (3.81) the following equivalent LMIs can be obtained [$i, j = 1, 2, ..., M$, and $l, q = 1, 2, ..., Q$; b in (3.80), and c in (3.82)]:

$$\boldsymbol{P} - \frac{18}{2} b \left(\boldsymbol{B}_{j,l}^T \boldsymbol{P} \boldsymbol{A}_i + \boldsymbol{A}_i^T \boldsymbol{P} \boldsymbol{B}_{j,l} \right) - \frac{18}{2} b \left(\boldsymbol{A}_j^T \boldsymbol{P} \boldsymbol{B}_{i,l} + \boldsymbol{B}_{i,l}^T \boldsymbol{P} \boldsymbol{A}_j \right) \ > \ \boldsymbol{0} \tag{3.86}$$

$$\boldsymbol{P} - \frac{36}{2} c (\boldsymbol{B}_{i,l}^T \boldsymbol{P} \boldsymbol{B}_{j,q} + \boldsymbol{B}_{j,l}^T \boldsymbol{P} \boldsymbol{B}_{i,q}) \ > \ \boldsymbol{0} \tag{3.87}$$

The stability conditions in (3.85)-(3.87) can be evaluated using standard software tools, such as the Matlab LMI toolbox[4] or the CVX[5] that have been developed to efficiently solve LMI problems.

If the LMIs given by (3.85)-(3.87) have a positive definite solution for $\boldsymbol{P}$, then system (3.65) is globally asymptotically stable.

3.3.4 Bounds for the controller tuning parameters

In this subsection a method is proposed for deriving the bounds for the controller tuning parameters m' and n' in (3.55). These bounds are used to find the controller tuning parameters in the next section. Paralleling the derivation in Section 3.2, it is straightforward to show that the controller tuning parameters, m' and n', are given by

$$m' = \frac{1}{2} + g_1' \tag{3.88}$$

$$n' = \frac{1}{2} + g_2' \tag{3.89}$$

where

$$g_1' = -\frac{1}{4} \frac{\sum_{i=1}^{Q} \left(\overline{v}^i(\boldsymbol{x}) - \underline{v}^i(\boldsymbol{x}) \right)}{\left[\sum_{i=1}^{Q} \underline{v}^i(\boldsymbol{x}) \sum_{p=1}^{n} c_p^i x(k-p+1) \right] \sum_{i=1}^{Q} \overline{v}^i(\boldsymbol{x})}$$
$$\times \frac{\sum_{i=1}^{Q} \left[\underline{v}^i(\boldsymbol{x}) \sum_{p=1}^{n} r_{i,p} c_p^i x(k-p+1) \right] \sum_{i=1}^{Q} \left[\overline{v}^i(\boldsymbol{x}) \sum_{p=1}^{n} s_{i,p} c_p^i x(k-p+1) \right]}{\sum_{i=1}^{Q} \left[\underline{v}^i(\boldsymbol{x}) \sum_{p=1}^{n} r_{i,p} c_p^i x(k-p+1) \right] + \sum_{i=1}^{Q} \left[\overline{v}^i(\boldsymbol{x}) \sum_{p=1}^{n} s_{i,p} c_p^i x(k-p+1) \right]} \tag{3.90}$$

[4] Matlab LMI toolbox solves semi-definite programming and LMI problems.

[5] CVX is a Matlab-based modeling system for convex optimization ([139], [140]).

$$g_2' = \frac{1}{4}\frac{\sum_{i=1}^{Q}\left(\overline{v}^i(\boldsymbol{x}) - \underline{v}^i(\boldsymbol{x})\right)}{\left[\sum_{i=1}^{Q}\overline{v}^i\left(\boldsymbol{x}\right)\sum_{p=1}^{n}c_p^i x(k-p+1)\right]\sum_{i=1}^{Q}\underline{v}^i(\boldsymbol{x})}$$

$$\times\frac{\sum_{i=1}^{Q}\left[\overline{v}^i(\boldsymbol{x})\sum_{p=1}^{n}r_{i,p}c_p^i x(k-p+1)\right]\sum_{i=1}^{Q}\left[\underline{v}^i(\boldsymbol{x})\sum_{p=1}^{n}s_{i,p}c_p^i x(k-p+1)\right]}{\sum_{i=1}^{Q}\left[\overline{v}^i(\boldsymbol{x})\sum_{p=1}^{n}r_{i,p}c_p^i x(k-p+1)\right]+\sum_{i=1}^{Q}\left[\underline{v}^i(\boldsymbol{x})\sum_{p=1}^{n}s_{i,p}c_p^i x(k-p+1)\right]} \quad (3.91)$$

and

$$r_{i,p} \equiv \frac{c_p^i - c_p^1}{c_p^i} \quad (3.92)$$

$$s_{i,p} \equiv \frac{c_p^Q - c_p^i}{c_p^i} \quad (3.93)$$

When IT2 TSK is used for control design, bounds on g_1' and g_2' from (3.90) and (3.91) can be determined, i.e.,

$$g_1'^{\min} \leq g_1' \leq g_1'^{\max} \quad (3.94)$$

$$g_2'^{\min} \leq g_2' \leq g_2'^{\max} \quad (3.95)$$

It follows from (3.88), (3.89), (3.94), and (3.95) that the bounds on m' and n' are given as

$$m'^{\min} \equiv \frac{1}{2} + g_1'^{\min} \leq m' \leq \frac{1}{2} + g_1'^{\max} \equiv m'^{\max} \quad (3.96)$$

$$n'^{\min} \equiv \frac{1}{2} + g_2'^{\min} \leq n' \leq \frac{1}{2} + g_2'^{\max} \equiv n'^{\max} \quad (3.97)$$

3.3.5 An Algorithm to find the controller tuning parameters

The proposed algorithm to find the controller tuning parameters, m' and n', is as follows:

Algorithm 2 Finding the controller tuning parameters.

```
$m' \leftarrow m'^{min}$
**repeat**
  **repeat**
    $n' \leftarrow n'^{min}$
    Solve the LMIs given by (3.85)-(3.87), and determine the feasibility/infeasibility [6]
    of $\boldsymbol{P}$
    If the LMIs are feasible, save $m'$, $n'$, and $\boldsymbol{P}$, and exit the loop (inner loop)
    Increment $n'$, i.e., $n' \leftarrow n' + \Delta n'$ (where $\Delta n' = 0.05n'$)
  **until** $n' \leq n'^{max}$
  $m' \leftarrow m' + \Delta m'$ (where $\Delta m' = 0.05m'$)
**until** $m' \leq m'^{max}$
```

At the end of this algorithm, a set of feasible (m', n', $\boldsymbol{P}$) are found. The designer chooses the specific set that achieves the best transient performance. An advantage of this algorithm is that there is no need to change the controller TSK parameters. Using this algorithm, when m' and n' are chosen by tuning only two controller parameters, stabilizing the system is easy to achieve.

3.4 Stability of MIMO IT2 TS FLCSs

In this section, unlike Section 3.3 where IT2 TSK was used for plant to derive the stability conditions, the plant model is simplified to a T1 TSK. This simplifies the process of stability analysis for multi-input multi-output (MIMO) IT2 control systems. This assumption is reasonable because T1 TSK FLSs have been proven to be universal approximators [122], and can model nonlinear plants relatively well [141]. Hence, the focus on this section will be the stability analysis of IT2 TS FLCSs that use T1 TS plant model. To begin, the structure of the hybrid system for a T1 TS FLCS is reviewed.

A. T1 TS FLCS

The general ith rule for the plant is now given as [141]

$$R_p^i\text{: If } x(k) \text{ is } P_1^i \text{ and } \cdots \text{ and } x(k-n+1) \text{ is } P_n^i\text{, Then } \boldsymbol{x}_i(k+1) = \boldsymbol{A}_i\mathbf{x}(k)+\boldsymbol{b}_i\boldsymbol{u}(k),\ i = 1, 2, \cdots, r \tag{3.98}$$

where r is the number of rules, $\boldsymbol{x}_i(k+1)$ is the output of each rule, P_j^i represents the T1 FS of input state j of rule i, $\boldsymbol{x}(k)$ is the state vector and is given by (2.20) , and $\boldsymbol{A}_i \in \mathbb{R}^{n\times n}$, $\boldsymbol{b}_i \in \mathbb{R}^{n\times m}$, $\boldsymbol{u}(k) \in \mathbb{R}^m$. The output of the system, $\boldsymbol{x}(k+1)$, is given by

$$\boldsymbol{x}(k+1) = \frac{\sum_{i=1}^{r} q_i(k)\left\{\boldsymbol{A}_i\mathbf{x}(k) + \boldsymbol{b}_i\boldsymbol{u}(k)\right\}}{\sum_{i=1}^{n} q_i(k)} \tag{3.99}$$

where

$$q_i(k) = \mu_{P_1^i}\left(x(k)\right) * \cdots * \mu_{P_n^i}\left(x(k-n+1)\right) \tag{3.100}$$

The ith control rule is [141]

$$R_c^i\text{: If } x(k) \text{ is } C_1^i \text{ and } \cdots \text{ and } x(k-n+1) \text{ is } C_n^i\text{, Then } \boldsymbol{u}_i(k) = \boldsymbol{F}_i\mathbf{x}(k),\ i = 1, 2, \cdots, r \tag{3.101}$$

where $\boldsymbol{F}_j$ is the jth feedback gain matrix of the consequent part, and C_j^i represents the T1 FS of input state j of rule i. Note that the number of rules for the controller is also r. The controller output, $\boldsymbol{u}(k)$, for a system that uses a T1 TS FLCS is given by [141]

$$\boldsymbol{u}(k) = \frac{\sum_{j=1}^{r} w_j(k)\boldsymbol{F}_j\mathbf{x}(k)}{\sum_{j=1}^{r} w_j(k)} \tag{3.102}$$

where

$$w_i(k) = \mu_{C_1^i}\left(x(k)\right) * \cdots * \mu_{C_n^i}\left(x(k-n+1)\right) \tag{3.103}$$

The complete closed-loop modified T1 TS FLCS is obtained by substituting (3.102) into (3.99).

B. IT2 TS FLCS

As mentioned at the start of Section 3.4 the IT2 TS FLS utilizes an IT2 TSK FLS for the controller and the T1 TS FLS in (3.99) for the plant. The rule structure for the IT2 TS FLCS is kept the same as (3.101) except that C_j^is are replaced with IT2 FSs, i.e., $\widetilde{C}_i^j$. Now, however, $\boldsymbol{u}(k)$, has the same structure as (3.55) and is given as

$$\boldsymbol{u}(k) = m'\frac{\sum_{j=1}^{r} \underline{w}_j(k)\boldsymbol{F}_j\mathbf{x}(k)}{\sum_{j=1}^{r} \underline{w}_j(k)} + n'\frac{\sum_{j=1}^{r} \overline{w}_j(k)\boldsymbol{F}_j\mathbf{x}(k)}{\sum_{j=1}^{r} \overline{w}_j(k)} \tag{3.104}$$

where

$$\underline{w}_j(k) = \underline{\mu}_{\widetilde{C}_1^j}\left(x(k)\right) * \cdots * \underline{\mu}_{\widetilde{C}_n^j}\left(x(k-n+1)\right) \tag{3.105}$$

$$\overline{w}_j(k) = \overline{\mu}_{\widetilde{C}_1^j}\left(x(k)\right) * \cdots * \overline{\mu}_{\widetilde{C}_n^j}\left(x(k-n+1)\right) \tag{3.106}$$

Substituting (3.104) into (3.99), the output of the system, $\boldsymbol{x}(k+1)$, can be expressed as

$$\begin{aligned}
\boldsymbol{x}(k+1) &= \frac{\sum_{i=1}^{r} q_i(k) \left\{ \boldsymbol{A}_i \mathbf{x}(k) + \boldsymbol{b}_i m' \frac{\sum_{j=1}^{r} \underline{w}_j(k) \boldsymbol{F}_j \mathbf{x}(k)}{\sum_{j=1}^{r} \underline{w}_j(k)} + n' \frac{\sum_{j=1}^{r} \overline{w}_j(k) \boldsymbol{F}_j \mathbf{x}(k)}{\sum_{j=1}^{r} \overline{w}_j(k)} \right\}}{\sum_{i=1}^{n} q_i(k)} \\
&= \frac{\sum_{i,j,l=1}^{r} q_i(k) \underline{w}_j(k) \overline{w}_l(k) \left\{ \boldsymbol{A}_i \mathbf{x}(k) + m' \boldsymbol{b}_i \boldsymbol{F}_j \mathbf{x}(k) + n' \boldsymbol{b}_i \boldsymbol{F}_l \mathbf{x}(k) \right\}}{\sum_{i,j,l=1}^{r} q_i(k) \underline{w}_j(k) \overline{w}_l(k)} \qquad (3.107)
\end{aligned}$$

which can be further expressed in a more compact form as

$$\boldsymbol{x}(k+1) = \frac{\sum_{i,j,l=1}^{r} g_{ijl}(k) \boldsymbol{G}_{ijl}}{\sum_{i,j,l=1}^{r} g_{ijl}(k)} \mathbf{x}(k) \qquad (3.108)$$

where

$$\begin{aligned}
g_{ijl}(k) &= q_i(k) \underline{w}_j(k) \overline{w}_l(k) \qquad (3.109) \\
\boldsymbol{G}_{ijl} &= \boldsymbol{A}_i + m' \boldsymbol{b}_i \boldsymbol{F}_j + n' \boldsymbol{b}_i \boldsymbol{F}_l \qquad (3.110)
\end{aligned}$$

It is straightforward to show that $\sum_{i,j,l=1}^{r} \boldsymbol{G}_{ijl}$ can be expressed as

$$\sum_{i,j,l=1}^{r} \boldsymbol{G}_{ijl} = \sum_{i=1}^{r} \boldsymbol{G}_{iii} + \sum_{i \neq j}^{r} \sum_{j=1}^{r} \boldsymbol{G}_{ijj} + \sum_{i=1}^{r} \sum_{j \neq l}^{r} \sum_{l=1}^{r} \boldsymbol{G}_{ijl} \qquad (3.111)$$

Observe that

$$\begin{aligned}
\sum_{i \neq j}^{r} \sum_{j=1}^{r} \boldsymbol{G}_{ijj} &= \sum_{i<j}^{r} \sum_{j=1}^{r} \boldsymbol{G}_{ijj} + \sum_{i>j}^{r} \sum_{j=1}^{r} \boldsymbol{G}_{ijj} \\
&= \sum_{i<j}^{r} \sum_{j=1}^{r} \boldsymbol{G}_{ijj} + \sum_{t<p}^{r} \sum_{t=1}^{r} \boldsymbol{G}_{ptt} = 2 \sum_{i<j}^{r} \sum_{j=1}^{r} \left[\frac{\boldsymbol{G}_{ijj} + \boldsymbol{G}_{jii}}{2} \right] \qquad (3.112)
\end{aligned}$$

and

$$\begin{aligned}
\sum_{i=1}^{r} \sum_{j \neq l}^{r} \sum_{l=1}^{r} \boldsymbol{G}_{ijl} &= \sum_{i=1}^{r} \sum_{j<l}^{r} \sum_{l=1}^{r} \boldsymbol{G}_{ijl} + \sum_{i=1}^{r} \sum_{j>l}^{r} \sum_{l=1}^{r} \boldsymbol{G}_{ijl} \\
&= \sum_{i=1}^{r} \sum_{j<l}^{r} \sum_{l=1}^{r} \boldsymbol{G}_{ijl} + \sum_{i=1}^{r} \sum_{p>t}^{r} \sum_{l=1}^{r} \boldsymbol{G}_{ipt} = \sum_{i=1}^{r} \sum_{j<l}^{r} \sum_{l=1}^{r} \boldsymbol{G}_{ijl} + \sum_{i=1}^{r} \sum_{l>j}^{r} \sum_{l=1}^{r} \boldsymbol{G}_{ilj} \\
&= 2 \sum_{i=1}^{r} \sum_{j<l}^{r} \sum_{l=1}^{r} \left[\frac{\boldsymbol{G}_{ijl} + \boldsymbol{G}_{ilj}}{2} \right] \qquad (3.113)
\end{aligned}$$

Substituting (3.112) and (3.113) into (3.111), (3.111) is obtained.

Next define $\boldsymbol{H}_t$ and $\boldsymbol{v}_t(k)$ as:

$$\boldsymbol{H}_t \equiv \begin{cases} \frac{\boldsymbol{G}_{ijl}+\boldsymbol{G}_{ilj}}{2} & t = i + r(j-1+\frac{(l-1)(l-2)}{2}) \quad \text{and} \quad j < l \\ \frac{\boldsymbol{G}_{ijj}+\boldsymbol{G}_{jii}}{2} & t = i + \frac{j(j-1)}{2} + \frac{r^2(r-1)}{2} \quad \text{and} \quad j = l, i < j \\ \boldsymbol{G}_{iii} & t = i + \frac{i(i-1)}{2} + \frac{r^2(r-1)}{2} \quad \text{and} \quad i = j = l \end{cases} \tag{3.114}$$

$$v_t(k) \equiv \begin{cases} 2g_{ijl}(k) & t = i + r(j-1+\frac{(l-1)(l-2)}{2}) \quad \text{and} \quad j < l \\ 2g_{ijj}(k) & t = i + \frac{j(j-1)}{2} + \frac{r^2(r-1)}{2} \quad \text{and} \quad j = l, i < j \\ g_{iii}(k) & t = i + \frac{i(i-1)}{2} + \frac{r^2(r-1)}{2} \quad \text{and} \quad i = j = l \end{cases} \tag{3.115}$$

Note that the number of $\boldsymbol{H}_t$ matrices given in (3.114) is calculated as

$$\frac{r(r+1)}{2} + \frac{r^2(r-1)}{2} = \frac{r(r^2+1)}{2} \tag{3.116}$$

where $\frac{r(r+1)}{2}$ is the number of $\boldsymbol{H}_t$ for the case when $i \leq j$ and $j = l$, and $\frac{r^2(r-1)}{2}$ is the number of $\boldsymbol{H}_t$ matrices for the case when $j < l$. Observe that $\boldsymbol{H}_t$ matrices are functions of the $\boldsymbol{G}_{ijl}$ matrices that are dependent on the control parameters m' and n' in (3.110).

Using (3.114) and (3.115), (3.108) can be re-expressed using a single summation, as:

$$\boldsymbol{x}(k+1) = \frac{\sum_{s=1}^{\frac{r(r^2+1)}{2}} v_s(k)\boldsymbol{H}_s}{\sum_{s=1}^{\frac{r(r^2+1)}{2}} v_s(k)}\mathbf{x}(k) \tag{3.117}$$

It is well known [141] that system (3.117) is globally asymptotically stable if there exists a common positive-definite matrix $\boldsymbol{P}$ such that

$$\boldsymbol{H}_s^T \boldsymbol{P} \boldsymbol{H}_s - \boldsymbol{P} < \mathbf{0} \tag{3.118}$$

for $s = 1, 2, \ldots, \frac{r(r^2+1)}{2}$.

Observe, from (3.108) that r^3 LMIs (note the three summations) must be satisfied to ensure the stability of the IT2 TS FLCS; however, by introducing the $\boldsymbol{H}_t$ matrices, only $\frac{r(r^2+1)}{2} < r^3$ LMIs need to be satisfied to arrive at the same result.

Before assessing the feasibility of the LMIs in (3.118), it is important to identify the bounds on the controller tuning parameters m' and n' in (3.104) because $\boldsymbol{H}_s$ depends on them [$\boldsymbol{H}_t$ matrices are functions of the $\boldsymbol{G}_{ijl}$ matrices (3.110)]. When WM UB in (3.22)-(3.23) is used an as inference engine, the controller output is $\boldsymbol{u}_{\text{WM}}(\boldsymbol{x})$ and hence each of its component $u^j_{\mathrm{WM}}(\boldsymbol{x})$ for $j = 1, ..., r$, can be expressed as $[[\boldsymbol{F}_i\mathbf{x}(k)]^j$ replaces $\sum_{p=1}^{n} a_p^i x(k-p+1)$ in (3.34)]

$$u^j_{\mathrm{WM}}(\boldsymbol{x}) = \frac{1}{2}\frac{\sum_{i=1}^{r}\underline{w}^i\left(\boldsymbol{x}\right)\left([\boldsymbol{F}_i\mathbf{x}(k)]^j\right)}{\sum_{i=1}^{r}\underline{w}^i\left(\boldsymbol{x}\right)} + \frac{1}{2}\frac{\sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})\left([\boldsymbol{F}_i\mathbf{x}(k)]^j\right)}{\sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})}$$
$$+ g_1^{\prime j} \times \frac{\sum_{i=1}^{r}\underline{w}^i\left(\boldsymbol{x}\right)[\boldsymbol{F}_i\mathbf{x}(k)]^j}{\sum_{i=1}^{r}\underline{w}^i\left(\boldsymbol{x}\right)} + g_2^{\prime j} \times \frac{\sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})[\boldsymbol{F}_i\mathbf{x}(k)]^j}{\sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})} \quad (3.119)$$

where $g_1^{\prime j}$ and $g_2^{\prime j}$ are given by

$$g_1^{\prime j} = \frac{1}{4}\frac{-\sum_{i=1}^{r}\left(\overline{w}^i(\boldsymbol{x}) - \underline{w}^i(\boldsymbol{x})\right)}{\sum_{i=1}^{r}\overline{w}^i\left(\boldsymbol{x}\right)\sum_{i=1}^{r}\underline{w}^i\left(\boldsymbol{x}\right)[\boldsymbol{F}_i\mathbf{x}(k)]^j}\frac{\sum_{i=1}^{r}\underline{w}^i(\boldsymbol{x})\left[\left(\boldsymbol{F}_i - \boldsymbol{F}_1\right)\mathbf{x}(k)\right]^j . \sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})\left[\left(\boldsymbol{F}_r - \boldsymbol{F}_i\right)\mathbf{x}(k)\right]^j}{\sum_{i=1}^{r}\underline{w}^i(\boldsymbol{x})\left[\left(\boldsymbol{F}_i - \boldsymbol{F}_1\right)\mathbf{x}(k)\right]^j + \sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})\left[\left(\boldsymbol{F}_r - \boldsymbol{F}_i\right)\mathbf{x}(k)\right]^j} \quad (3.120)$$

$$g_2^{\prime j} = \frac{1}{4}\frac{\sum_{i=1}^{r}\left(\overline{w}^i(\boldsymbol{x}) - \underline{w}^i(\boldsymbol{x})\right)}{\sum_{i=1}^{r}\underline{w}^i\left(\boldsymbol{x}\right)\sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})[\boldsymbol{F}_i\mathbf{x}(k)]^j} . \frac{\sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})\left[\left(\boldsymbol{F}_i - \boldsymbol{F}_1\right)\mathbf{x}(k)\right]^j . \sum_{i=1}^{r}\underline{w}^i(\boldsymbol{x})\left[\left(\boldsymbol{F}_r - \boldsymbol{F}_i\right)\mathbf{x}(k)\right]^j}{\sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})\left[\left(\boldsymbol{F}_i - \boldsymbol{F}_1\right)\mathbf{x}(k)\right]^j + \sum_{i=1}^{r}\underline{w}^i(\boldsymbol{x})\left[\left(\boldsymbol{F}_r - \boldsymbol{F}_i\right)\mathbf{x}(k)\right]^j} \quad (3.121)$$

$u^j_{\mathrm{WM}}(\boldsymbol{x})$ can therefore be expressed as

$$u^j_{\mathrm{WM}}(\boldsymbol{x}) = \frac{\sum_{i=1}^{r}\underline{w}^i\left(\boldsymbol{x}\right)\overbrace{\left(\frac{1}{2} + g_1^{\prime j}\right)}^{m'_j}[\boldsymbol{F}_i\mathbf{x}(k)]^j}{\sum_{i=1}^{r}\underline{w}^i\left(\boldsymbol{x}\right)} + \frac{\sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})\overbrace{\left(\frac{1}{2} + g_2^{\prime j}\right)}^{n'_j}[\boldsymbol{F}_i\mathbf{x}(k)]^j}{\sum_{i=1}^{r}\overline{w}^i(\boldsymbol{x})} \quad (3.122)$$

Finally, the bounds of m' and n' are given by

$$m'^{\min} \equiv \min_j(\frac{1}{2} + g_1^{\prime j\,\min}) \leq m' \leq \max_j\left(\frac{1}{2} + g_1^{\prime j\,\max}\right) \equiv m'^{\max} \quad (3.123)$$
$$n'^{\min} \equiv \min_j(\frac{1}{2} + g_2^{\prime j\,\min}) \leq n' \leq \max_j\left(\frac{1}{2} + g_2^{\prime j\,\max}\right) \equiv n'^{\max} \quad (3.124)$$

where $j = 1, ..., r$.

Next observe that (3.118) can be transformed into the following LMIs ($s = 1, 2, \ldots, \frac{r(r^2+1)}{2}$):

$$\boldsymbol{P} - \boldsymbol{H}_s^T\boldsymbol{P}\boldsymbol{H}_s > \mathbf{0} \quad (3.125)$$

Let $\boldsymbol{X} \equiv \boldsymbol{P}^{-1}$ and multiply both sides of (3.125) by $\boldsymbol{X}$ from left and right. It is straightforward to see that (3.125) becomes

$$\boldsymbol{X} - \boldsymbol{X}\boldsymbol{H}_s^T\boldsymbol{X}^{-1}\boldsymbol{H}_s\boldsymbol{X}^{-1} > \mathbf{0} \quad (3.126)$$

which is equivalent to the following LMIs ($s = 1, 2, \ldots, \frac{r(r^2+1)}{2}$):

$$\begin{bmatrix} \boldsymbol{X} & \boldsymbol{X}\boldsymbol{H}_s^T \\ \boldsymbol{H}_s\boldsymbol{X} & \boldsymbol{X} \end{bmatrix} > \mathbf{0} \tag{3.127}$$

Using the bounds of m' and n' found in (3.123)-(3.124) and a procedure similar to the algorithm in Subsection 3.3.5 [(3.85)-(3.87) are replaced with (3.127)], the feasibility of (3.127) can be investigated using the Matlab LMI toolbox or the CVX. If a positive definite $\boldsymbol{X}$ exists, then the closed-loop system will be asymptotically stable.

To ensure stability in a Lyapunov sense, it is required that $\Delta V(\boldsymbol{x}(k)) < 0$; hence, if all the components of $\Delta V(\boldsymbol{x}(k))$ are made negative (equivalently, all the components of $\boldsymbol{Z}$ are made negative definite) the result will be an asymptotically stable system.

3.5 Examples

This section introduces two examples. The first presents applications of the developed theory to analyze the stability of SISO and MIMO systems, respectively. The second example presents case studies demonstrating the effectiveness of the IT2 TS FLCS in tracking applications and control of nonlinear systems.

Example 1:

This example presents two case studies. The first deals with a SISO system defined in Subsection 3.3.3. The second deals with a MIMO system described in Section 3.4.

Case study a: Consider the following SISO IT2 TSK FLS:

If $x(k)$ is $\widetilde{F}^1$and $x(k-1)$ is $\widetilde{F}^2$, then $x^1(k+1) = 2.3x(k) - 2x(k-1) + 0.7u(k)$.

If $x(k)$ is $\widetilde{F}^2$and $x(k-1)$ is $\widetilde{F}^1$, then $x^2(k+1) = 1.5x(k) - 1x(k-1) + 0.01u(k)$.

This system has two control rules:

If $x(k)$ is $\widetilde{C}^1$and $x(k-1)$ is $\widetilde{C}^2$ then $u^1(k+1) = -0.9x(k) - 1.08x(k-1)$.

If $x(k)$ is $\widetilde{C}^2$and $x(k-1)$ is $\widetilde{C}^1$ then $u^2(k+1) = 1.4x(k) - 2.1x(k-1)$.

The MFs for the plant, $\widetilde{F}^1$ and $\widetilde{F}^2$, and the controller, $\widetilde{C}^1$ and $\widetilde{C}^2$, are shown in Figure 3.2.

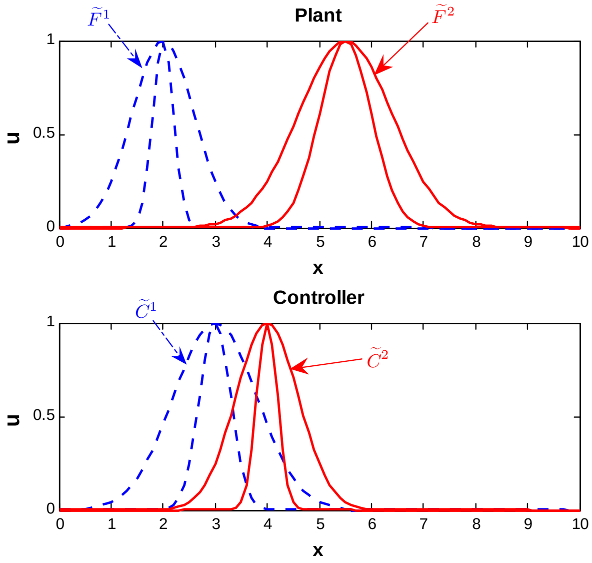


Figure 3.2: MFs for Example 1, case study a.

The $\boldsymbol{B}_{ij}$ and $\boldsymbol{A}_i$ matrices, according to (3.66), are:

$$\boldsymbol{A}_1 = \begin{bmatrix} 2.3 & -2 \\ 1 & 0 \end{bmatrix}, \; \boldsymbol{A}_2 = \begin{bmatrix} 1.5 & -1 \\ 1 & 0 \end{bmatrix} \tag{3.128}$$

$$\boldsymbol{B}_{11} = \begin{bmatrix} -0.63 & -0.756 \\ 1 & 0 \end{bmatrix}, \; \boldsymbol{B}_{12} = \begin{bmatrix} -0.009 & -0.011 \\ 1 & 0 \end{bmatrix} \tag{3.129}$$

$$\boldsymbol{B}_{21} = \begin{bmatrix} 0.980 & -1.470 \\ 1 & 0 \end{bmatrix}, \; \boldsymbol{B}_{22} = \begin{bmatrix} 0.014 & -0.021 \\ 1 & 0 \end{bmatrix} \tag{3.130}$$

Note that in this example m and n in (3.62), which are plant parameters, are assumed given as $m = 0.1$ and $n = 0.1$. The controller tuning parameters, m' and n', were designed based on the method introduced in Sections 3.3.4 and 3.3.5 (note that this is a stabilizing controller design problem). Bounds on the states were assumed to be $[-3, 1]$. Using the algorithm of Section 3.3.5, the bounds for m' and n' were obtained as $[0.176, 0.5]$ and $[0.149, 1.299]$, respectively. Table 3.1 summarizes some of the selected values for m' and n' and their corresponding matrix $\boldsymbol{P}$.

Table 3.1: Some selected controller tuning parameters and their corresponding $\boldsymbol{P}$.

m'	n'	$\boldsymbol{P}$
0.2	0.2	$\begin{bmatrix} 24.691 & -23.500 \\ -23.500 & 44.044 \end{bmatrix}$
0.32	0.32	$\begin{bmatrix} 21.279 & -20.197 \\ -20.197 & 37.998 \end{bmatrix}$
0.45	0.38	$\begin{bmatrix} 16.645 & -15.745 \\ -15.745 & 29.746 \end{bmatrix}$

Tuning parameters m' and n' were selected that resulted in the best output transient response, e.g, when the initial conditions are $x(1) = 0.1$ and $x(2) = 0.01$, the response of the system for the m' and n' in Table 3.1 are shown in Figure 3.3.

The output of the system shown with solid line ($m' = n' = 0.2$) has the best transient response.

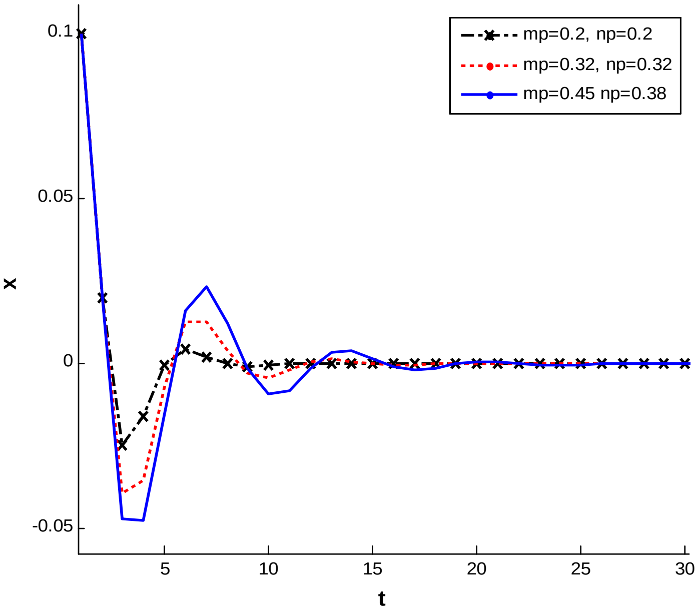


Figure 3.3: Closed-loop system response for different controller tuning parameters.

Case study b:

In this case study, an IT2 TS FLC is applied to stabilize an inverted pendulum which is an example of a benchmark problem often used in the design of controllers. Figure 3.4

shows an inverted pendulum located on a cart. The control problem is to stabilize the inverted pendulum by applying a horizontal force to the system (control action).

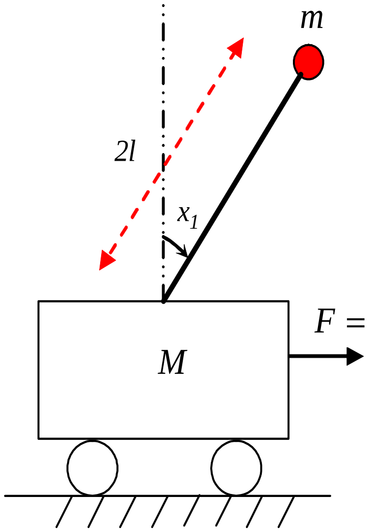


Figure 3.4: Inverted pendulum.

Tanaka and Wang [54] demonstrated that for certain initial conditions (angles), a linear controller is not capable of stabilizing the system. Hence, they introduced a T1 TS FLS that can be used to model as well as control this inverted pendulum. In this case study, it is shown that the proposed IT2 TS FLC in Section 3.4 is capable of stabilizing the inverted pendulum while achieving a better performance compared to its T1 counterpart.

The inverted pendulum system has nonlinear dynamics and the equations of motion are given as

$$\dot{x}_1(t) = x_2(t) \tag{3.131}$$

$$\dot{x}_2(t) = \frac{g\sin(x_1(t)) - amlx_2^2(t)\sin(2x_1(t))/2 - a\cos(x_1(t))u(t)}{4l/3 - aml\cos^2(x_1(t))} \tag{3.132}$$

where $x_1(t)$ and $x_2(t)$ are the angular position and velocity of the pendulum, respectively, $u(t)$ is the control input, m is the pendulum mass, M is the cart mass, $2l$ is the length of the pendulum, and $a \equiv \frac{1}{m+M}$.

The performance of the IT2 TS FLC is also compared with a nonlinear controller [142] as well as a linear controller. The structure of the nonlinear controller is given by

$$\begin{aligned} u(t) &= \frac{g}{a}\tan(x_1(t)) + \frac{4le_1e_2}{3a}\ln[\sec(x_1(t) + \tan(x_1(t)] - e_1e_2ml\sin(x_1(t)) \\ & -\frac{(e_1+e_2)x_2(t)}{a}[\frac{4l}{3}\sec(x_1(t)) - aml\cos(x_1(t))] \end{aligned} \tag{3.133}$$

where e_1 and e_2 are specified closed-loop eigenvalues. To compare the performance of the IT2 TS FLC with the T1 controller, the model of the plant is kept as a T1 TS and only the controller is replaced with an IT2 TS model. To make a fair comparison, the parameters of the plants and controllers are kept unchanged for both control systems, and only the MFs for the IT2 controller are designed.

Define $\boldsymbol{x}(t) \equiv [x_1(t), x_2(t)]^T$ where $x_1(t)$ and $x_2(t)$ are the state variables, i.e., angular position and velocity of the pendulum. The structure of the plant and the controllers is given by

Plant rules: (see Figure 3.5 for antecedent MFs)[7]

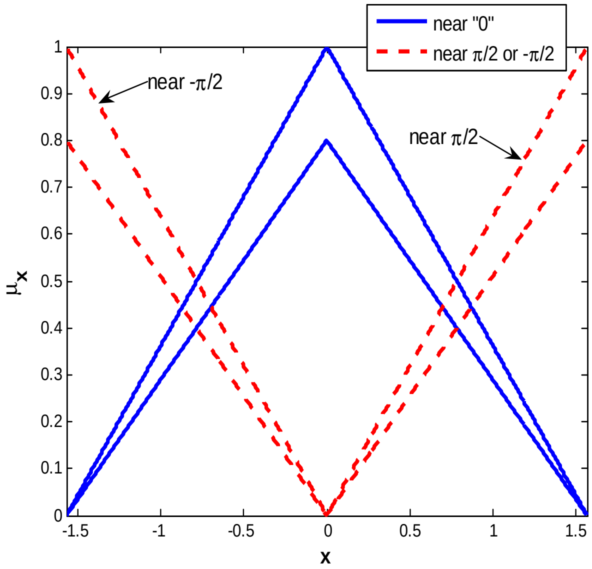


Figure 3.5: IT2 MFs for Example 1, case study b.

Rule 1: If $x_1(t)$ is "about 0", then $\dot{\boldsymbol{x}} = \boldsymbol{A}_1\boldsymbol{x}(t) + \boldsymbol{b}_1\boldsymbol{u}(t)$

Rule 2: If $x_1(t)$ is "about $\frac{\pi}{2}$ or $-\frac{\pi}{2}$", then $\dot{\boldsymbol{x}} = \boldsymbol{A}_2\boldsymbol{x}(t) + \boldsymbol{b}_2\boldsymbol{u}(t)$

Control rules:

[7] x_1 is given in radians.

Control Rule 1: If $x_1(t)$ is "about 0", then $u(t) = \boldsymbol{f}_1\boldsymbol{x}(t)$

Control Rule 2: If $x_1(t)$ is "about $\frac{\pi}{2}$ or $-\frac{\pi}{2}$ ", then $u(t) = \boldsymbol{f}_2\boldsymbol{x}(t)$

in which

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 \\ 17.3118 & 0 \end{bmatrix}, \ \boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 \\ 9.3696 & 0 \end{bmatrix} \tag{3.134}$$

$$\boldsymbol{b}_1 = \begin{bmatrix} 0 \\ -0.1765 \end{bmatrix}, \ \boldsymbol{b}_2 = \begin{bmatrix} 0 \\ -0.0052 \end{bmatrix} \tag{3.135}$$

$$\boldsymbol{f}_1 = \begin{bmatrix} 120.6667 & 22.6667 \end{bmatrix}, \ \boldsymbol{f}_2 = \begin{bmatrix} 2551.6 & 0.7640 \end{bmatrix} \tag{3.136}$$

Note that in order to make an unbiased comparison, $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are adopted from [141]. Those gains were chosen by a pole-placement method. The linear controller is given by $u(t) = \boldsymbol{f}_1\boldsymbol{x}(t)$. The values of the parameters used in this case study are: $m = 2kg$, $M = 8kg$, $l = 0.5m$, $a = 0.1kg^{-1}$, $g = 9.81\frac{m}{s^2}$, $e_1 = e_2 = -2$, and the tuning parameters for the IT2 controllers are $m' = 1$ and $n' = 0.9$. Note that this example deals with a continuous system. Following the same approach explained in Section 3.4, it is very straightforward to show that the stability conditions for the continuous system can be simply written as

$$\boldsymbol{H}_s^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{H}_s < \boldsymbol{0} \tag{3.137}$$

where $\boldsymbol{H}_s$ matrices are defined by (3.114), and $\boldsymbol{P}$ is a positive definite matrix. Using the Matlab LMI toolbox, it is easy to obtain the following $\boldsymbol{P}$ matrix satisfying (3.137):

$$\boldsymbol{P} = \begin{bmatrix} 0.6219 & 0.0852 \\ 0.0852 & 0.0324 \end{bmatrix} \tag{3.138}$$

Figure 3.6 shows the performance of the different controllers simulated for the initial angle of $x_1(0) = 0.105$ rad (note that the responses of the linear and nonlinear controllers are almost the same for this specific initial angle). Clearly, IT2 controller not only stabilizes the system, but also results in enhanced transient performances, i.e, reduced rise time and faster settling time. The rise and settling times for the T1 controller are $t_r = 0.64$ sec and $t_s = 1.16$ sec, respectively. For the nonlinear controller, the corresponding values are $t_r = 0.49$ sec and $t_s = 0.9$ sec. Finally, for the IT2 controller these values are $t_r = 0.49$ sec and $t_s = 0.90$ sec. The transient response of each controller for different initial angles has been summarized in Table 3.2.

Table 3.2: Rise and settling times of different controllers for different initial angles.

	$x_1(0) = 0.105$ rad		$x_1(0) = 0.873$ rad		$x_1(0) = 1.483$ rad	
	$t_r(s)$	$t_s(s)$	$t_r(s)$	$t_s(s)$	$t_r(s)$	$t_s(s)$
Linear	1.72	3.42	unstable	unstable	unstable	unstable
T1	0.64	1.16	0.63	1.53	0.65	1.75
Nonlinear	1.72	3.42	1.83	3.47	2.44	3.47
IT2	0.49	0.90	0.62	1.45	0.63	1.64

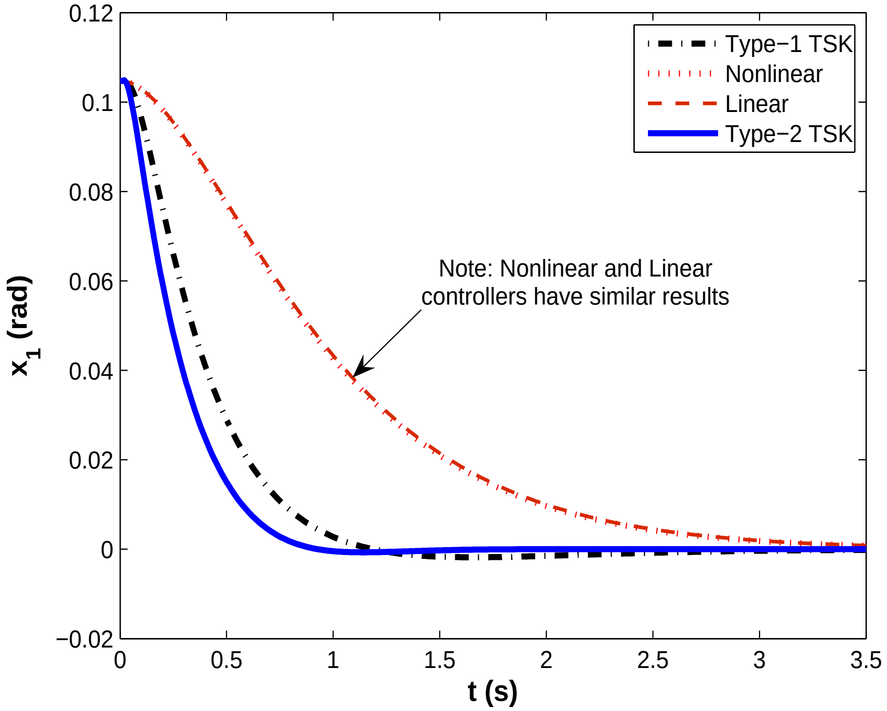
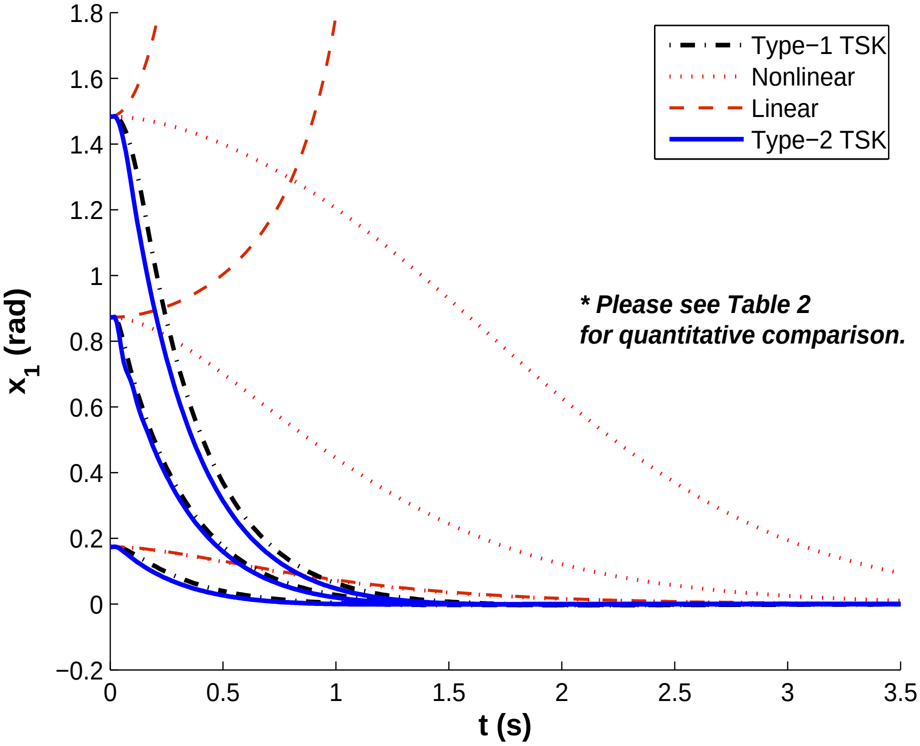


Figure 3.6: Outputs of different controllers for the initial angle of $x_1(0) = 0.105rad$.

Figure 3.7 compares the response of each controller for different initial angles. The nonlinear, T1, and IT2 controllers are capable of stabilizing the system for all initial angles $x_1(0) \in (0, \frac{\pi}{2})$. For initial angles $x_1(0) > 0.7854$ rad, the linear controller, however, fails to stabilize the system. Furthermore, the IT2 controller is consistently outperforming other controllers in terms of transient response. The IT2 controller offers a simpler structure that does not have the complexity of the nonlinear controller in (3.133), yet it performs considerably better while concurrently ensuring the stability of the nonlinear system.



Please see Table 2 for quantitative comparison.

Figure 3.7: Outputs of different controllers for different initial angles.

Example 2:

This example offers two more case studies that demonstrate the effectiveness of the proposed IT2 TS FLC for tracking applications and control of nonlinear dynamic systems such as chaotic oscillators.

Case study c: (tracking application)

This control example is adopted from [141] where a T1 TS controller was designed to track a predefined trajectory of a model car. The specific problem is to control a computer-simulated model car from an arbitrary initial position by manipulating the steering angle and allowing only forward movements. The car is the plant and is modeled by a T1 TS FLCS. In [141], it was verified that the dynamics of the approximated fuzzy model agree with the original model. In Figure 3.8, observe that x_0 is the angle that the car makes with the horizontal axis and x_1 is the vertical position of the rear end of the car. The control objective is to track the car from a given initial position to the position where $x_0 = x_1 = 0$ with no backward movement.

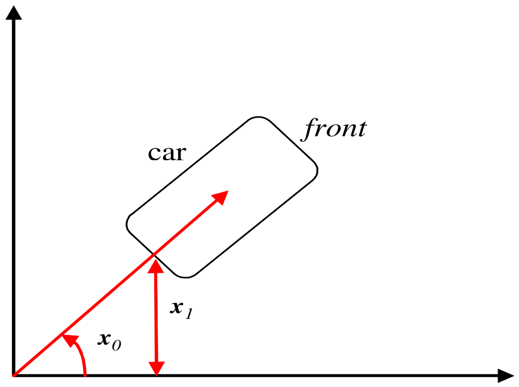


Figure 3.8: Coordinate system used to describe the car position and orientation.

This example has two parts. First, stability conditions are derived for a system that utilizes an IT2 TS FLCS in controller design. Second, the performance of the designed IT2 TS FLCS is compared with its T1 counterpart.

(a) Stability

As mentioned earlier, a T1 TS model is a valid approximation for the plant. Moreover, in order to make an unbiased comparison of the performance of T1 TS FLCS and IT2 TS FLCS, the plant is considered as a T1 TS model and only the controller is redesigned.

Plant and control rules are given as:

Plant rules: (see Figure 3.9 for antecedent MFs)[8]

[8] x_0 is given in radians.

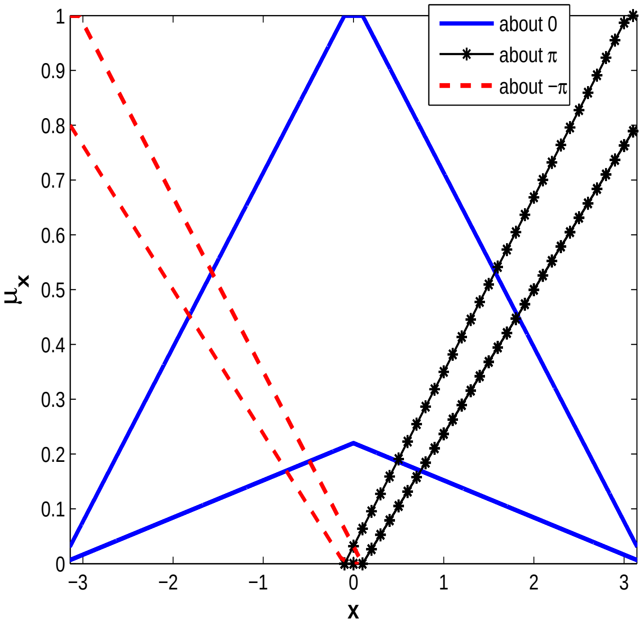


Figure 3.9: Coordinate system used to describe the car position and orientation.

Rule 1: If $x_0(k)$ is "about 0", then $\boldsymbol{x}(k+1) = \boldsymbol{A}_1\boldsymbol{x}(k) + \boldsymbol{b}_1\boldsymbol{u}(k)$

Rule 2: If $x_0(k)$ is "about π or $-\pi$", then $\boldsymbol{x}(k+1) = \boldsymbol{A}_2\boldsymbol{x}(k) + \boldsymbol{b}_2\boldsymbol{u}(k)$

Control rules:

Control Rule 1: If $x_0(k)$ is "about 0", then $u(k) = \boldsymbol{f}_1\boldsymbol{x}(k)$

Control Rule 2: If $x_0(k)$ is "about π or $-\pi$ ", then $u(k) = \boldsymbol{f}_2\boldsymbol{x}(k)$

in which

$$\boldsymbol{A}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \; \boldsymbol{A}_2 = \begin{bmatrix} 1 & 0 \\ 0.003183 & 1 \end{bmatrix} \tag{3.139}$$

$$\boldsymbol{b}_1 = \begin{bmatrix} 0.357143 \\ 1 \end{bmatrix}, \; \boldsymbol{b}_2 = \begin{bmatrix} 0.357143 \\ 1 \end{bmatrix} \tag{3.140}$$

$$\boldsymbol{f}_1 = \begin{bmatrix} -0.4212 & -0.02933 \end{bmatrix}, \; \boldsymbol{f}_2 = \begin{bmatrix} -0.0991 & -0.00967 \end{bmatrix} \tag{3.141}$$

Note that in order to make an unbiased comparison, $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are adopted from [141], where they were obtained by a pole-placement method. Similar to Example 1, the bounds for m' and n' are obtained as $[-6.207, 1.275]$ and $[-2.235, 4.935]$, respectively. In this example, $r = 2$ and hence the number of LMIs to be satisfied is 5. Using (3.114), the $\boldsymbol{H}_i$

matrices are calculated as

$$\begin{aligned}\boldsymbol{H}_1 &= \begin{bmatrix} 0.8495 & -0.0105 \\ 1 & 1 \end{bmatrix}, \ \boldsymbol{H}_2 = \begin{bmatrix} 0.9071 & -0.0070 \\ 1 & 1 \end{bmatrix} \\ \boldsymbol{H}_3 &= \begin{bmatrix} 0.9071 & -0.0070 \\ 0.5016 & 1 \end{bmatrix}, \ \boldsymbol{H}_4 = \begin{bmatrix} 0.9071 & -0.0070 \\ 0.0032 & 1 \end{bmatrix}, \ \boldsymbol{H}_5 = \begin{bmatrix} 0.9646 & -0.0035 \\ 1 & 1 \end{bmatrix}\end{aligned} \tag{3.142}$$

From which it follows that, $\boldsymbol{P}$, computed from the Matlab LMI toolbox, is

$$\boldsymbol{P} = \begin{bmatrix} 699.6386 & 57.3766 \\ 57.3766 & 11.7997 \end{bmatrix} \tag{3.143}$$

It can be verified that $\boldsymbol{P}$ satisfies the stability conditions for all 5 LMIs in (3.118), i.e.,

$$\begin{aligned}\boldsymbol{H}_1^T\boldsymbol{P}\boldsymbol{H}_1 - \boldsymbol{P} &= \begin{bmatrix} -85.369 & -3.659 \\ -3.659 & -1.125 \end{bmatrix} < \mathbf{0}, \ \boldsymbol{H}_2^T\boldsymbol{P}\boldsymbol{H}_2 - \boldsymbol{P} = \begin{bmatrix} -8.077 & 1.650 \\ 1.650 & -0.765 \end{bmatrix} < \mathbf{0} \\ \boldsymbol{H}_3^T\boldsymbol{P}\boldsymbol{H}_3 - \boldsymbol{P} &= \begin{bmatrix} -68.788 & -4.033 \\ -4.033 & -0.765 \end{bmatrix} < \mathbf{0}, \ \boldsymbol{H}_4^T\boldsymbol{P}\boldsymbol{H}_4 - \boldsymbol{P} = \begin{bmatrix} -123.637 & -9.714 \\ -9.714 & -0.765 \end{bmatrix} < \mathbf{0} \\ \boldsymbol{H}_5^T\boldsymbol{P}\boldsymbol{H}_5 - \boldsymbol{P} &= \begin{bmatrix} -48.296 & -4.325 \\ -4.325 & -0.3880 \end{bmatrix} < \mathbf{0}\end{aligned} \tag{3.144}$$

Therefore, the closed-loop system is asymptotically stable.

(b) Performance Evaluation

Here, the performance of T1 TS FLC and the IT2 TS FLC given in part (a) of this example are compared. The rules and consequent parameters of T1 TS are kept the same as in part (a) and only the MFs of the antecedents for the proposed T2 inference engine are redesigned from T1 to IT2. The MFs are shown in Figure 3.9, and the initial conditions are $x_0(0) = \pi$, and $x_1(0) = 20$. To make the comparison more realistic, a steering angle threshold has been set to $\pm\frac{\pi}{3}$ which is the threshold of a steering angle of a typical car.

Figure 3.10 compares the performance of the T1 TS and IT2 TS systems. Observe that the T1 TS controller has a noticeable overshoot[9] and a slower convergence to the set point in comparison to the IT2 TS controller. Clearly, the IT2 TS controller has less overshoot. It is easy to see that the IT2 TS controller has a faster settling time than the T1 TS. Moreover, the T1 TS controller has a noticeably undesirable undershoot between $t = 57$ and $t = 80\,\text{sec}$, where the IT2 TS controller does not.

[9] 'Overshoot' occurs when the system's response overshoots the starting trajectory (initial position), and 'undershoot' occurs when the system's response undershoots the zero trajectory.

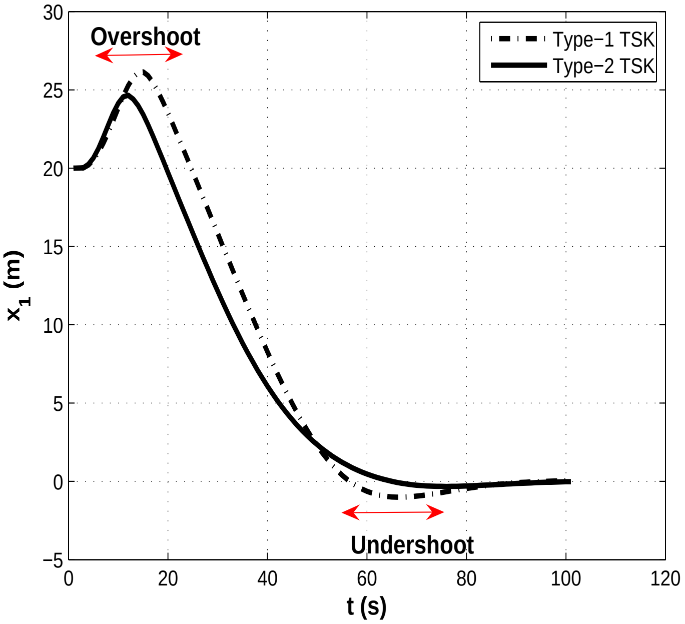


Figure 3.10: Trajectories of the car model for the two controllers.

Figure 3.11 compares the angular positions of the two controllers. Angular position with the IT2 TS controller has a slightly larger negative slope that helps the car to get to $x_0 = 0$ and $x_1 = 0$ faster. This is noticeable by comparing the time required for both controllers to reach the final angle, i.e., 0°.

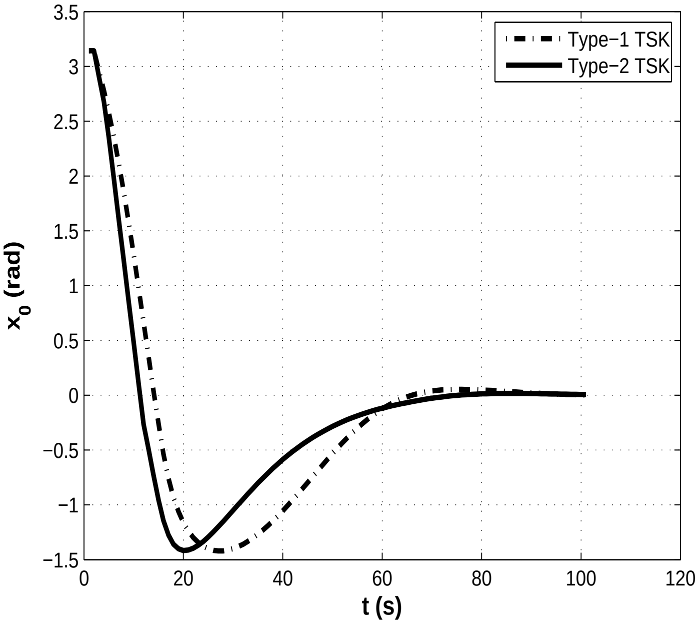


Figure 3.11: Angular position of the car for T1 and T2 fuzzy controllers.

To further explain this, the control efforts (steering angles) of both controllers are compared in Figure 3.12. As can be seen, initially, the control effort of the IT2 TS controller is significantly greater than the T1 TS controller. This results in less overshoot in the transient response of the IT2 TS controller. Additionally, the IT2 TS control effort reveals a large slope in decreasing the steering angle, rendering less undershoot compared to the T1 TS controller. This is attributed to the shape of the T2 MFs that allows more uncertainty in the controller structure.

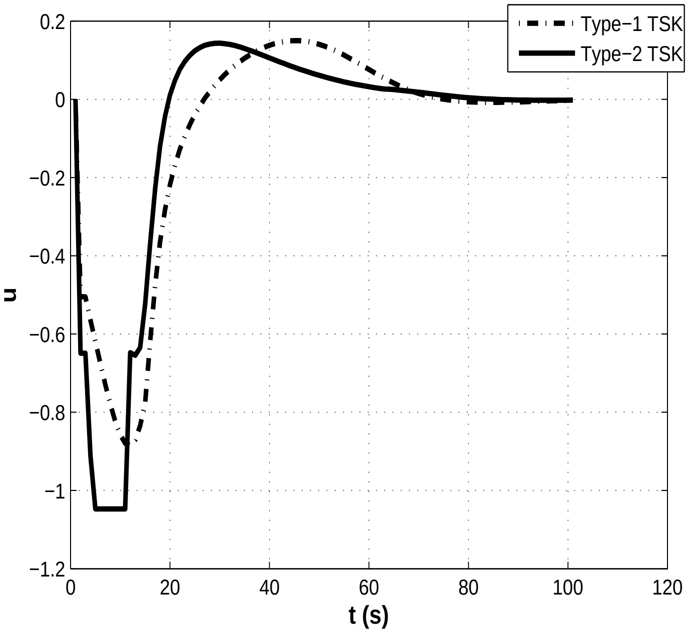


Figure 3.12: Controller outputs for T1 and T2 fuzzy systems.

This example demonstrates that the proposed IT2 TS with the proposed inference, $Y_{\mathrm{TSK/NEW}}$, is capable of outperforming a well-tuned T1 TS FLCS.

Case study d (control of a nonlinear/chaotic system)

In the second part of this example, an IT2 TS FLC is developed and is applied to another popular nonlinear system. This case study shows that the IT2 controller can stabilize a chaotic system while simultaneously achieving enhanced results compared to its T1 counterpart. The system considered in this problem is an electrical circuit, known as Chua's circuit [143], that exhibits chaotic output behaviors. The Chua's circuit consists of one inductor (L), two capacitors (C_1, C_2), one linear resistor (R), and one piece-wise linear resistor ($g(v_{c1})$). This circuit is described by the following equations [144]:

$$\dot{v}_{c1} = \frac{1}{C_1}\left(\frac{1}{R}(v_{c2} - v_{c1}) - g(v_{c1})\right) + u_1 \tag{3.145}$$

$$\dot{v}_{c2} = \frac{1}{C_2}\left(\frac{1}{R}(v_{c1} - v_{c2}) + i_L\right) + u_2 \tag{3.146}$$

$$\dot{i}_L = \frac{1}{L}(-v_{c2} - R_0 i_L) + u_3 \tag{3.147}$$

where $g(v_{c1})$ is given by

$$g(v_{c1}) = \begin{cases} G_b v_{c1} + (G_a - G_b)E & v_{c1} \geq E \\ G_a v_{c1} & -E < v_{c1} < E \\ G_b v_{c1} - (G_a - G_b)E & v_{c1} \leq -E \end{cases} \tag{3.148}$$

where v_{c1}, v_{c2}, i_L are state variables, G_a, G_b, E are the characteristics of the piece-wise linear resistor, and u_1, u_2, u_3 are the control inputs. For more information about the Chua's circuit see [143] and [145]. It is well-known that Chua's circuit has nonlinear dynamics such as bifurcation and chaos. Chaotic systems can reveal large oscillations/motions and hence there is a growing interest in the controllers that can effectively handle such systems. Wang and Tanaka in [144] developed a T1 TS model that represents this circuit well. Hence, similar to previous examples, the plant is kept as a T1 TS and the controller is replaced with the proposed IT2 TS FLC. To make an unbiased comparison, only the IT2 MFs are redesigned while keeping all other parameters the same for both control systems. Let $\boldsymbol{x}(t) \equiv [x_1(t), x_2(t), x_3(t)]^T$ where $x_i(t)$'s are the state variables, i.e., $x_1 = v_{c1}$, $x_2 = v_{c2}$, $x_3 = i_L$. The values for the parameters used in this example are: $R = 1.4286, R_0 = 0\Omega, C_1 = 0.1, C_2 = 0.2, L = 0.1429, G_a = -2, G_b = 0.1$, and $E = 1$.

Plant and control rules are given as:

Plant rules:

Rule 1: If $x_1(t)$ is M_1, then $\dot{\boldsymbol{x}} = \boldsymbol{A}_1\boldsymbol{x}(t) + \boldsymbol{b}\boldsymbol{u}(t)$

Rule 2: If $x_1(t)$ is M_2, then $\dot{\boldsymbol{x}} = \boldsymbol{A}_2\boldsymbol{x}(t) + \boldsymbol{b}\boldsymbol{u}(t)$

Figure 3.13 shows the antecedent MFs M_1 and M_2.

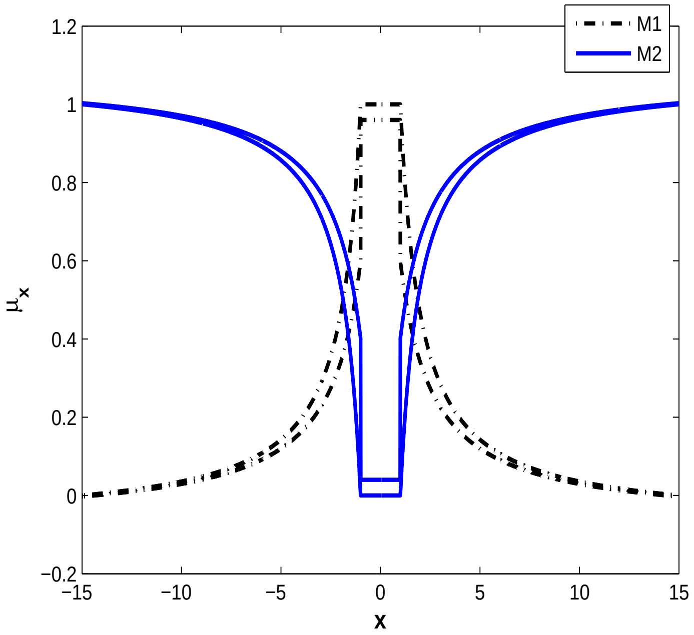


Figure 3.13: IT2 MFs for Example 2, case study d.

Control rules:

Control Rule 1: If $x_1(t)$ is M_1, then $\boldsymbol{u}(t) = \boldsymbol{F}_1\boldsymbol{x}(t)$

Control Rule 2: If $x_1(t)$ is M_2, then $\boldsymbol{u}(t) = \boldsymbol{F}_2\boldsymbol{x}(t)$

in which

$$\begin{aligned}\boldsymbol{A}_1 &= \begin{bmatrix} 5.7143 & 14.2857 & 0 \\ 0.7143 & -0.7143 & 0.5 \\ 0 & -7 & 0 \end{bmatrix}, \ \boldsymbol{A}_2 = \begin{bmatrix} -12.0190 & 14.2857 & 0 \\ 0.7143 & -0.7143 & 0.5 \\ 0 & -7 & 0 \end{bmatrix} \\ \boldsymbol{F}_1 &= \begin{bmatrix} -33.3333 & -31.6202 & -1.7961 \\ 24.2702 & 0.0167 & -1.9808 \\ 1.7961 & 8.4808 & -0.3333 \end{bmatrix}, \ \boldsymbol{F}_2 = \begin{bmatrix} 3.0667 & 21.4379 & -3.7158 \\ -28.7879 & 0.0167 & -20.2722 \\ 3.7158 & 26.7722 & -0.3333 \end{bmatrix}\end{aligned} \tag{3.149}$$

and $\boldsymbol{b}$ is a 3 by 3 identity matrix. $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ are adopted from [144]. It is very easy to show that for $m' = n' = 0.8$, the following $\boldsymbol{P}$ matrix satisfies the LMIs in (3.137):

$$\boldsymbol{P} = \begin{bmatrix} 2.2240 & 0.0112 & 0.1701 \\ 0.0112 & 2.5747 & -0.0198 \\ 0.1701 & -0.0198 & 2.0743 \end{bmatrix} \tag{3.150}$$

Figure 3.14 illustrates the response of the Chua's circuit for the duration of 50 sec when both T1 and IT2 controllers are applied. The initial conditions considered for simulations are $\boldsymbol{x}(0) = [1, 0, 0]^T$.

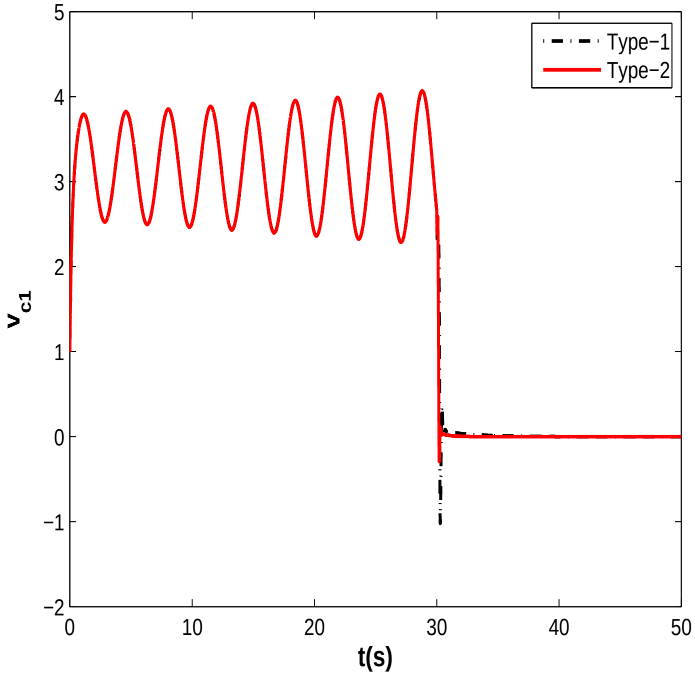


Figure 3.14: Chua's circuit response to the T1 and IT2 controllers.

Note that the controllers are invoked at $t = 30\,\text{sec}$. Before the controllers are activated, the system's output is oscillating. As shown in Figure 3.14, both controllers stabilize the system. However, examining the performance after 30 sec using Figure 3.15 (enlarged plot of the controllers output in Figure 3.14 when $t \geq 30\,\text{sec}$), it is easy to observe that the IT2 controller has a much better transient response. The rise and settling times for the IT2 and T1 controllers, respectively, are $t_r = 0.15\,\text{sec}$, $t_s = 0.19\,\text{sec}$, and $t_r = 0.22\,\text{sec}$ and $t_s = 0.25\,\text{sec}$. Moreover, the IT2 controller produces much less overshoot compared to the T1 controller when $30\,\text{sec} \leq t \leq 31\,\text{sec}$.

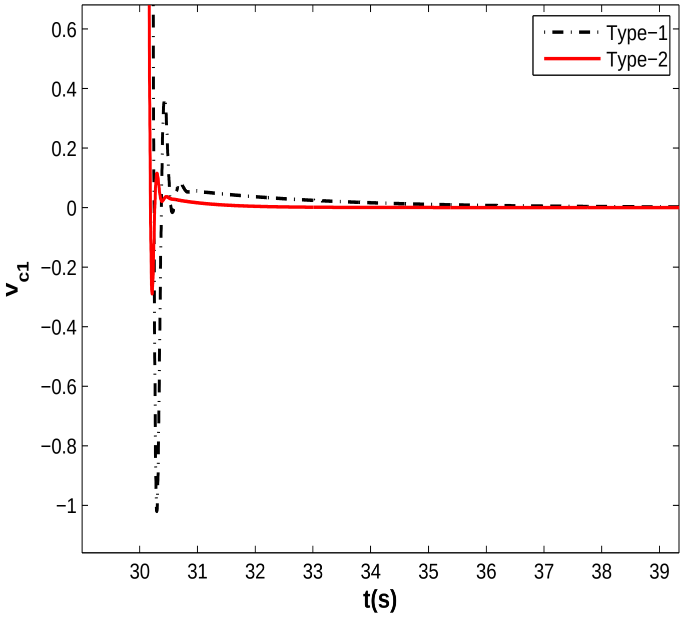


Figure 3.15: Chua's circuit response to the T1 and IT2 controllers (controllers are invoked at $t = 30\,\mathrm{sec}$).

3.6 Conclusion

In this chapter, novel inference engines were proposed to facilitate the stability and design of IT2 FLCs. More specifically, the WM UBs were modified to develop the most general inference engine, amongst the proposed engines, for IT2 TSK A2-C0 and IT2 TS FLSs. The inference engine was formulated in closed-form and hence does not require using the the iterative KM algorithms. Due to the simple structure of the proposed inference engine, it can be adopted to design IT2 TSK and IT2 TS FLCSs for real-time control applications. Using the proposed inference mechanism, stability conditions in terms of LMIs for IT2 TSK FLCSs and IT2 TS FLCSs were derived and transformed into the standard formats that can be easily solved using software tools such as the Matlab LMI toolbox. Consequently, the stability of a control system can be proved analytically when the proposed inference mechanism is used to design both IT2 TSK FLCSs and IT2 TS FLCSs. To evaluate the performance of IT2 TS FLCSs with the proposed inference engine in control and tracking applications, two benchmark examples were adopted from the literature. It was shown that a well-tuned IT2 TS FLCS has the potential to outperform its T1 counterpart, and that is because IT2 FLCs have a more flexible structure than T1. Finally, using the proposed model for IT2 TSK A2-C0 or IT2 TS FLSs will enable control engineers to design and implement stable IT2 FLCSs with enhanced performance.

In the next chapter, the proposed inference mechanism for the design of an IT2 TSK FLC is used in tracking control applications of MRRs.

Chapter 4

Design of Novel IT2 TSK FLCs with Applications to Robot Manipulators

This chapter presents a novel adaptive control design paradigm of IT2 TSK FLCs. Specifically, a design methodology is developed for the tracking control of MRRs with uncertain dynamic parameters. Experiments are performed to verify the effectiveness of the proposed approach. This chapter is organized as follows: Section 4.1 provides background on MRR and available techniques for their control. Section 4.2 presents the IT2 TSK FLC structure. Section 4.3 reviews the governing equations for MRRs and presents the control design for tracking purposes. Section 4.4 presents the experimental results of the proposed controller on an MRR, and finally, Section 4.5 draws conclusions.

4.1 Introduction

Based on the literature survey conduced, no prior work has been published that provides a systematic method for the design of IT2 TSK FLCs for MRRs. Therefore, a novel IT2 TSK FLC is developed for tracking applications of MRR. The contributions of this chapter lie in: 1) a rigorous design paradigm of novel IT2 TSK FLCs for MRRs, 2) validation of the performance of the proposed control strategy experimentally, and 3) comparison of the proposed control methodology with some well-known controllers.

This chapter has two main parts. In the first part, an IT2 TSK FLC is designed for trajectory tracking. The consequent parameters as well as IT2 MFs are designed, thus eliminating the extra effort usually needed to define these functions. Moreover, by adopting the closed-form inference engine in [146], a computationally effective IT2 TSK FLCS is proposed that can be easily implemented in real-time control applications.

In the second part of the chapter, a serial MRR is considered with two degrees-of-freedom and present a novel control design methodology for such systems. An IT2 TSK FLC is rigorously designed with the guaranteed stability and the performance of the controller through experiments is validated.

4.2 Design of IT2 TSK FLCSs

To identify IT2 TSK FLCSs, rules, MFs, and TSK consequent parameters must be designed, all of which are described in this section.

4.2.1 Rule bases

Rules are one of the main components of any FLS. Obtaining suitable rules for FLCSs that can capture the behavior of the plant or controller is very crucial and not an easy exercise. Due to the several parameters needed to characterize a FLCS, there is no general and yet systematic method to effectively define the rules. Designers usually exploit expert knowledge or sometimes use their intuition to define the rules for their model. However, it was shown that MacVicar-Whelan rule base can be effectively adopted for tracking design problems in FLCSs [147]. The rule structure for an IT2 TSK FLC is as follows:

$$\text{If } e \text{ is } \widetilde{F}_1^i \text{ and } \dot{e} \text{ is } \widetilde{F}_2^i\text{, Then } u_i = a_1^i e + a_2^i \dot{e} \tag{4.1}$$

where $i = 1, ..., M$, $\widetilde{F}_j^i$ represents the IT2 FS of input state j in rule i, a_1^i and a_2^i are the coefficients of the output function for rule i (and hence are crisp numbers, i.e., type-0 FSs), u_i is the output of the ith rule for the two controllers, respectively, and M is the number of rules. Error, e, and its rate of change, $\dot{e}$, are given as

$$e \equiv r - y \tag{4.2}$$

$$\dot{e} = \frac{de}{dt} \tag{4.3}$$

where r is the set point and y is the output of the closed-loop system. The above rules allow us to model the uncertainties encountered in the antecedents. Lower and upper firing strengths of the ith rule, $\underline{f}^i$ and $\overline{f}^i$, are given by [1]

$$\underline{f}^i(\boldsymbol{e}) = \underline{\mu}_{\widetilde{F}_1^i}(e) * \underline{\mu}_{\widetilde{F}_2^i}(\dot{e}) \tag{4.4}$$

$$\overline{f}^i(\boldsymbol{e}) = \overline{\mu}_{\widetilde{F}_1^i}(e) * \overline{\mu}_{\widetilde{F}_2^i}(\dot{e}) \tag{4.5}$$

where $\underline{\mu}_{\widetilde{F}_j^i}$ and $\overline{\mu}_{\widetilde{F}_j^i}$ represent the jth $(j = 1, 2)$ lower and upper antecedent MFs of rule i, respectively, and "$*$" is a t-norm operator. Error vector, $\boldsymbol{e}$, is defined as

$$\boldsymbol{e} = [e, \dot{e}]^T \tag{4.6}$$

Using the inference engine introduced in [146], the controller output, u, is given as

$$u \;=\; m\frac{\sum_{i=1}^{M} \underline{f}^i\left(\boldsymbol{e}\right) u_i}{\sum_{i=1}^{M} \underline{f}^i\left(\boldsymbol{e}\right)} + n\frac{\sum_{i=1}^{M} \overline{f}^i\left(\boldsymbol{e}\right) u_i}{\sum_{i=1}^{M} \overline{f}^i\left(\boldsymbol{e}\right)} \tag{4.7}$$

where m and n are the controller parameters. Note that T1 FLSs are a **special case** of T2 FLSs. When the FOUs of IT2 MFs are removed, they become T1 MFs. Thus, removing the FOUs and letting $m + n = 1$ in the inference engine, (4.7), will turn the designed IT2 TSK FLC into a T1 TSK FLC, i.e., the IT2 TSK FLC structure in (4.7) is a generalized form of T1 TSK FLCs. Hence, if the designer has the freedom to select m and n producing the best possible performance (not necessarily $m + n = 1$), then it is expected that the IT2 TSK FLC outperforms its T1 counterpart. This fact will be also verified during the experimental analyses.

A general MacVicar-Whelan rule-base ([148], [149]) uses error, e, and change in error, Δe, to determine u for IT2 TSK-PD. This rule-base has 9 rules and is defined in Table 4.1 where three linguistic variables for the states of e and $\dot{e}$, "NB," "ZE," and "PB," represent "negative big," "zero," and "positive big," respectively.

Table 4.1: Fuzzy rule-base for a system with 9 rules.

$\dot{e}/e$	**NB**	**ZE**	**PB**
NB	**NB**	**NB**	**ZE**
ZE	**NB**	**ZE**	**PB**
PB	**ZE**	**PB**	**PB**

4.2.2 MFs

Figure 4.1 shows the proposed T1 and IT2 MFs for systems with 9 rules. The 'x' axis is a normalized representation of e or $\dot{e}$. The IT2 MFs are based on the generalization of T1 MFs used in [147]. The IT2 MFs have been selected to have comparable parameters to their T1 counterparts hence making a fair comparison between the two systems.

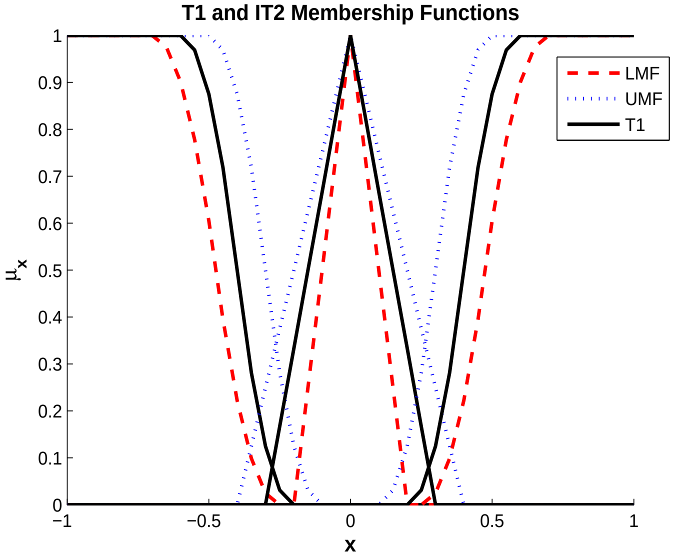


Figure 4.1: MFs for e and $\dot{e}$ of the proposed IT2 and T1 TSK FLCS for a system with 9 rules.

4.2.3 Initial TSK consequent parameters

To trigger the IT2 TSK FLC, the *initial* values of the TSK consequent parameters must be determined (in Section 4.3.3 it is explained how these parameters are adjusted in a real-time process to achieve a good tracking performance); therefore, in this subsection, a straightforward approach is presented for determining the initial TSK consequent parameters of the proposed IT2 TSK FLC. Note that there is no unique method to find the consequent parameters, i.e., a_1^i's and a_2^i's for every rule in (4.1); hence, to make the most efficient use of the rule-base system, conditions are considered where e_n and Δe_n correspond exactly to the linguistic terms defined in the rule-base given in Table 4.1, i.e., to 'NB,' 'ZE,' and 'PB':

1. If e is NB ($e \longrightarrow -1$), then
   Rule 1. If Δe is NB, $\Delta e \longrightarrow -1$: $u_1 = -a_1^1 - a_2^1 \longrightarrow -1$

   Rule 2. If Δe is ZE, $\Delta e \longrightarrow 0$: $u_2 = -a_1^2 \longrightarrow -1$

   Rule 3. If Δe is PB, $\Delta e \longrightarrow 1$: $u_3 = -a_1^3 + a_2^3 \longrightarrow 0$

2. If e_n is ZE ($e_n \longrightarrow 0$), then
   Rule 4. If Δe_n is NB, $\Delta e_n \longrightarrow -1$: $u_4 = -a_2^4 \longrightarrow -1$

   Rule 5. If Δe_n is ZE, $\Delta e_n \longrightarrow 0$: $u_5 \longrightarrow 0$

Rule 6. If Δe_n is PB, $\Delta e_n \longrightarrow 1$: $u_6 = a_2^6 \longrightarrow 1$

3. If e is PB ($e \longrightarrow 1$), then
   Rule 7. If Δe is NB, $\Delta e \longrightarrow -1$: $u_7 = a_1^7 - a_2^7 \longrightarrow 0$

   Rule 8. If Δe is ZE, $\Delta e \longrightarrow 0$: $u_8 = a_1^8 \longrightarrow 1$

   Rule 9. If Δe is PB, $\Delta e \longrightarrow 1$: $u_9 = a_1^9 + a_2^9 \longrightarrow 1$

where '$a \longrightarrow b$' denotes 'a' approaches/equals 'b'. From the above 9 conditions, one can solve for a_1^i and a_2^i, $i = 1, \cdots, 9$, to find the consequent parameters of the IT2 TSK FLC. It is worth noting that the above constraints do not *explicitly* determine all of the TSK parameters. For instance, a_1^1, a_2^1, a_1^4, a_1^6, and a_2^8 cannot be determined. As a result, these parameters are restricted to be in $[0, 1]$ and allow designers the freedom to choose their values within this range to satisfy the required design criteria. For instance, in Table 4.2, some arbitrary values in $[0, 1]$ were selected for those parameters that can not be explicitly determined. All the TSK consequent parameters obtained by the aforementioned method are summarized in Table 4.2.

Table 4.2: TSK consequent parameters for a system with 9 rules.

a_1^1	a_1^2	a_1^3	a_1^4	a_1^5	a_1^6	a_1^7	a_1^8	a_1^9
0.1	1	0.5	0.5	0.3	0.2	0.9	1	0.8
a_2^1	a_2^2	a_2^3	a_2^4	a_2^5	a_2^6	a_2^7	a_2^8	a_2^9
0.9	0.01	0.5	1	0.9	1	0.9	0	0.2

In this chapter, the controller structure in (4.7) with fixed parameters is referred to as "**fixed-parameters**." Using the proposed TSK consequent parameters in Table 4.2, the designer only needs to tune the controller parameters, m and n. The advantage of the fixed-parameters controller over a linear controller such as PD is that it can handle the nonlinear dynamics of an MRR to a certain degree. Yet, the parameters of this controller need to be re-tuned by the designer every time the MRR is reconfigured, and this need is seen as the limitation of the fixed-parameters controller. To handle an MRR with unmodeled dynamics and changing parameters, a control system is needed that is capable of adjusting/tuning its parameters to be able to cope with the resulting uncertainties, i.e., a controller with varying parameters. For mathematical derivation of this controller with adjustable parameters, first some background on MRRs and their dynamics are introduced. The structure of this controller will then be derived in Section 4.3.3.

4.3 Robot Manipulators

In this section, the dynamics of robot manipulators (RMs) is reviewed. First, fundamental properties of RMs, needed for control design, are provided. Next, the tracking problem in RMs are defined and the assumptions made to achieve this objective are presented. The dynamics of robot manipulators with p-link arms can be written as [100]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{V}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_d = \boldsymbol{\tau} \tag{4.8}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{p\times p}$ is the inertia matrix, $\boldsymbol{V}_m(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{p\times p}$ is the Coriolis matrix, $\boldsymbol{F}(\dot{\boldsymbol{q}}) \in \mathbb{R}^p$ contains the friction terms, $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^p$ is the gravity vector, $\boldsymbol{\tau}_d \in \mathbb{R}^p$ represents disturbances, $\boldsymbol{\tau} \in \mathbb{R}^p$ is the torque vector or control input, and $\boldsymbol{q} \in \mathbb{R}^p$ is the joint variable.

RM Properties

The fundamental properties of RMs are [101]:

1. $\boldsymbol{M}(\boldsymbol{q})$ is a symmetric, positive-definite matrix and bounded, i.e., $\|\boldsymbol{M}(\boldsymbol{q})\| \leq M_B$ where M_B is a constant.
2. $\boldsymbol{V}_m(\boldsymbol{q},\dot{\boldsymbol{q}})$ is bounded, i.e., $\|\boldsymbol{V}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\| < V_B \|\dot{\boldsymbol{q}}\|$ with V_B being a constant.
3. The matrix $\boldsymbol{M}(\boldsymbol{q}) - 2\boldsymbol{V}_m(\boldsymbol{q},\dot{\boldsymbol{q}})$, is skew-symmetric.
4. $\boldsymbol{F}(\dot{\boldsymbol{q}})$ and $\boldsymbol{G}(\boldsymbol{q})$ are bounded, i.e., $\|\boldsymbol{F}(\dot{\boldsymbol{q}})\| \leq F_B \|\dot{\boldsymbol{q}}\| + K_B$ and $\|\boldsymbol{G}(\boldsymbol{q})\| \leq G_B$ (F_B, K_B, and G_B are constants).
5. The disturbances are bounded, i.e., $\|\boldsymbol{\tau}_d\| \leq D_B$.

4.3.1 Tracking Problem

In robot position control, the objective is to make the RM follow a prescribed designed trajectory, i.e., $\boldsymbol{q}_d$. The tracking problem is then defined as finding a control input, $\boldsymbol{\tau}$, to achieve this objective. First, define tracking error, $\boldsymbol{e}$, and filtered tracking error, $\boldsymbol{r}$, as

$$\boldsymbol{e} \equiv \boldsymbol{q}_d - \boldsymbol{q} \tag{4.9}$$

$$\boldsymbol{r} \equiv \dot{\boldsymbol{e}} + \boldsymbol{\Lambda}\boldsymbol{e} \tag{4.10}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times p}$ is a positive-definite design matrix. $\boldsymbol{\Lambda}$ is usually selected to be diagonal with positive elements.

Using the norm properties, it follows that

$$\begin{aligned} \|\boldsymbol{r}\| &\geq \|\dot{\boldsymbol{e}}\| & (4.11) \\ \|\boldsymbol{r}\| &\geq \|\boldsymbol{\Lambda e}\| & (4.12) \end{aligned}$$

It is easy to re-write (4.10) as

$$\boldsymbol{\Lambda}^{-1}\boldsymbol{r} = \boldsymbol{\Lambda}^{-1}\dot{\boldsymbol{e}} + \boldsymbol{e} \tag{4.13}$$

Thus

$$\left\|\boldsymbol{\Lambda}^{-1}\boldsymbol{r}\right\| \geq \|\boldsymbol{e}\| \tag{4.14}$$

Using the properties of matrix norms and knowing that $\boldsymbol{\Lambda}$ is positive-definite, the left-hand side of (4.13) can be expressed as

$$\left\|\boldsymbol{\Lambda}^{-1}\boldsymbol{r}\right\| \leq \left\|\boldsymbol{\Lambda}^{-1}\right\| \|\boldsymbol{r}\| = \frac{\|\boldsymbol{r}\|}{\min(eig(\boldsymbol{\Lambda}))} \tag{4.15}$$

where $eig(\boldsymbol{\Lambda})$ represents the eigenvalues of $\boldsymbol{\Lambda}$. Hence, from (4.14) and (4.15) it is concluded that

$$\frac{\|\boldsymbol{r}\|}{\min(eig(\boldsymbol{\Lambda}))} \geq \|\boldsymbol{e}\| \tag{4.16}$$

The inequalities in (4.11) and (4.16) are exploited in the next section. Solving for $\boldsymbol{q}$ from (4.9) and $\dot{\boldsymbol{e}}$ from (4.10), and inserting their first and second derivatives in (4.8), the dynamics of the RM can be written as[1]

$$\begin{aligned} \boldsymbol{M}\dot{\boldsymbol{r}} &= \boldsymbol{M}\left[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\,\dot{\boldsymbol{e}}\right] + \boldsymbol{V}_m\left[\dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda e}\right] - \boldsymbol{V}_m\boldsymbol{r} + \boldsymbol{F}(\dot{\boldsymbol{q}}) \\ &\quad + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_d - \boldsymbol{\tau} \end{aligned} \tag{4.17}$$

Alternatively, (4.17) is expressed as follows:

$$\boldsymbol{M}\dot{\boldsymbol{r}} = -\boldsymbol{V}_m\boldsymbol{r} + \boldsymbol{f} + \boldsymbol{\tau}_d - \boldsymbol{\tau} \tag{4.18}$$

where $\boldsymbol{f}$ is given by

$$\boldsymbol{f} = \boldsymbol{M}\left[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\,\dot{\boldsymbol{e}}\right] + \boldsymbol{V}_m\left[\dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda e}\right] + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) \tag{4.19}$$

Note that $\boldsymbol{f}$ contains nonlinear terms as well as some unmodeled dynamics. For MRRs the dynamic parameters in (4.17)-(4.19) will change when a robot arm is reconfigured; hence, model-based control is not feasible for MRRs that can assume multitude of configuration. Therefore, those parameters are dealt with as unknowns regardless of the robot configuration. Section 4.3.2 deals with the control design for an MRR with dynamic parameters (of any assumed configurations) described by (4.18) and (4.19).

[1] $\boldsymbol{M}$ and $\boldsymbol{V}_m$ are short for $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{V}_m(\boldsymbol{q}, \dot{\boldsymbol{q}})$, respectively.

4.3.2 MRR Control Structure

Figure 4.2 shows the structure of IT2 TSK FLCS, where q_d is the reference trajectory. Error, e, is defined as the difference between r and y. The inputs to the IT2 TSK PD-FLCS are e_n and $\dot{e}_n$, and the output is u_{Fuzzy}.

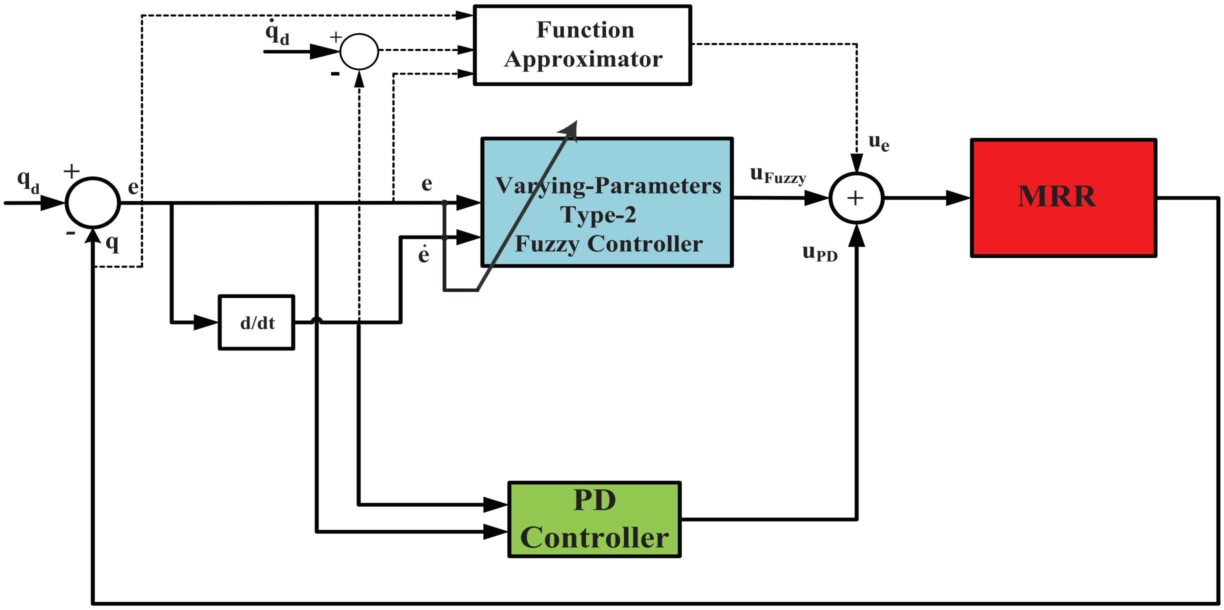


Figure 4.2: Controller structure for an MRR.

The following theorem is used later in the design of the IT2 TSK FLC:

Theorem 1 *[101] : If for a nonlinear system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{d}(t)$, there exists a Lyapunov function $V(x,t)$ with continuous partial derivatives such that for x in a compact set $\boldsymbol{S} \subset \mathbb{R}^n$*

$$V(x,t) > 0 \tag{4.20}$$

and

$$\dot{V}(x,t) < 0 \text{ for } \|\boldsymbol{x}\| > R \tag{4.21}$$

for some $R > 0$ such that the ball of radius of R is contained in $\boldsymbol{S}$, then the system is uniformly ultimately bounded (UUB), and the norm of the state is bounded to within a neighborhood of R.

Next, the MRR control structure is proposed to be of the form

$$\boldsymbol{\tau} = \boldsymbol{u}_{PD} + \boldsymbol{u}_{Fuzzy} + \boldsymbol{u}_e \tag{4.22}$$

where

$$\begin{aligned} \boldsymbol{u}_{PD} &= \boldsymbol{K}_{PD}\boldsymbol{r} && (4.23) \\ \boldsymbol{u}_{Fuzzy} &= \boldsymbol{g}(\boldsymbol{e}, \dot{\boldsymbol{e}}) && (4.24) \end{aligned}$$

where $\boldsymbol{u}_{PD}$ is the control effort of a PD controller, $\boldsymbol{u}_{Fuzzy}$ is the control effort of the IT2 TSK FLC (a function of $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$), $\boldsymbol{u}_e$ is an extra compensation torque that will be explained in Section 4.3.3, and $\boldsymbol{K}_{PD}$ is a design gain matrix of the linear controller. Lewis [101] showed that a PD controller can not handle the entire dynamics of an MRR by itself. Therefore, a nonlinear controller that can handle uncertainties and varying dynamics, must be used to result in acceptable tracking performances. While the PD controller is handling the linear dynamics, the IT2 TSK FLC is handling uncertainties, varying dynamics, and unknown parameters of the MRR. Note that the IT2 TSK FLC is designed to handle most of the nonlinearities and uncertainties embedded in $\boldsymbol{f}$ given by (4.19). The total control effort is the summation of each controller's output as given by (4.22).

In the following section, it is shown that if (4.22) is used to control the robot, under certain assumptions, UUB stability is guaranteed.

4.3.3 Design of Varying-Parameter Controller

In a decentralized control method, 'p' controllers are used (corresponding to 'p' links of an MRR) which control the motion of each joint. Using (4.7) as an inference engine of the control command for each individual joint, it is easy to show that the output of an IT2 TSK FLC for the entire MRR can be expressed as

$$\boldsymbol{u}_{Fuzzy} = \boldsymbol{g}(\boldsymbol{e}, \dot{\boldsymbol{e}}) = \boldsymbol{X\Theta} \tag{4.25}$$

where $\boldsymbol{X} \in \mathbb{R}^{p\times(2M)}$ and $\boldsymbol{\Theta} \in \mathbb{R}^{2M}$ are given by

$$\boldsymbol{X} \equiv (\underline{\boldsymbol{\phi}} + \overline{\boldsymbol{\phi}}) \tag{4.26}$$

$$\boldsymbol{\Theta} \equiv \left[\begin{array}{cc} a_1^1, \cdots, a_1^M & , a_2^1, \cdots, a_2^M \end{array} \right]^T \tag{4.27}$$

in which

$$\underline{\boldsymbol{\phi}} \equiv \begin{bmatrix} m_1(\frac{\underline{f}_1^1}{\sum_{i=1}^M \underline{f}_1^i})e_1 & \cdots & m_1(\frac{\underline{f}_1^M}{\sum_{i=1}^M \underline{f}_1^i})e_1 & m_1(\frac{\underline{f}_1^1}{\sum_{i=1}^M \underline{f}_1^i})\dot{e}_1 & \cdots & m_1(\frac{\underline{f}_1^M}{\sum_{i=1}^M \underline{f}_1^i})\dot{e}_1 \\ m_2(\frac{\underline{f}_2^1}{\sum_{i=1}^M \underline{f}_2^i})e_2 & \cdots & m_2(\frac{\underline{f}_2^M}{\sum_{i=1}^M \underline{f}_2^i})e_2 & m_2(\frac{\underline{f}_2^1}{\sum_{i=1}^M \underline{f}_2^i})\dot{e}_2 & \cdots & m_2(\frac{\underline{f}_2^M}{\sum_{i=1}^M \underline{f}_2^i})\dot{e}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_p(\frac{\underline{f}_p^1}{\sum_{i=1}^M \underline{f}_p^i})e_p & \cdots & m_p(\frac{\underline{f}_p^M}{\sum_{i=1}^M \underline{f}_p^i})e_p & m_p(\frac{\underline{f}_p^1}{\sum_{i=1}^M \underline{f}_p^i})\dot{e}_p & \cdots & m_p(\frac{\underline{f}_p^M}{\sum_{i=1}^M \underline{f}_p^i})\dot{e}_p \end{bmatrix} \tag{4.28}$$

and,

$$\overline{\boldsymbol{\phi}} \equiv \begin{bmatrix} n_1(\frac{\overline{f}_1^1}{\sum_{i=1}^M \overline{f}_1^i})e_1 & \cdots & n_1(\frac{\overline{f}_1^M}{\sum_{i=1}^M \overline{f}_1^i})e_1 & n_1(\frac{\overline{f}_1^1}{\sum_{i=1}^M \overline{f}_1^i})\dot{e}_1 & \cdots & n_1(\frac{\overline{f}_1^M}{\sum_{i=1}^M \overline{f}_1^i})\dot{e}_1 \\ n_2(\frac{\overline{f}_2^1}{\sum_{i=1}^M \overline{f}_2^i})e_2 & \cdots & n_2(\frac{\overline{f}_2^M}{\sum_{i=1}^M \overline{f}_2^i})e_2 & n_2(\frac{\overline{f}_2^1}{\sum_{i=1}^M \overline{f}_2^i})\dot{e}_2 & \cdots & n_2(\frac{\overline{f}_2^M}{\sum_{i=1}^M \overline{f}_2^i})\dot{e}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n_p(\frac{\overline{f}_p^1}{\sum_{i=1}^M \overline{f}_p^i})e_p & \cdots & n_p(\frac{\overline{f}_p^M}{\sum_{i=1}^M \overline{f}_p^i})e_p & n_p(\frac{\overline{f}_p^1}{\sum_{i=1}^M \overline{f}_p^i})\dot{e}_p & \cdots & n_p(\frac{\overline{f}_p^M}{\sum_{i=1}^M \overline{f}_p^i})\dot{e}_p \end{bmatrix} \tag{4.29}$$

Substituting (4.23) and (4.25) into (4.18) yields

$$\boldsymbol{M}\dot{\boldsymbol{r}} = -\boldsymbol{V}_m\boldsymbol{r} - \boldsymbol{K}_{PD}\boldsymbol{r} - \boldsymbol{X}\boldsymbol{\Theta} - \boldsymbol{u}_e + \boldsymbol{f} + \boldsymbol{\tau}_d \tag{4.30}$$

One can decompose $\boldsymbol{f}$ into $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ such that $\boldsymbol{f} = \boldsymbol{f}_1 + \boldsymbol{f}_2$ where

$$\boldsymbol{f}_1 = \boldsymbol{M}\,[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\,\dot{\boldsymbol{e}}] + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) \tag{4.31}$$

$$\boldsymbol{f}_2 = \boldsymbol{V}_m\,[\dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\boldsymbol{e}] \tag{4.32}$$

In this decomposition, the Coriolis/centripetal term through a nonlinear function approximator is estimated. The IT2 TSK FLC handles the nonlinearities and uncertainties in $\boldsymbol{f}_1$. Hence, (4.30) can be rewritten as

$$\boldsymbol{M}\dot{\boldsymbol{r}} = -\boldsymbol{V}_m\boldsymbol{r} + \boldsymbol{f}_1 + \boldsymbol{V}_m\,[\dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\boldsymbol{e}] + \boldsymbol{\tau}_d - \boldsymbol{K}_{PD}\boldsymbol{r} - \boldsymbol{X}\boldsymbol{\Theta} - \boldsymbol{u}_e \tag{4.33}$$

Based on the universal approximation property of neural networks and/or fuzzy logic systems [100], there exists a nonlinear function (a neural network or a fuzzy logic system) that can approximate any given real continuous function such as $\boldsymbol{V}_m\,[\dot{\boldsymbol{q}}_d{+}\boldsymbol{\Lambda e}] \in \mathbb{R}^p$. Assume the nonlinear function approximator, $\boldsymbol{u}_e$, is used to approximate $\boldsymbol{V}_m\,[\dot{\boldsymbol{q}}_d{+}\boldsymbol{\Lambda e}]$. Define

$$\widehat{\boldsymbol{u}_e} \equiv \boldsymbol{V}_m\,[\dot{\boldsymbol{q}}_d{+}\boldsymbol{\Lambda e}] - \boldsymbol{u}_e \tag{4.34}$$

with $\|\widehat{\boldsymbol{u}_{\boldsymbol{e}}}\| \leq \epsilon_N$ (ϵ_N is a constant), and $\widehat{\boldsymbol{u}_e}$ is the difference between the function and its estimated value.

Assumption on the boundedness of the desired trajectory

The desired trajectory and its first and second derivatives are assumed to be bounded, i.e., $\|\boldsymbol{q}_d\| \leq q_d$, $\|\dot{\boldsymbol{q}}_d\| \leq \dot{q}_d$, and $\|\ddot{\boldsymbol{q}}_d\| \leq \ddot{q}_d$. These bounds are used for the control design.

Now consider the following Laypunov function:

$$V = \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}\boldsymbol{r} + \int_0^t \boldsymbol{\Theta}^T\boldsymbol{F}\boldsymbol{\Theta}dt \tag{4.35}$$

Since $\boldsymbol{M}$ and $\boldsymbol{F}$ are positive-definite, then $V > 0$. Derivative of V is given by

$$\begin{aligned}\dot{V} &= \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{r}^T\boldsymbol{M}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{r}^T\dot{\boldsymbol{M}}\boldsymbol{r} + \boldsymbol{\Theta}^T\boldsymbol{F}\boldsymbol{\Theta}\\ &= \boldsymbol{r}^T\boldsymbol{M}\dot{\boldsymbol{r}} + \frac{1}{2}\boldsymbol{r}^T\dot{\boldsymbol{M}}\boldsymbol{r} + \boldsymbol{\Theta}^T\boldsymbol{F}\boldsymbol{\Theta}\end{aligned} \tag{4.36}$$

Substituting (4.30) into (4.36) yields

$$\begin{aligned}\dot{V} &= \boldsymbol{r}^T\big[-\boldsymbol{V}_m\boldsymbol{r} - \boldsymbol{K}_{PD}\boldsymbol{r} - \boldsymbol{X\Theta} - \boldsymbol{u}_e + \boldsymbol{f} + \boldsymbol{\tau}_d\big]\\ &\quad+\frac{1}{2}\boldsymbol{r}^T\dot{\boldsymbol{M}}\boldsymbol{r} + \boldsymbol{\Theta}^T\boldsymbol{F}\boldsymbol{\Theta}\end{aligned} \tag{4.37}$$

Substituting $\boldsymbol{f}$ in (4.19) into (4.37), $\dot{V}$ can be expressed as

$$\begin{aligned}\dot{V} &= \boldsymbol{r}^T\Big\{\boldsymbol{M}\,[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\,\dot{\boldsymbol{e}}] + \boldsymbol{V}_m\,[\dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda e} - \boldsymbol{r}]\\ &\quad+\boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_d - \boldsymbol{K}_{PD}\boldsymbol{r} - \boldsymbol{X\Theta} - \boldsymbol{u}_e\Big\}\\ &\quad+\frac{1}{2}\boldsymbol{r}^T\dot{\boldsymbol{M}}\boldsymbol{r} + \boldsymbol{\Theta}^T\boldsymbol{F}\boldsymbol{\Theta}\\ &= -\boldsymbol{r}^T\boldsymbol{K}_{PD}\boldsymbol{r} + \frac{1}{2}\boldsymbol{r}^T(\dot{\boldsymbol{M}} - 2\boldsymbol{V}_m)\boldsymbol{r} - \boldsymbol{r}^T\boldsymbol{X\Theta}\\ &\quad+\boldsymbol{\Theta}^T\boldsymbol{F}\boldsymbol{\Theta} + \boldsymbol{r}^T\Big\{\boldsymbol{M}\,[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\,\dot{\boldsymbol{e}}] + \boldsymbol{V}_m\,[\dot{\boldsymbol{q}}_d{+}\boldsymbol{\Lambda e}]\\ &\quad-\boldsymbol{u}_e + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_d\Big\}\end{aligned} \tag{4.38}$$

(4.38) be can expressed as

$$\begin{aligned}\dot{V} &= \frac{1}{2}\boldsymbol{r}^T(\dot{\boldsymbol{M}} - 2\boldsymbol{V}_m)\boldsymbol{r} - \boldsymbol{\Theta}^T(\boldsymbol{X}^T\boldsymbol{r} - \boldsymbol{F\Theta}) - \boldsymbol{r}^T\boldsymbol{K}_{PD}\boldsymbol{r} \\ &\quad +\boldsymbol{r}^T\boldsymbol{\phi} \end{aligned} \tag{4.39}$$

where $\boldsymbol{\phi}$ is given by

$$\boldsymbol{\phi} \equiv \boldsymbol{M}\,[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\,\dot{\boldsymbol{e}}] + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\epsilon} + \boldsymbol{\tau}_d \tag{4.40}$$

Since $\boldsymbol{r}^T\boldsymbol{\phi} \leq \|\phi\|.\|\boldsymbol{r}\|$ (using **P1** in Chapter 2), it follows that

$$\boldsymbol{r}^T\boldsymbol{\phi} \leq \left\|\boldsymbol{M}\,[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\,\dot{\boldsymbol{e}}] + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\epsilon} + \boldsymbol{\tau}_d\right\|.\|\boldsymbol{r}\| \tag{4.41}$$

Next, the upper bounds of $\|\boldsymbol{\phi}\|$: is derived

$$\begin{aligned}\|\boldsymbol{\phi}\| &\leq \|\boldsymbol{M}\,[\ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\,\dot{\boldsymbol{e}}]\| + \|\boldsymbol{F}(\dot{\boldsymbol{q}})\| + \|\boldsymbol{G}(\boldsymbol{q})\| + \|\boldsymbol{\tau}_d\| \\ &\leq \|\boldsymbol{M}\ddot{\boldsymbol{q}}_d\| + \|\boldsymbol{M\Lambda}\dot{\boldsymbol{e}}\| + \|\boldsymbol{F}(\dot{\boldsymbol{q}})\| + \|\boldsymbol{G}(\boldsymbol{q})\| + \|\boldsymbol{\epsilon}\| \\ &\quad + \|\boldsymbol{\tau}_d\| \end{aligned} \tag{4.42}$$

From the properties of RMs, the boundedness of the desired trajectories and their derivatives, and using (4.11), each term of (4.42) is bounded from above as follows:

$$\begin{aligned}\|\boldsymbol{M}\ddot{\boldsymbol{q}}_d\| &\leq \|\boldsymbol{M}\|\,\|\ddot{\boldsymbol{q}}_d\| \leq M_B\ddot{q}_d \qquad (4.43)\\ \|\boldsymbol{M\Lambda}\dot{\boldsymbol{e}}\| &\leq \|\boldsymbol{M}\|\,\|\boldsymbol{\Lambda}\|\,\|\dot{\boldsymbol{e}}\| \leq M_B\max(eig(\boldsymbol{\Lambda}))\,\|\boldsymbol{r}\| \\ \|\boldsymbol{F}(\dot{\boldsymbol{q}})\| &\leq F_B\,\|\dot{\boldsymbol{q}}\| + K_B \leq F_B\,\|\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{e}}\| + K_B \\ &\leq F_B\,\|\dot{\boldsymbol{q}}_d\| + F_B\,\|\dot{\boldsymbol{e}}\| + K_B \leq F_B(\dot{q}_d + \|\boldsymbol{r}\|) + K_B \\ \|\boldsymbol{G}(\boldsymbol{q})\| &\leq G_B \\ \|\boldsymbol{\epsilon}\| &\leq \epsilon_N \\ \|\boldsymbol{\tau}_d\| &\leq D_B \end{aligned}$$

Using the bounds in (4.43), (4.42) is expressed as

$$\begin{aligned}\|\boldsymbol{\phi}\| &\leq M_B\ddot{q}_d + M_B\max(eig(\boldsymbol{\Lambda}))\,\|\boldsymbol{r}\| + F_B(\dot{q}_d + \|\boldsymbol{r}\|) \\ &\quad +K_B + G_B + D_B + \epsilon_N \end{aligned} \tag{4.44}$$

Using (4.44), it is easy to show that the summation of the last two terms in (4.39), $-\boldsymbol{r}^T\boldsymbol{K}_{PD}\boldsymbol{r} + \boldsymbol{r}^T\boldsymbol{\phi}$, is bounded, i.e.,

$$\begin{aligned} -\boldsymbol{r}^T\boldsymbol{K}_{PD}\boldsymbol{r} + \boldsymbol{r}^T\boldsymbol{\phi} &\leq -K_{PD\min}\,\|\boldsymbol{r}\|^2 + \|\boldsymbol{\phi}\|\,\|\boldsymbol{r}\| \\ \leq -K_{PD\min}\,\|\boldsymbol{r}\|^2 &+ M_B\ddot{q}_d\,\|\boldsymbol{r}\| + M_B\max(eig(\boldsymbol{\Lambda}))\,\|\boldsymbol{r}\|^2 \\ +F_B(\dot{q}_d + \|\boldsymbol{r}\|)\,\|\boldsymbol{r}\| &+ (K_B + G_B + D_B + \epsilon_N)\,\|\boldsymbol{r}\| \end{aligned} \tag{4.45}$$

where $K_{PD\,\min}$ is the minimum eigenvalue of the matrix $\boldsymbol{K}_{PD}$. Note that **P2** in Chapter 2 have been used to derive (4.45). One can rewrite (4.45) as

$$-\boldsymbol{r}^T\boldsymbol{K}_{PD}\boldsymbol{r}+\boldsymbol{r}^T\boldsymbol{\phi}\leq(A\left\|\boldsymbol{r}\right\|+B)\left\|\boldsymbol{r}\right\| \tag{4.46}$$

where A and B are given by

$$\begin{aligned} A &= -K_{PD\,\min}+M_B\max(eig(\boldsymbol{\Lambda}))+F_B & (4.47)\\ B &= M_B\ddot{q}_d+F_B\dot{q}_d+(K_B+G_B+D_B+\epsilon_N) & (4.48)\end{aligned}$$

To ensure $\dot{V}$ in (4.39) is negative, it is enforced that $-\boldsymbol{r}^T\boldsymbol{K}_{PD}\boldsymbol{r}+\boldsymbol{r}^T\boldsymbol{\phi}<0$ and $\boldsymbol{X}^T\boldsymbol{r}-\boldsymbol{F\Theta}=\boldsymbol{0}$ [note that $\dot{\boldsymbol{M}}-2\boldsymbol{V}_m$, using the third property of RMs, is a skew-symmetric matrix and hence $\boldsymbol{r}^T(\dot{\boldsymbol{M}}-2\boldsymbol{V}_m)\boldsymbol{r}$ vanishes[2]]. First, to make $-\boldsymbol{r}^T\boldsymbol{K}_{PD}\boldsymbol{r}+\boldsymbol{r}^T\boldsymbol{\phi}<0$, one needs to ensure $(A\left\|\boldsymbol{r}\right\|+B)\left\|\boldsymbol{r}\right\|<0$. Thus

$$A\left\|\boldsymbol{r}\right\|+B<0 \tag{4.49}$$

Let $R\equiv-\frac{B}{A}$. Then, by making $A<0$ and knowing that B is a positive constant, for $\forall\left\|\boldsymbol{r}\right\|>R$, the inequality (4.49) holds and hence $-\boldsymbol{r}^T\boldsymbol{K}_{PD}\boldsymbol{r}+\boldsymbol{r}^T\boldsymbol{\phi}<0$.

The condition, $\boldsymbol{X}^T\boldsymbol{r}-\boldsymbol{F\Theta}=\boldsymbol{0}$, which is needed to ensure $\dot{V}<0$, will result in the derivation of a law dictating how the TSK parameters must change, referred to as the adjustment (adaptive) law. To achieve the adjustment law which is simply given by $\boldsymbol{X}^T\boldsymbol{r}-\boldsymbol{F\Theta}=\boldsymbol{0}$, it is required that

$$\boldsymbol{\Theta}=\boldsymbol{F}^{-1}\boldsymbol{X}^T\boldsymbol{r} \tag{4.50}$$

where $\boldsymbol{F}$ is the positive-definite design matrix, $\boldsymbol{X}$ includes the TSK as well as the controller parameters, and $\boldsymbol{r}$ is the filtered tracking error. To satisfy the adjustment law, (4.50), the controller parameters, m and n, are initially selected and kept fixed (they do not change in the adjustment process); hence, the TSK consequent parameters are adjusted according to (4.50) for each joint of the MRR to ensure $\dot{V}<0$. As a design guideline, m and n, are recommended to be chosen small (start with low gains).

In an independent joint control scheme, if the sufficient conditions are satisfied to ensure $\dot{V}<0$, i.e., (4.49) and (4.50), then the system is UUB stable and hence the error, $\boldsymbol{e}$, is bounded which means the MRR trajectories, i.e., $\boldsymbol{q}$ will be bounded (it was assumed $\boldsymbol{q}_d$ is bounded [see Subsection 4.3.3]).

4.4 Experiments

In this section, first the experimental setup is described. Next, the results of the experiments are presented.

[2] See Chapter 2, **P6**.

4.4.1 Experimental setup

Control implementations are performed on an MRR with two degrees-of-freedom, and there is a payload of $M = 6.80kg$ at the end of the second arm. Each joint uses a harmonic drive transmission mechanism. In this chapter, the controller is implemented and its performance for two different MRR configurations is evaluated.

The MRR is controlled by a MSK2812 DSP-based micro-controller via a controller area network (CAN) communication bus. The DSP is connected to a computer through a RS-232 to implement a developed control algorithm written in C++ language. The control command as well as the feedback signals are sent to and transmitted back from each joint via the CAN bus.

The MSK2812 DSP is operated at 150MHz and has powerful calculation capabilities, but the data transmission rate of the CAN bus is 1 Mbit/s. As a result, the control frequency is limited by the CAN bus data transmission rate; hence, only a low control frequency is possible for implementing the controller algorithm. Figure 4.3 shows the system hardware used in the experiments.

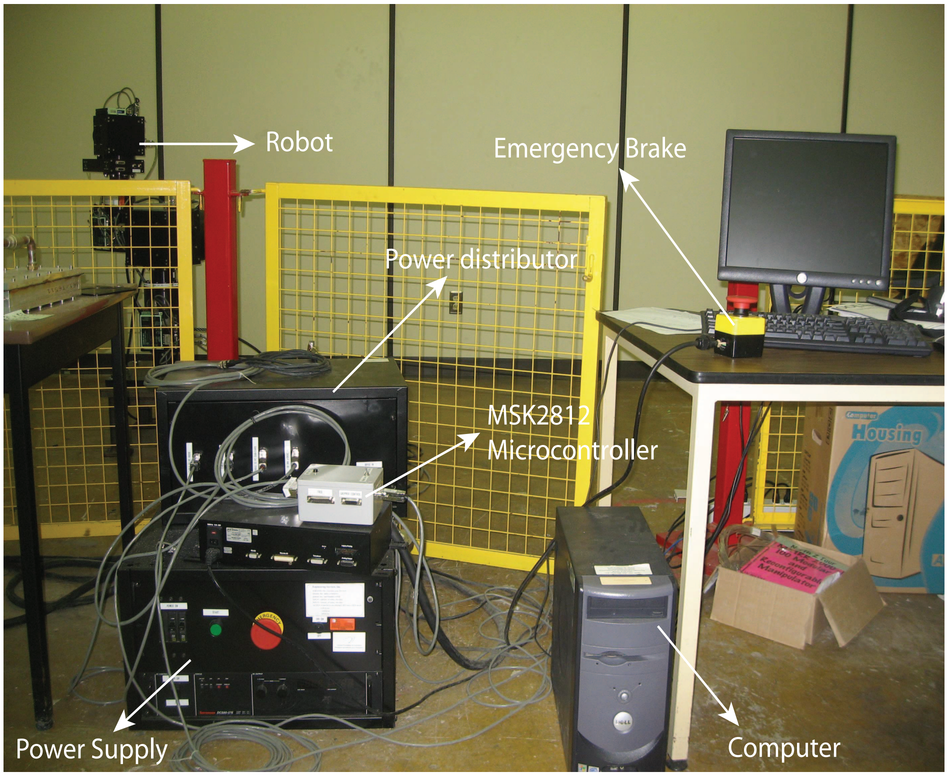


Figure 4.3: System hardware.

In the experiments, five controller structures are tested and evaluated, namely, PD, T1 TSK FLC with fixed-parameters, T1 TSK FLC with varying parameters, IT2 TSK FLC with fixed-parameters, and IT2 TSK FLC with varying parameters and compare their performances. The fixed-parameters controller uses the TSK parameters chosen from Table 4.2 and are kept unchanged. In this scheme, only the controller parameters m and n need to be tuned. For the varying-parameters controller, the same m and n (properly tuned in the design of the fixed-parameter controller) are used, and the adjustment or adaptive law adjusts the TSK parameters when the dynamic parameters of the MRR are unknown due to reconfigurability. An Radial Basis Function (RBF) neural network [150] is used to approximate $\boldsymbol{u}_e$. However, for the MRR configurations in the experiments and by using sinusoids (see Subsection 4.4.2 for the details of the reference trajectory), the magnitude of $\boldsymbol{u}_e$ is small compared to $\boldsymbol{u}_{PD}$ and $\boldsymbol{u}_{Fuzzy}$. Therefore, this term is omitted in the experiments to reduce the online computational complexity.

The linear controller parameters are $Kp = 0.25, Kd = 0.65$. The tuning parameters of the IT2 TSK FLC are $m = 0.001, n = 0.0012$. To make a fair comparison between T1 and IT2 FLCs, the total control efforts of these controllers should be comparable. Therefore, the T1 TSK consequent parameters are multiplied by $k = m + n$ to get the best possible performance of the T1 FLC. This will ensure the control effort of both T2 and the IT2 controllers are comparable ; otherwise, the T1 will have larger control output which results in an undesirable tracking performance and hence an unfair judgment. For the ease of real-time implementation, the following design matrices are chosen to be diagonal:

$$\boldsymbol{F} \in \mathbb{R}^{18\times 18}$$

with $\boldsymbol{F}_{ii} = 10\ (i = 1, \cdots, 18)$, and $\boldsymbol{\Lambda}_{IT2} = diag[0.001, 0.0019]$, and $\boldsymbol{\Lambda}_{T1} = diag[0.0005, 0.0004]$.

4.4.2 Results

A sinusoidal trajectory is applied to each joint with amplitude and frequency of $\frac{\pi}{2}$ and $0.05Hz$, respectively, i.e.,

$$q_d = Asin(2\pi f) \tag{4.51}$$

where $A = 90°$ and $f = 0.05Hz$. To compare the performance of the controllers, mean squared error (MSE) and percentage improvement (PIm) are used, which are defined as follows:

$$MSE = \frac{\sum_{i=1}^{n} e_i^2}{n} \tag{4.52}$$

where n is the number of sampled points in each experiment. And,

$$PIm = \frac{MSE_{controller} - MSE_{PD}}{MSE_{PD}} \times 100 \tag{4.53}$$

where MSE_{PD} and $MSE_{controller}$ are the *MSE*s of the PD controller and a specific controller (either fixed-parameters or varying-parameters), respectively.

Configuration 1

In this configuration, the axes of the first and second joints are parallel to each other, as shown in Figure 4.4.

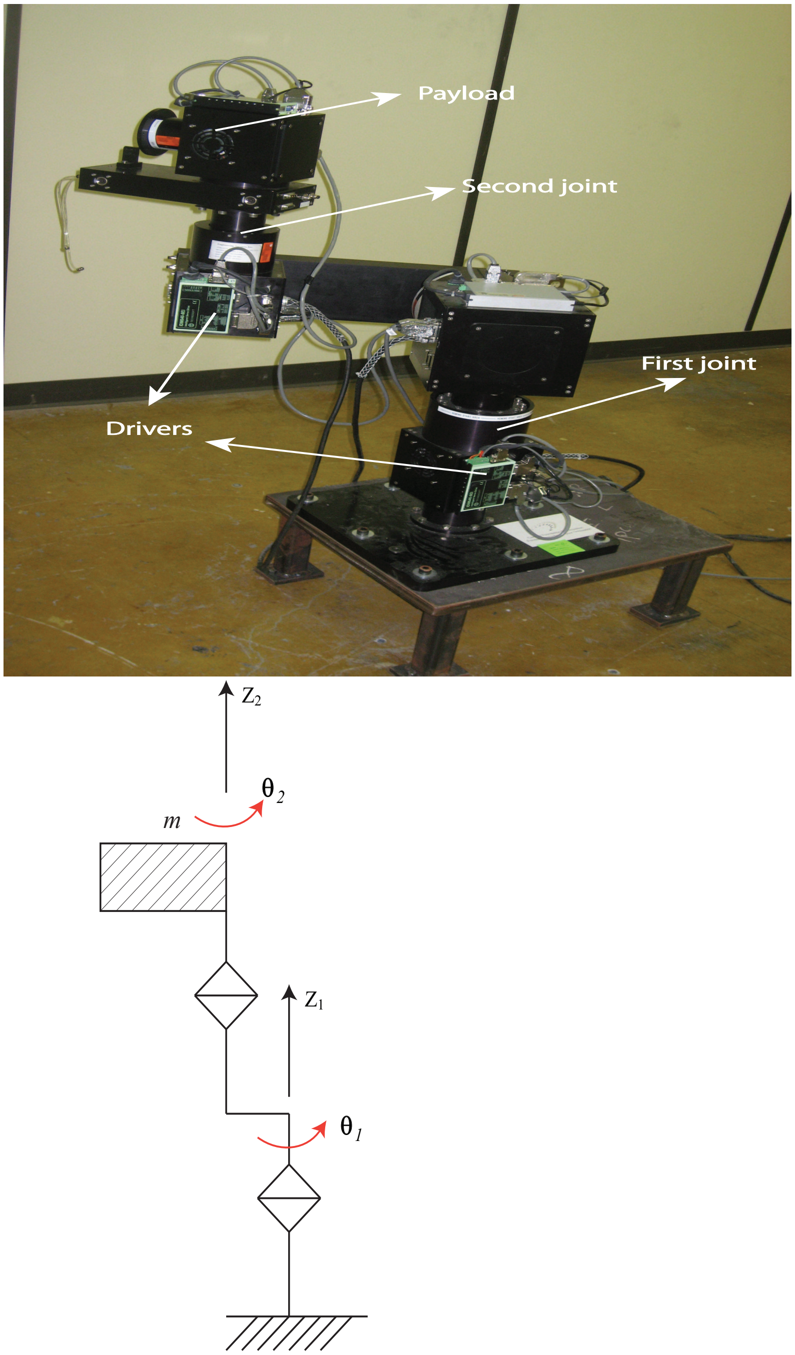


Figure 4.4: First configuration and its schematic.

Table 4.3 summarizes the results (averaged results) for the PD controller, T1 and IT2 controllers with fixed-parameters, and T1 and IT2 controllers with varying-parameters in terms of *MSE* and also *PI* (%).

Table 4.3: First configuration results.

	First joint		Second joint	
Controller	*MSE*	*PIm* (%)	*MSE*	*PIm* (%)
PD	1.2478	–	0.3323	–
T1 fixed-parameters	0.7682	38.44	0.2572	22.61
T1 varying-parameters	0.7294	41.55	0.2313	30.41
IT2 fixed-parameters	0.6098	51.13	0.1851	44.29
IT2 varying-parameters	0.5124	58.93	0.1724	48.12

From the tabulated results, it is concluded that the IT2 varying-parameters controller produces the best tracking output and outperforms the other controllers significantly, as shown in the third and fifth columns. The effectiveness of the IT2 varying-parameters controller is more apparent in the second joint where the dynamic coupling effect is more profound. Those results are consistent over several runs for the same controller and thus indicating the repeatability of the result; hence, it is deduced that the IT2 FLC can better handle the uncertain portion of the MRR dynamics compared to linear or fixed-parameters T1 or IT2 TSK FLCs.

Control efforts for different controllers are shown in Figures 4.5-4.8. Figures 4.5 and 4.7 compare the control efforts of the PD and T1 controllers, for the first and second joints, respectively, and Figures 4.6 and 4.8 show a comparison between the control outputs of the PD and IT2 controllers for the first and second joints, respectively. As can be seen, the control outputs of the fixed-parameters and varying-parameters controllers (all T1 and IT2) are comparable to the PD controller. Moreover, the control efforts of other controllers are comparable to each other. Hence, without increasing the control efforts, the developed fixed-parameters as well as varying-parameters controllers outperform the PD controller significantly. More specifically, the varying-parameters IT2 TSK FLC has the best performance without increasing the control output compared to others. Note that the effect of dynamic coupling is noticeable in the second joint as more control action is required to minimize the tracking error. This observation is consistent for all the controllers. The oscillation observed in the control effort of the controllers in the second joint is due to the relatively heavy payload at the end of the second arm.

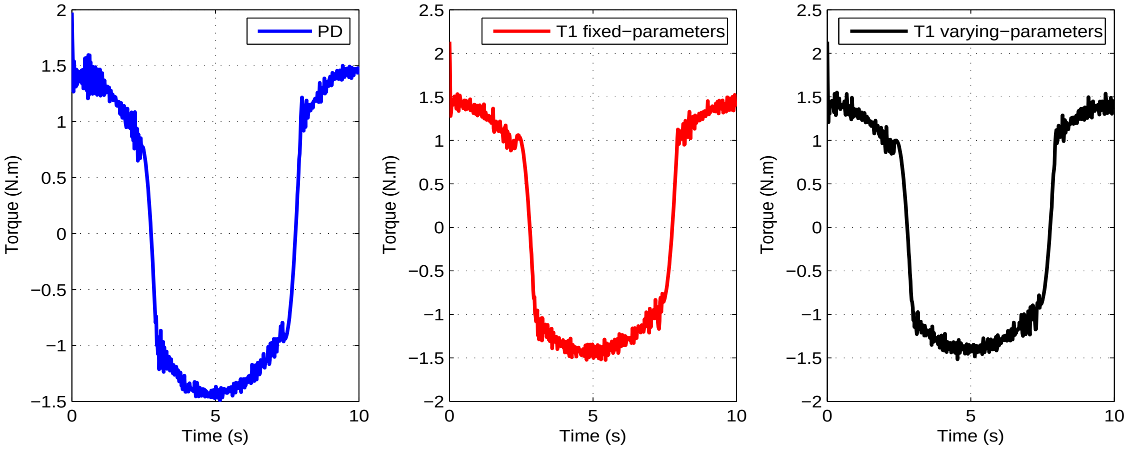


Figure 4.5: Control efforts of PD and T1 controllers for joint 1: first configuration.

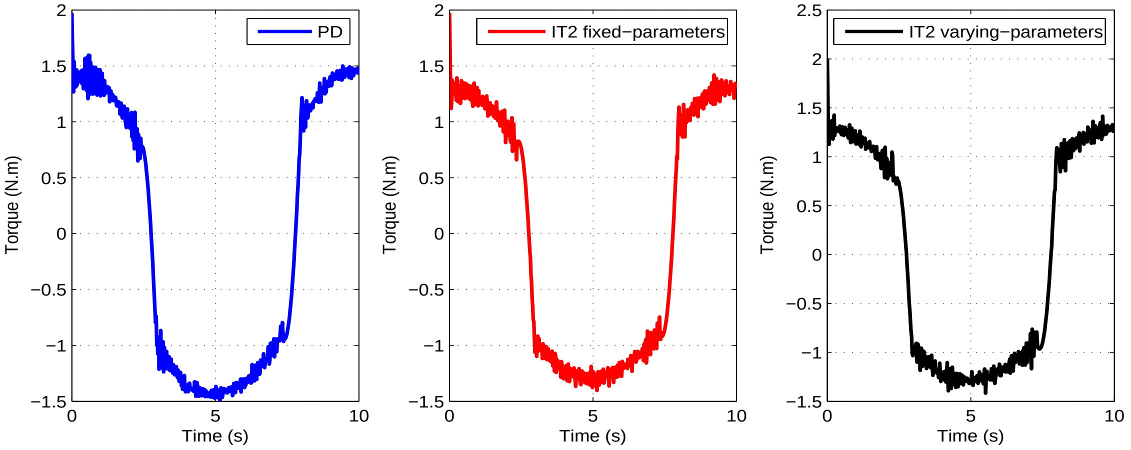


Figure 4.6: Control efforts of PD and IT2 controllers for joint 1: first configuration.

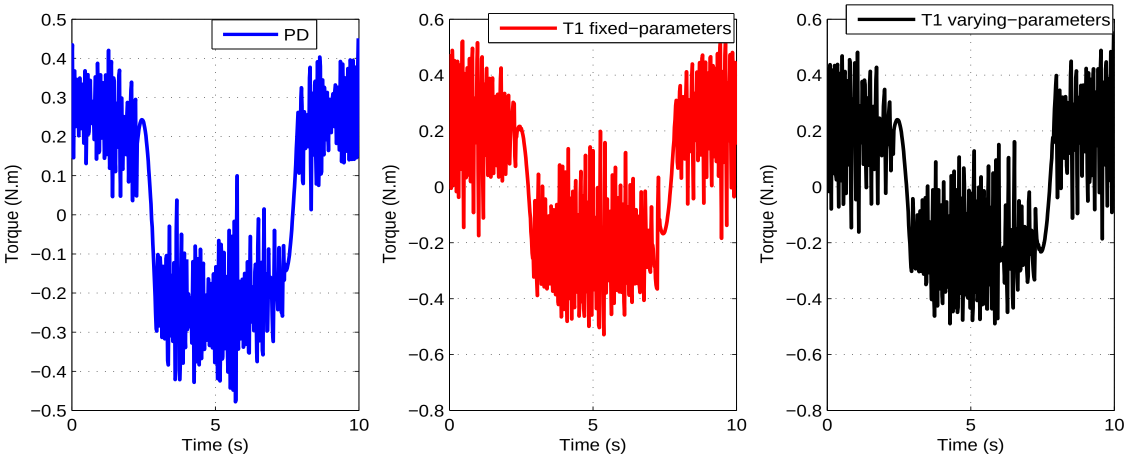


Figure 4.7: Control efforts of PD and T1 controllers for joint 2: first configuration.

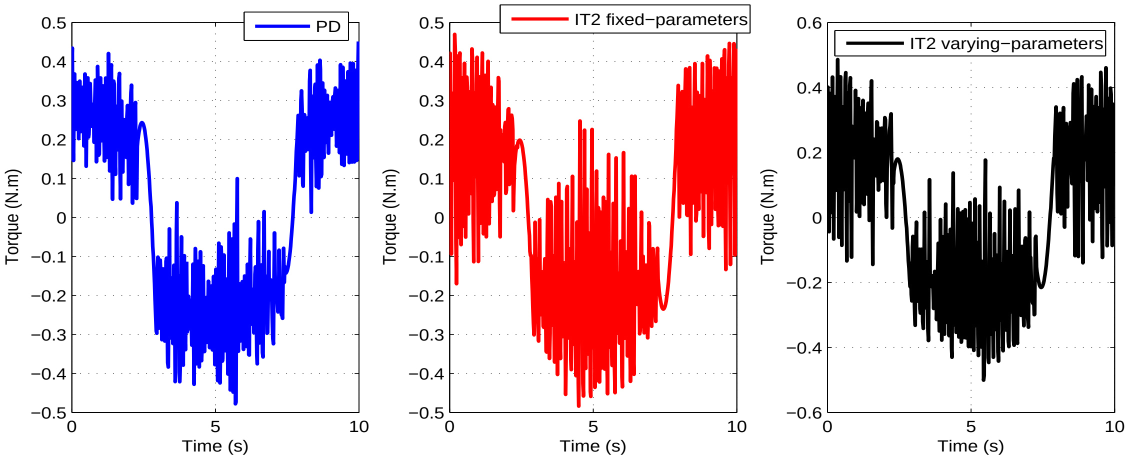


Figure 4.8: Control efforts of PD and IT2 controllers for joint 2: first configuration.

The plots that compare the position errors of different controllers testes are provided in Appendix A.

Configuration 2

In this configuration, the axes of the first and second joints are perpendicular to each other. Figure 4.9 shows the configuration under investigation. It should be noted that the control parameters are kept unchanged from configuration 1. When the robot is reconfigured, dynamic parameters of the MRR change and are still unknown; hence, the adjustment law will be invoked to handle the varying dynamics of the system.

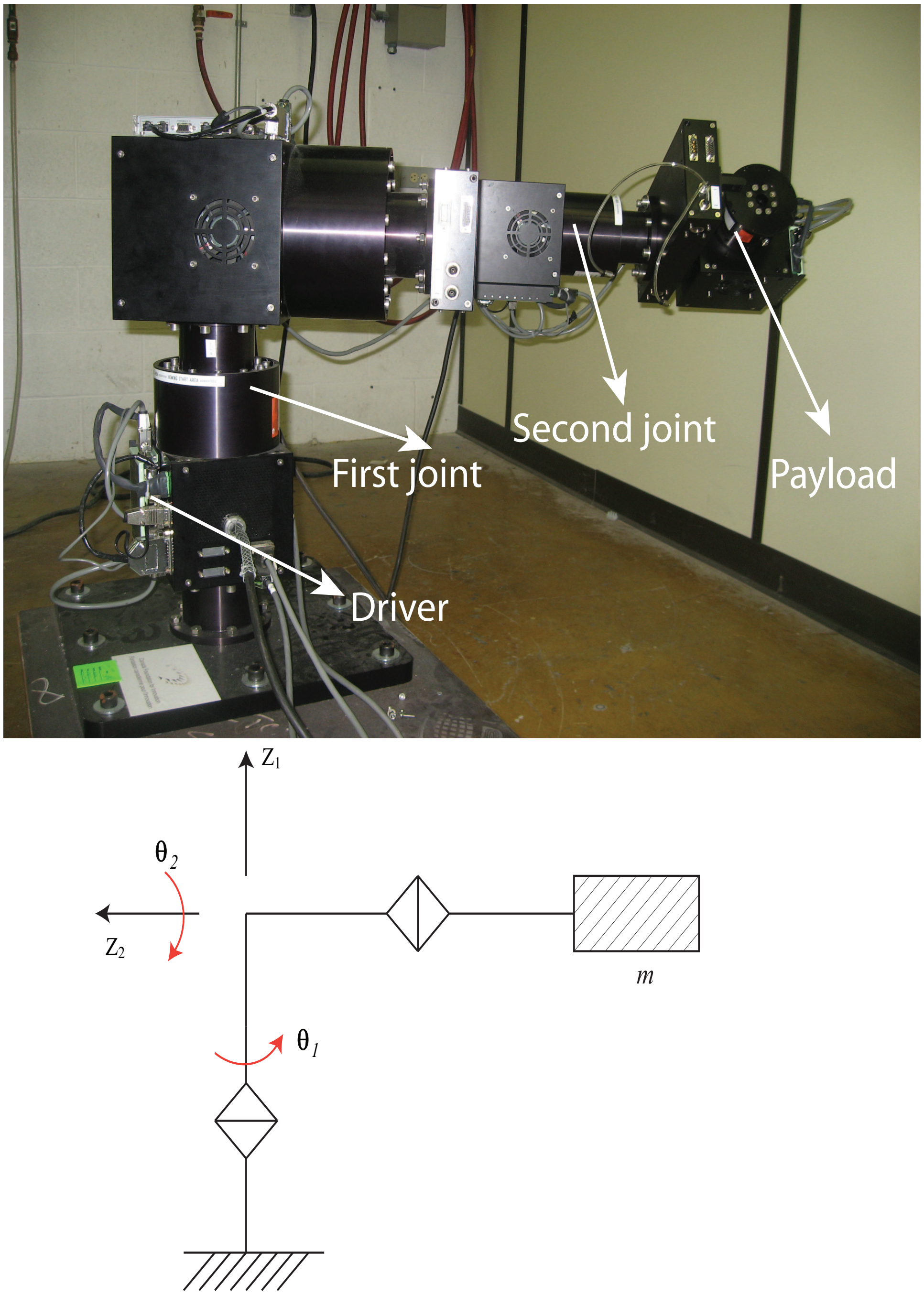


Figure 4.9: Second configuration and its schematic.

Similar to the first configuration, the results from Table 4.4 verify the superiority of the IT2 controller with varying-parameters compared to other controllers.

Figures 4.10-4.13 compare the control efforts of the different controllers. Similar to the first configuration, while the IT2 varying-parameters controller is producing enhanced results, its control output is very much comparable to the other controllers. This is seen as another major advantage of the proposed varying-parameters control strategy.

Table 4.4: Second configuration results.

	First joint		Second joint	
Controller	*MSE*	*PIm* (%)	*MSE*	*PIm* (%)
PD	1.1849	–	0.5709	–
T1 fixed-parameters	0.7391	37.62	0.5303	7.11
T1 varying-parameters	0.7344	38.02	0.5129	8.34
IT2 fixed-parameters	0.6302	46.82	0.5107	10.54
IT2 varying-parameters	0.5637	52.43	0.4181	26.77

Note also the noticeable tracking performance enhancement in the second joint which validates the ability of the varying-parameters controller to compensate for the dynamic coupling effect. Please see Appendix B for position error trajectories of each controller.

In the above experiments, adding an integral term to the PD controller, resulting in a PID controller, will not increase the tracking performance, because the PID controller acts as a high gain controller at only low frequencies and not at the desired frequency. Experiments with different K_I gains were performed and results were verified/observed.

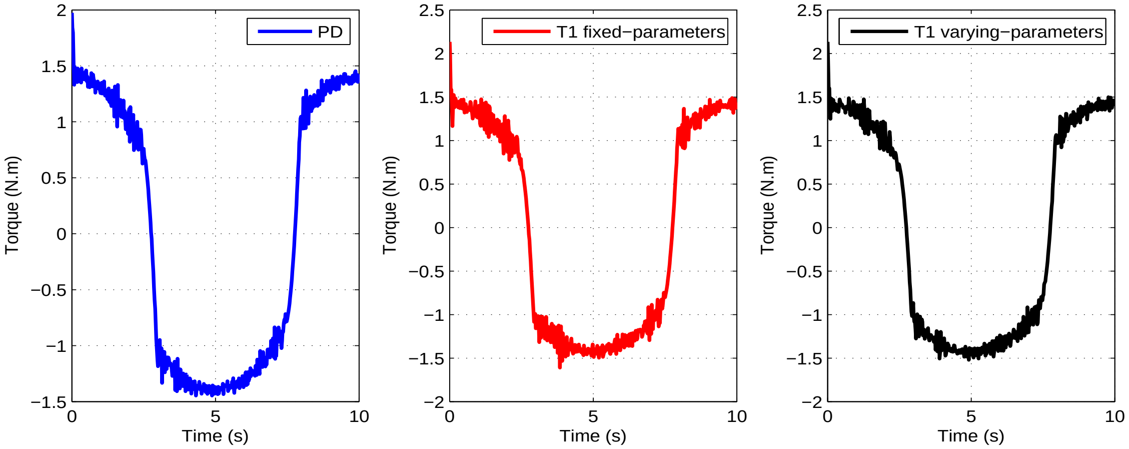


Figure 4.10: Control efforts of PD and T1 controllers for joint 1: second configuration.

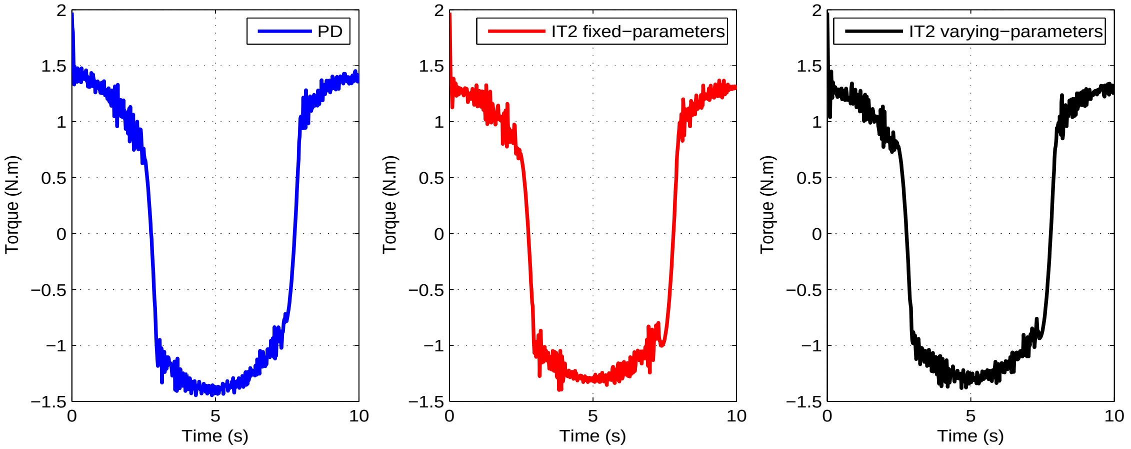


Figure 4.11: Control efforts of PD and IT2 controllers for joint 1: second configuration.

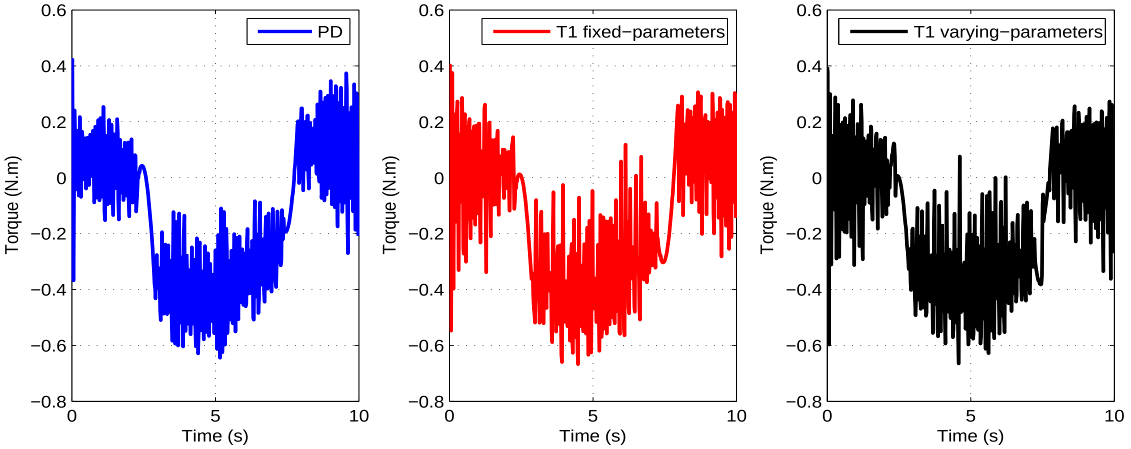


Figure 4.12: Control efforts of PD and T1 controllers for joint 2: second configuration.

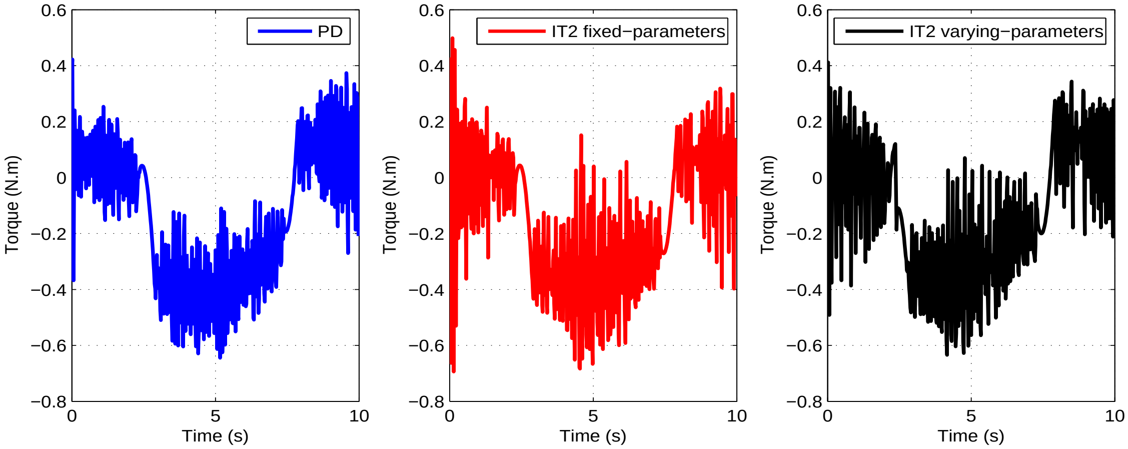


Figure 4.13: Control efforts of PD and IT2 controllers for joint 2: second configuration.

4.5 Conclusion

Experimental results verified the enhanced performance of the varying-parameters IT2 TSK FLC for both configurations. Moreover, the advantage of the proposed controller methodology is that a minimum effort is required for the designer to tune the controller parameters, m and n. It is sufficient for these parameters to be initially chosen small (small gains). During the adjustment process described by (4.50), the TSK parameters are adjusted automatically to ensure enhanced tracking and UUB stability. Furthermore, during experiments it was observed that the performance of both T1 and IT2 TSK FLCs are not sensitive to the change of initial TSK consequent parameters.

This chapter presented the design of a novel adaptive IT2 TSK FLC for tracking purposes of MRRs. It was shown that the proposed controller can effectively handle uncertainties as well as the dynamic coupling between the joints of two serial MRR configurations while outperforming a well-known linear controller.

Chapter 5

On the Robustness of T1 and IT2 Fuzzy Logic Systems in Modeling and Identification

5.1 Introduction

Modeling is the main step in the identification of physical systems such as robotics, communications, medical systems, to name a few. More specifically, identification is vital in engineering systems estimation and control, both of which have found a wide range of applications in aerospace, automotive, micro-robotics, weather forecasting, etc.

In identification problems, usually some data (referred to as sampling points and often obtained by dedicated experiments) are used to capture the entire dynamics (or portions) of plants. To accurately determine the dynamics of the system considered, it is crucial to capture the behavior of systems through I/O mappings. However, hardware/software limitations, unavoidable round off, and truncation of a system's errors will ultimately result in deviation of the output from the optimum point for which the system was originally designed: the desired output. The output of a system can also change because of uncertainties, perhaps substantial ones, in sensory data. Moreover, in control, considerable deviations in the output change the closed-loop performance that, in severe cases, might even lead to an unstable system.

When a system is subjected to small deviations around the sampling points (operating points), it is essential to find the maximum tolerance of the system with respect to those perturbations, referred to herein as the system's robustness. Thus, in the context of modeling, robustness is a metric for measuring the impact of input deviations on the desired output.

Based on the literature review presented in Chapter 1, research on the robustness of T1 FLSs is mostly limited to fuzzy operators and, hence more in-depth analyses into robust system design is required. In this chapter, the robustness of T1 and IT2 FLSs is investigated. The robustness problems of T1 and IT2 TSK FLSs are mathematically formulated and a procedure is proposed for the design of robust FLSs. The contributions of this chapter are a) a systematic robustness analysis of FLSs in the presence of parameter perturbations, and, b) an efficient procedure for the design of robust FLSs that will enable engineers and researches to design more robust FLSs for modeling and control applications. The focus of this work is on IT2 FLSs, but since T1s are a special class of IT2s, the proposed methodology in this thesis is general and can be used for both T1 and IT2 FLSs. The organization of this chapter is as follows: Section 5.2 presents a definition and mathematical derivation of FLSs' robustness. Section 5.3 provides the upper bounds of the maximum output deviation. Section 5.4 presents a procedure for designing robust FLSs. Section 5.5 presents numerical examples. Finally, Section 5.6 provides the chapter conclusions.

5.1.1 Problem Statement

Given an FLS, either T1 or IT2, it is of great interest to find the maximum input deviations so that the output deviation of the FLS, ΔY, does not exceed a threshold determined by a designer.

The purpose of this chapter is to present a systematic methodology to investigate the robustness of FLSs (both T1 and IT2). The robustness analysis will exploit a general structure of IT2 TSK FLSs, and since T1 TSK FLSs are a special case of IT2 TSK FLSs, the presented results can be readily used for T1s as well.

In Chapter 3 a new inference engine for IT2 TSK FLSs was introduced which its output is given by

$$Y_{\text{NEW}}(\boldsymbol{x}^*) = m\frac{\sum_{i=1}^{M} \underline{f}^i\,(\boldsymbol{x}^*)\,y_i(\boldsymbol{x}^*)}{\sum_{i=1}^{M} \underline{f}^i\,(\boldsymbol{x}^*)} + n\frac{\sum_{i=1}^{M} \overline{f}^i\,(\boldsymbol{x}^*)\,y_i(\boldsymbol{x}^*)}{\sum_{i=1}^{M} \overline{f}^i\,(\boldsymbol{x}^*)} \tag{5.1}$$

where y_i is the output of each rule, and $\underline{f}^i\,(\boldsymbol{x}^*)$ and $\overline{f}^i\,(\boldsymbol{x}^*)$ are given by (2.18) and (2.19), respectively (if $M = 1$, then $m + n = 1$). In this chapter, $Y_{\text{NEW}}(\boldsymbol{x}^*)$ is adopted as the inference engine simply because it will make it possible to analytically investigate the robustness of IT2 TSK FLSs.

5.2 Robustness of FLSs

In this section, the robustness of FLSs is formulated and the governing equations are derived. The final expressions derived herein are used for the analysis and design of robust FLSs in the subsequent sections.

To begin, define

$$\underline{w}_i\left(\boldsymbol{x}^*\right) \equiv \frac{\underline{f}_i\left(\boldsymbol{x}^*\right)}{\sum_{i=1}^{M} \underline{f}_i\left(\boldsymbol{x}^*\right)} \tag{5.2}$$

$$\overline{w}_i\left(\boldsymbol{x}^*\right) \equiv \frac{\overline{f}_i\left(\boldsymbol{x}^*\right)}{\sum_{i=1}^{M} \overline{f}_i\left(\boldsymbol{x}^*\right)} \tag{5.3}$$

Thus, (5.1) can be expressed as

$$Y(\boldsymbol{x}^*) = m\sum_{i=1}^{M} \underline{w}^i\left(\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^*) + n\sum_{i=1}^{M} \overline{w}^i\left(\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^*) \tag{5.4}$$

Suppose $\boldsymbol{x}^*$ is deviated by a small $\Delta\boldsymbol{x}^*$. The new input to the FLS would be $\boldsymbol{x}^* + \Delta\boldsymbol{x}^*$. The deviated output, ΔY, is given by

$$\begin{aligned} \Delta Y &= Y(\boldsymbol{x}^* + \Delta\boldsymbol{x}^*) - Y(\boldsymbol{x}^*) \\ &= m\sum_{i=1}^{M} \underline{w}^i\left(\boldsymbol{x}^* + \Delta\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^* + \Delta\boldsymbol{x}^*) + n\sum_{i=1}^{M} \overline{w}^i\left(\boldsymbol{x}^* + \Delta\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^* + \Delta\boldsymbol{x}^*) \\ &- m\sum_{i=1}^{M} \underline{w}^i\left(\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^*) - n\sum_{i=1}^{M} \overline{w}^i\left(\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^*) \end{aligned} \tag{5.5}$$

To further reduce the notations, define

$$\Delta A \equiv m\sum_{i=1}^{M} \underline{w}^i\left(\boldsymbol{x}^* + \Delta\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^* + \Delta\boldsymbol{x}^*) - m\sum_{i=1}^{M} \underline{w}^i\left(\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^*) \tag{5.6}$$

$$\Delta B \equiv n\sum_{i=1}^{M} \overline{w}^i\left(\boldsymbol{x}^* + \Delta\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^* + \Delta\boldsymbol{x}^*) - n\sum_{i=1}^{M} \overline{w}^i\left(\boldsymbol{x}^*\right) y_i(\boldsymbol{x}^*) \tag{5.7}$$

Then, (5.5) can be written as

$$\Delta Y = \Delta A + \Delta B \tag{5.8}$$

In what follows, ΔA and ΔB are expanded to help formulate ΔY. The expression for only ΔA is derived (similar analysis can be done for ΔB). Since it is easier to work with $\frac{\Delta A}{m}$, divide ΔA in (5.6) by m to get

$$\frac{\Delta A}{m} = \sum_{i=1}^{M} \underline{w}^{i}\left(\boldsymbol{x}^{*}+\Delta \boldsymbol{x}^{*}\right) y_{i}(\boldsymbol{x}^{*}+\Delta \boldsymbol{x}^{*}) - \sum_{i=1}^{M} \underline{w}^{i}\left(\boldsymbol{x}^{*}\right) y_{i}(\boldsymbol{x}^{*}) \tag{5.9}$$

Assuming $\Delta \boldsymbol{x}^{*}$ is small and using a Taylor-series expansion, $y^{i}\left(\boldsymbol{x}^{*}+\Delta \boldsymbol{x}^{*}\right)$ and $\underline{w}^{i}\left(\boldsymbol{x}^{*}+\Delta \boldsymbol{x}^{*}\right)$ can be expressed around $\boldsymbol{x}^{*}$ as

$$\begin{aligned} \underline{w}^{i}\left(\boldsymbol{x}^{*}+\Delta \boldsymbol{x}^{*}\right) &\simeq \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)+\sum_{j=1}^{p} \frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}+h.o.t & (5.10) \\ y^{i}\left(\boldsymbol{x}^{*}+\Delta \boldsymbol{x}^{*}\right) &\simeq y^{i}\left(\boldsymbol{x}^{*}\right)+\sum_{j=1}^{p} \frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}+h.o.t & (5.11) \end{aligned}$$

where $h.o.t$ is short for higher order terms. For the derivation of $\frac{\partial \underline{w}^{i}(\boldsymbol{x}^{*})}{\partial x_{j}}$ see Appendix C.1.

Using (5.10) and (5.11) and neglecting the $h.o.t$, (5.9) can be written as

$$\begin{aligned} \frac{\Delta A}{m} &= \sum_{i=1}^{M}\left[\left(\underline{w}^{i}\left(\boldsymbol{x}^{*}\right)+\sum_{j=1}^{p} \frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}\right)\left(y^{i}\left(\boldsymbol{x}^{*}\right)+\sum_{j=1}^{p} \frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}\right)\right]-\sum_{i=1}^{M} \underline{w}^{i}\left(\boldsymbol{x}^{*}\right) y_{i}(\boldsymbol{x}^{*}) \\ &= \sum_{i=1}^{M}\left[\underline{w}^{i}\left(\boldsymbol{x}^{*}\right) y^{i}\left(\boldsymbol{x}^{*}\right)+\underline{w}^{i}\left(\boldsymbol{x}^{*}\right) \sum_{j=1}^{p} \frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}+y^{i}\left(\boldsymbol{x}^{*}\right) \sum_{j=1}^{p} \frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}\right. \\ &+ \left.\left(\sum_{j=1}^{p} \frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}\right)\left(\sum_{j=1}^{p} \frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}\right)\right]-\sum_{i=1}^{M} \underline{w}^{i}\left(\boldsymbol{x}^{*}\right) y_{i}(\boldsymbol{x}^{*}) \end{aligned} \tag{5.12}$$

Assuming Δx_{j} is small, the fourth term in (5.12),$\left(\sum_{j=1}^{p} \frac{\partial \underline{w}^{i}(\boldsymbol{x}^{*})}{\partial x_{j}} \Delta x_{j}\right)\left(\sum_{j=1}^{p} \frac{\partial y^{i}(\boldsymbol{x}^{*})}{\partial x_{j}} \Delta x_{j}\right)$, is small and hence can be further neglected. Therefore, $\frac{\Delta A}{m}$ can be simply rewritten as

$$\begin{aligned} \frac{\Delta A}{m} &\simeq \sum_{i=1}^{M}\left[\underline{w}^{i}\left(\boldsymbol{x}^{*}\right) y^{i}\left(\boldsymbol{x}^{*}\right)+\underline{w}^{i}\left(\boldsymbol{x}^{*}\right) \sum_{j=1}^{p} \frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}+y^{i}\left(\boldsymbol{x}^{*}\right) \sum_{j=1}^{p} \frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}\right] \\ &- \sum_{i=1}^{M} \underline{w}^{i}\left(\boldsymbol{x}^{*}\right) y_{i}(\boldsymbol{x}^{*}) \\ &= \sum_{i=1}^{M}\left[\underline{w}^{i}\left(\boldsymbol{x}^{*}\right) \sum_{j=1}^{p} \frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}+y^{i}\left(\boldsymbol{x}^{*}\right) \sum_{j=1}^{p} \frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} \Delta x_{j}\right] \end{aligned} \tag{5.13}$$

Thus, ΔA is given by

$$\Delta A = m\sum_{i=1}^{M}\left[\underline{w}^{i}\left(\boldsymbol{x}^{*}\right)\sum_{j=1}^{p}\frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j} + y^{i}\left(\boldsymbol{x}^{*}\right)\sum_{j=1}^{p}\frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j}\right] \tag{5.14}$$

Paralleling the above analyses, a similar expression for ΔB is derived

$$\Delta B = n\sum_{i=1}^{M}\left[\overline{w}^{i}\left(\boldsymbol{x}^{*}\right)\sum_{j=1}^{p}\frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j} + y^{i}\left(\boldsymbol{x}^{*}\right)\sum_{j=1}^{p}\frac{\partial \overline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j}\right] \tag{5.15}$$

Therefore, ΔY can be expressed as

$$\begin{aligned}\Delta Y &= m\sum_{i=1}^{M}\left[\underline{w}^{i}\left(\boldsymbol{x}^{*}\right)\sum_{j=1}^{p}\frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j} + y^{i}\left(\boldsymbol{x}^{*}\right)\sum_{j=1}^{p}\frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j}\right]\\ &+ n\sum_{i=1}^{M}\left[\overline{w}^{i}\left(\boldsymbol{x}^{*}\right)\sum_{j=1}^{p}\frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j} + y^{i}\left(\boldsymbol{x}^{*}\right)\sum_{j=1}^{p}\frac{\partial \overline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j}\right]\end{aligned} \tag{5.16}$$

It is straightforward to show that (5.16) can be written as

$$\begin{aligned}\Delta Y &= \sum_{i=1}^{M}\sum_{j=1}^{p}\left[m\underline{w}^{i}\left(\boldsymbol{x}^{*}\right)\frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j} + my^{i}\left(\boldsymbol{x}^{*}\right)\frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j}\right]\\ &+ \sum_{i=1}^{M}\sum_{j=1}^{p}\left[n\overline{w}^{i}\left(\boldsymbol{x}^{*}\right)\frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j} + ny^{i}\left(\boldsymbol{x}^{*}\right)\frac{\partial \overline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\Delta x_{j}\right]\end{aligned} \tag{5.17}$$

Rearranging (5.17) gives

$$\begin{aligned}\Delta Y &= \sum_{j=1}^{p}\sum_{i=1}^{M}\left[m\underline{w}^{i}\left(\boldsymbol{x}^{*}\right)\frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} + my^{i}\left(\boldsymbol{x}^{*}\right)\frac{\partial \underline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\right].\Delta x_{j}\\ &+ \sum_{j=1}^{p}\sum_{i=1}^{M}\left[n\overline{w}^{i}\left(\boldsymbol{x}^{*}\right)\frac{\partial y^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}} + ny^{i}\left(\boldsymbol{x}^{*}\right)\frac{\partial \overline{w}^{i}\left(\boldsymbol{x}^{*}\right)}{\partial x_{j}}\right].\Delta x_{j}\end{aligned} \tag{5.18}$$

Define

$$X_j^l \equiv \sum_{i=1}^{M} \left[m\underline{w}^i\left(\boldsymbol{x}^*\right) \frac{\partial y^i\left(\boldsymbol{x}^*\right)}{\partial x_j} + my^i\left(\boldsymbol{x}^*\right) \frac{\partial \underline{w}^i\left(\boldsymbol{x}^*\right)}{\partial x_j} \right] \tag{5.19}$$

$$X_j^u \equiv \sum_{i=1}^{M} \left[n\overline{w}^i\left(\boldsymbol{x}^*\right) \frac{\partial y^i\left(\boldsymbol{x}^*\right)}{\partial x_j} + ny^i\left(\boldsymbol{x}^*\right) \frac{\partial \overline{w}^i\left(\boldsymbol{x}^*\right)}{\partial x_j} \right] \tag{5.20}$$

Then, (5.18) can be expressed as

$$\Delta Y \;=\; \sum_{j=1}^{p} X_j^l \Delta x_j + \sum_{j=1}^{p} X_j^u \Delta x_j = \sum_{j=1}^{p} \left(X_j^l + X_j^u\right) \Delta x_j \tag{5.21}$$

The final expression for ΔY, given by (5.21), is significant because it allows us to express the output deviation, ΔY, as a linear combinations of Δx_j's. Hence, connects the output deviation to the perturbations in the inputs.

When TSK rule structure is used in the design of a FLS, $\frac{\partial y^i(\boldsymbol{x}^*)}{\partial x_j} = a_j^i$. Therefore, the expressions in (5.19) and (5.20)[1] can be simplified as follows:

$$\begin{aligned} X_j^l \;=\; & \sum_{i=1}^{M} \left[m \frac{\underline{f}^i}{\sum_{l=1}^{M} \overline{f}^l} a_j^i + m \left(a_0 + \sum_{k=1}^{p} a_k^i x_k \right) \right. \\ & \left. \times \; \frac{\left(\prod_{\substack{k=1 \\ k \neq j}}^{p-1} \underline{\mu}_k^i(x_k) \right) \frac{\partial \underline{\mu}_j^i(x_j)}{\partial x_j} \left[\sum_{l=1}^{M} \underline{f}^l \right] - \underline{f}^i \sum_{l=1}^{M} \left[\left(\prod_{\substack{k=1 \\ k \neq j}}^{p-1} \underline{\mu}_k^l(x_k) \right) \frac{\partial \underline{\mu}_j^l(x_j)}{\partial x_j} \right]}{\left[\sum_{l=1}^{M} \underline{f}^l \right]^2} \right] \end{aligned} \tag{5.22}$$

$$\begin{aligned} X_j^u \;=\; & \sum_{i=1}^{M} \left[n \frac{\overline{f}^i}{\sum_{l=1}^{M} \overline{f}^l} a_j^i + n \left(a_0 + \sum_{k=1}^{p} a_k^i x_k \right) \right. \\ & \left. \times \; \frac{\left(\prod_{\substack{k=1 \\ k \neq j}}^{p-1} \overline{\mu}_k^i(x_k) \right) \frac{\partial \overline{\mu}_j^i(x_j)}{\partial x_j} \left[\sum_{l=1}^{M} \overline{f}^l \right] - \overline{f}^i \sum_{l=1}^{M} \left[\left(\prod_{\substack{k=1 \\ k \neq j}}^{p-1} \overline{\mu}_k^l(x_k) \right) \frac{\partial \overline{\mu}_j^l(x_j)}{\partial x_j} \right]}{\left[\sum_{l=1}^{M} \overline{f}^l \right]^2} \right] \end{aligned} \tag{5.23}$$

5.2.1 Robustness Definition

Suppose $\Delta Y_{desired} \geq 0$ is the desired (allowable) output deviation. Robustness is defined as follows:

[1] Note that $\underline{f}^i$ and $\overline{f}^i$ are short for $\underline{f}^i\left(\boldsymbol{x}^*\right)$ and $\overline{f}^i\left(\boldsymbol{x}^*\right)$, respectively

Given $\Delta Y_{desired}$, find the maximum allowable $|\Delta x_i|$'s such that $|\Delta Y| \leq \Delta Y_{desired}$. In other words,

$$|\Delta Y| = \left| \sum_{j=1}^{p} \left(X_j^l + X_j^u \right) \Delta x_j \right| \ \leq \ \Delta Y_{desired} \tag{5.24}$$

where X_j^l and X_j^u are given by 5.22 and 5.23, respectively.

In general, especially for applications in physical systems, Δx_j's are bounded. Consider the bounds as additional constraints, i.e.,

$$\begin{aligned} -\beta_1 &\leq \Delta x_1 \leq \alpha_1 \\ -\beta_2 &\leq \Delta x_2 \leq \alpha_2 \\ &\vdots \\ -\beta_p &\leq \Delta x_p \leq \alpha_p \end{aligned} \tag{5.25}$$

Next, formulate (5.24) as a multi-objective optimization problem subject to constraints given in (5.25). Since Δx_j's are small, the optimization problem with constrains is expressed as follows:

$$\begin{cases} \text{Maximize :} & |\Delta x_1|, \cdots, |\Delta x_p| \\ & \\ \text{Subject to :} & \left| \sum_{j=1}^{p} \left(X_j^l + X_j^u \right) \Delta x_j \right| \leq \Delta Y_{desired} \\ & -\beta_1 \leq \Delta x_1 \leq \alpha_1 \\ & -\beta_2 \leq \Delta x_2 \leq \alpha_2 \\ & \vdots \\ & -\beta_p \leq \Delta x_p \leq \alpha_p \end{cases}$$

Next, extend the above analysis for the case when q pairs of training (sampling) points is given for a FLS. First, for each pair, calculate the maximum input deviation of each input resulting in less than or equal to the desired output deviation. Suppose the maximum allowable deviation of the jth input corresponding to the ith training pair is $\Delta^i x_j$ ($i = 1, \cdots, q$ and $j = 1, \cdots, p$). Then, the maximum acceptable deviation of each individual input of the FLS is obtained by getting the minimum of the obtained input deviations of all the training pairs corresponding to that specific input, i.e.,

$$\Delta x_j = \min \left\{ \Delta^i x_j \mid i = 1, \cdots, q \right\} \tag{5.26}$$

The robustness problem in this case is re-formulated as

Problem 1 *Given q pairs of training data and* $\Delta Y_{desired} \geq 0$*, Maximize* $|\Delta x_i|$*'s such that* $|\Delta Y| \leq \Delta Y_{desired}$ *subject to the constraints in* (5.25).

The above optimization problem can be solved for Δx_j's using software such as Matlab optimization toolbox. Note that designers have the freedom to select the bounds on Δx_j's as long as they are feasible for a given physical problem.

Next, define a robustness measure that enables us to quantitatively compare the robustness performance of FLSs. According to the definition of robustness, for a given $\Delta Y_{desired}$, $\max \Delta x_i$'s are sought. Thus, the robustness measure is defined as

$$R_i \equiv \frac{\max |\Delta x_i|}{\Delta Y_{desired}} \tag{5.27}$$

where the index i corresponds to the ith input. This index is used as a base line for determining the robustness of a given FLS, and also as a metric to compare different FLSs' robustness based on (5.24).

Note 1 *By removing FOUs, the above results hold for the robustness analysis of T1 FLS and hence making the proposed methodology more general.*

Now the expressions for two special cases of FLSs with one or two inputs are derived, which are common in many applications.

- For a system with a single input, (5.18) is simplified to

$$\begin{aligned} \Delta Y &= \Delta x \sum_{i=1}^{M} \left[m\underline{w}^i(x^*) \frac{\partial y^i(x^*)}{\partial x} + m y^i(\boldsymbol{x}^*) \frac{\partial \underline{w}^i(x^*)}{\partial x} \right] \\ &+ \Delta x \sum_{i=1}^{M} \left[n\overline{w}^i(x^*) \frac{\partial y^i(x^*)}{\partial x} + n y^i(x^*) \frac{\partial \overline{w}^i(x^*)}{\partial x} \right] \end{aligned} \tag{5.28}$$

  Solving for Δx yields

$$\begin{aligned} \Delta x &= \frac{\Delta Y}{\sum_{i=1}^{M} \left[m\underline{w}^i(x^*) \frac{\partial y^i(x^*)}{\partial x} + m y^i(\boldsymbol{x}^*) \frac{\partial \underline{w}^i(x^*)}{\partial x} \right]} \\ &+ \frac{\Delta Y}{\sum_{i=1}^{M} \left[n\overline{w}^i(x^*) \frac{\partial y^i(x^*)}{\partial x} + n y^i(x^*) \frac{\partial \overline{w}^i(x^*)}{\partial x} \right]} \end{aligned} \tag{5.29}$$

  The expression given in (5.29) is interesting because it directly outputs Δx which can be easily used to compute maximum allowable deviation in the input.

- For a system with two inputs,

$$\begin{aligned}\Delta Y &= \Delta x_1 \sum_{i=1}^{M}\left[m\underline{w}^i\left(x^*\right)\frac{\partial y^i\left(x^*\right)}{\partial x_1}+my^i\left(x^*\right)\frac{\partial \underline{w}^i\left(x^*\right)}{\partial x_1}+n\overline{w}^i\left(x^*\right)\frac{\partial y^i\left(x^*\right)}{\partial x_1}+ny^i\left(x^*\right)\frac{\partial \overline{w}^i\left(x^*\right)}{\partial x_1}\right]\\ &+ \Delta x_2 \sum_{i=1}^{M}\left[m\underline{w}^i\left(x^*\right)\frac{\partial y^i\left(x^*\right)}{\partial x_2}+my^i\left(x^*\right)\frac{\partial \underline{w}^i\left(x^*\right)}{\partial x_2}+n\overline{w}^i\left(x^*\right)\frac{\partial y^i\left(x^*\right)}{\partial x_2}+ny^i\left(x^*\right)\frac{\partial \overline{w}^i\left(x^*\right)}{\partial x_2}\right]\end{aligned} \tag{5.30}$$

So far, a first-order approximation has been used in the robustness analyses. To obtain more accurate approximation of the output of an FLS, $\underline{w}^i\left(\boldsymbol{x}^*\right)$ and $y\left(\boldsymbol{x}^*\right)$ can be expanded around small $\Delta\left(\boldsymbol{x}^*\right)$ up to their second derivatives. It is easy to see that, for systems with more than two inputs, the computational effort will be considerable, and hence, satisfying the corresponding inequalities might be intractable. Therefore, in this thesis, the expansion of higher order derivatives for only FLSs with one and two inputs are considered. Details of the derivations are summarized in Appendix C.2.

5.3 Upper bound of the output deviation

In this section, an upper bound of $|\Delta Y|$ is found. This result will be of interest to designers for estimating the maximum deviations expected for a given problem. To do so, it is only needed to find the upper bounds of ΔA and ΔB. Here, the derivation for the upper bound of $|\Delta A|$ is presented (similar analysis can be performed to obtain ΔB). $|\Delta A|$ can be written as

$$\begin{aligned}|\Delta A| &= |m|\left|\sum_{i=1}^{M}\underline{w}^i\left(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*\right)y_i(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*)-\sum_{i=1}^{M}\underline{w}^i\left(\boldsymbol{x}^*\right)y_i(\boldsymbol{x}^*)\right|\\ &\leq |m|\left|\sum_{i=1}^{M}\underline{w}^i\left(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*\right)y_i(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*)\right|+|m|\left|\sum_{i=1}^{M}\underline{w}^i\left(\boldsymbol{x}^*\right)y_i(\boldsymbol{x}^*)\right|\end{aligned} \tag{5.31}$$

Observe that

$$\begin{aligned}\left|\sum_{i=1}^{M}\underline{w}^i\left(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*\right)y_i(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*)\right| &\leq \sum_{i=1}^{M}\left|\underline{w}^i\left(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*\right)y_i(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*)\right|\\ &\leq \sum_{i=1}^{M}\left|\underline{w}^i\left(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*\right)\right|.\left|y_i(\boldsymbol{x}^*+\Delta\boldsymbol{x}^*)\right|\end{aligned} \tag{5.32}$$

Since $0 \leq \underline{w}^i \leq 1$, (5.32) can be expressed as

$$\left| \sum_{i=1}^{M} \underline{w}^i \left(\boldsymbol{x}^* + \Delta \boldsymbol{x}^* \right) y_i(\boldsymbol{x}^* + \Delta \boldsymbol{x}^*) \right| \leq \sum_{i=1}^{M} |y_i(\boldsymbol{x}^* + \Delta \boldsymbol{x}^*)| \tag{5.33}$$

Similarly

$$\left| \sum_{i=1}^{M} \underline{w}^i \left(\boldsymbol{x}^* \right) y_i(\boldsymbol{x}^*) \right| \leq \sum_{i=1}^{M} |y_i(\boldsymbol{x}^*)| \tag{5.34}$$

Therefore,

$$|\Delta A| \;\; \leq \;\; |m| \left(\sum_{i=1}^{M} |y_i(\boldsymbol{x}^* + \Delta \boldsymbol{x}^*)| + \sum_{i=1}^{M} |y_i(\boldsymbol{x}^*)| \right) \tag{5.35}$$

Suppose

$$\begin{aligned} y^1_{max} &= \max_i \{ |y_i(\boldsymbol{x}^*)| \} \\ y^2_{max} &= \max_i \{ |y_i(\boldsymbol{x}^* + \Delta \boldsymbol{x}^*)| \} \end{aligned} \tag{5.36}$$

Using (5.36), ΔA is bounded from above by

$$|\Delta A| \;\; \leq \;\; |m| \left(y^1_{max} + y^2_{max} \right) \tag{5.37}$$

In a similar fashion,

$$|\Delta B| \;\; \leq \;\; |n| \left(y^1_{max} + y^2_{max} \right) \tag{5.38}$$

Therefore, the upper bound of $|\Delta Y|$ is given by

$$|\Delta Y| \;\; \leq \;\; (|m| + |n|) \left(y^1_{max} + y^2_{max} \right) \tag{5.39}$$

For a T1 FLS, it is true that

$$|\Delta Y_{T1}| \;\; \leq \;\; \left(y^1_{max} + y^2_{max} \right) \tag{5.40}$$

In summary, the maximum allowable deviations for IT2 and T1 FLSs are respectively given by

$$|\Delta Y_{IT2}|_{max} \;\; = \;\; \Big(|m| + |n| \Big) \left(y^1_{max} + y^2_{max} \right) \tag{5.41}$$

$$|\Delta Y_{T1}|_{max} \;\; = \;\; \left(y^1_{max} + y^2_{max} \right) \tag{5.42}$$

Observe that for a designed FLS, the TSK parameters are known and hence $|\Delta Y_{IT2}|_{max}$ can be simply determined because y^1_{max} and y^2_{max} are polynomial functions of the consequent TSK parameters.

5.4 Procedure to design robust FLSs

This section presents a procedure to design robust TSK FLSs.

First, a T1 TSK FLS is designed using available software tools such as ANFIS/Matlab. The model-structure of the developed T1 is exploited for the design of an IT2 TSK.

Next, define the error vector as the difference between the sampling (training) output vector, $\boldsymbol{Y}_{sampling}$, and the T1 output vector, $\boldsymbol{Y}_{T1}$, as $\boldsymbol{e}_{T1} \equiv \boldsymbol{Y}_{sampling} - \boldsymbol{Y}_{T1}$. The total error for the T1 TSK, e_{Type-1}, is defined as

$$e_{Type-1} \equiv (\boldsymbol{e}_{T1})^T . \boldsymbol{e}_{T1} \tag{5.43}$$

In Chapter 3 it was shown that the TSK consequent parameters of an IT2 TSK is given by the following expression:

$$\boldsymbol{\theta} = (m\underline{\boldsymbol{\phi}} + n\overline{\boldsymbol{\phi}})^{\dagger} \boldsymbol{Y}_{IT2} \tag{5.44}$$

where $\boldsymbol{\theta}$ contains the TSK consequent parameters, $\boldsymbol{Y}_{IT2}$ is the output of the IT2, m, n are the tuning parameters, $\underline{\boldsymbol{\phi}}$ and $\overline{\boldsymbol{\phi}}$ are the functions of lower and upper fired rules, respectively, and $(m\underline{\boldsymbol{\phi}}+n\overline{\boldsymbol{\phi}})^{\dagger}$ is the pseudo-inverse of $(m\underline{\boldsymbol{\phi}}+n\overline{\boldsymbol{\phi}})$. Similar to the definitions presented earlier for the T1, define the error vector as the difference between the sampling (training) output vector, $\boldsymbol{Y}_{sampling}$, and the IT2 output vector, $\boldsymbol{Y}_{IT2}$, as $\boldsymbol{e}_{IT2} \equiv \boldsymbol{Y}_{sampling} - \boldsymbol{Y}_{IT2}$. The total error for T2 TSKs, e_{Type-2}, is defined as

$$e_{Type-2} \equiv (\boldsymbol{e}_{T2})^T . \boldsymbol{e}_{T2} \tag{5.45}$$

Expressions given in (5.43) and (5.45) are used for the design of robust FLSs.

Subsequently, the IT2 MFs must be designed. To reduce the complexity in the design of IT2 MFs, the T1 TSK MFs generated by ANFIS are used as a starting point and only the FOUs are designed to extend the structure to an IT2 FLS. Note that this work uses mainly Gaussian MFs, simply because these functions and their derivatives are smooth. As well, in comparison to trapezoidal MFs, Gaussian MFs require fewer parameters to be identified. Hence, the mean of IT2 MFs are kept the same as the means of their T1 MFs and only lower and upper standard deviations of each MF are designed. To design the FOUs (or lower and upper MFs) as well as the IT2 tuning parameters, m and n, an optimization algorithm such as Genetic Algorithm (GA) can be used to output the optimum values of these parameters.

Finally, the robustness design is formulated through a constraint optimization problem. To ensure the robustness objective has been achieved, (5.43) is used as a basis for the design. The procedure for the design of robust TSK FLSs is as follows:

1. Design a T1 (using ANFIS for example) for a given training data to approximate the objective function, f, and calculate the output error, i.e., e_{Type-1} using (5.43)

2. Use the T1 MF parameters obtained in step 1 in conjunction with an optimization technique like GA to design the FOUs as well as the optimum m and n to design an IT2 FLS that approximates f. Next, construct the following constrained optimization problem:

   $$\begin{cases} \text{Maximize :} & R_i \\ \\ \text{Subject to :} & |e_{Type-2}| \leq |e_{Type-1}| \end{cases} \tag{5.46}$$

   Note that to perform this step, one needs to perform iterations for obtaining the consequent TSK parameters given by (5.44).

3. If the above constraint problem is feasible, save the parameters of the IT2; otherwise, T1

The feasibility of the constraint optimization problem can be easily verified using software such as Matlab. If the problem is feasible, then IT2 is selected as it reveals more robustness; otherwise, T1 will be preferred.

To summarize the process of designing robust FLSs:

- Design a T1 FLS to approximate a given static/dynamic function for the known sampling points.
- Assign the maximum allowable output deviation, i.e., $\Delta Y_{desired}$.
- Calculate the maximum allowable input deviations using (5.26), and check whether the system is able to handle uncertainties (due to sensor noise, disturbance, etc.) associated with those of Δx's . If not, go back to the previous step and set a reasonable $\Delta Y_{desired}$ for the given system.
- Use the procedure explained above for the design of a robust FLS (could be either T1 or IT2).

Section 5.5 provides several examples to illustrate the application of the proposed robustness analysis in several modeling and control case studies.

5.5 Examples

In this section, several examples are presented to demonstrate how the methodology presented in Section 5.4 is used for the analysis and design of the FLS robustness.

The error performance index (EPI) (in percentage) is defined as

$$\%EPI \equiv \frac{|e_{T1}| - |e_{IT2}|}{|e_{T1}|} \times 100 \tag{5.47}$$

Similarly, to compare the robustness performance of the two FLSs qualitatively, define

$$\%RPI_i \equiv \frac{\max|\Delta x_i|_{IT2} - \max|\Delta x_i|_{T1}}{\max|\Delta x_i|_{T1}} \times 100 \tag{5.48}$$

where RPI_i is the robustness performance improvement and i corresponds to the ith input.

In the following examples, Gaussian functions are considered in the design of T1 and IT2 MFs. The parameters of these MFs are characterized by standard deviation, σ, and mean, μ, respectively. The mean of the MFs are kept the same for T1 and IT2. The absolute values of the maximum input deviations are reported in all the following results. Figure 5.1 shows the IT2 FOU and standard deviations of lower and upper MFs, σ_l, and σ_u. Also note that ΔY_{des} is short for $Y_{desired}$.

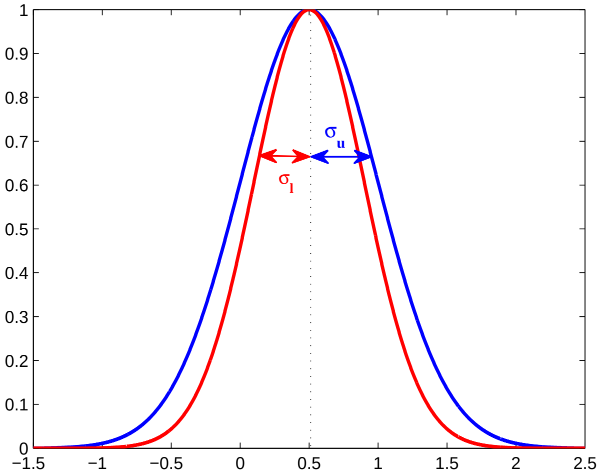


Figure 5.1: IT2 MFs and FOU.

Example 1: Function approximation- single-input

Two examples of single-input function are presented in Parts A and B.

Part A

In this example, T1 and IT2 TSK FLSs are generated to approximate a single-input function given by $f(x) = x + \frac{1}{x^2}$. The purpose of this example is to demonstrate different design capabilities of IT2 and T1. First, a sample training data of 75 I/O pairs, chosen randomly in $[1, 3]$, was used to develop a T1 TSK model with four Gaussian MFs, which captures the function very accurately. Using these results, the optimum parameters of the IT2 TSK FLS are designed. The parameters of the MFs of the T1 and IT2 TSK FLSs are shown in Table 5.1. The IT2 TSK FLS tuning parameters, m and n, are 0.88 and 1.1, respectively.

Table 5.1: Membership function parameters of T1 and IT2 TSK FLSs- first example: Part A.

	MF1	*MF2*	*MF3*	*MF4*
T1	$\mu = 0.8375$ $\sigma = 0.1111$	$\mu = 1.4490$ $\sigma = 0.1876$	$\mu = 2.1370$ $\sigma = 0.1231$	$\mu = 2.7950$ $\sigma = 0.1196$
IT2	$\sigma_l = 0.0978$ $\sigma_u = 0.1222$	$\sigma_l = 0.1651$ $\sigma_u = 0.2064$	$\sigma_l = 0.1083$ $\sigma_u = 0.1354$	$\sigma_l = 0.1052$ $\sigma_u = 0.1316$

For a single input system, using (5.29), $|\Delta x|$ is expressed as

$$|\Delta x| = \frac{|\Delta y|}{|\Delta|} \tag{5.49}$$

where Δ is the denominator of (5.29) and is given by

$$\Delta = \sum_{i=1}^{M} \left[m\underline{w}^i\left(x^*\right) \frac{\partial y^i\left(x^*\right)}{\partial x} + m y^i\left(\boldsymbol{x}^*\right) \frac{\partial \underline{w}^i\left(x^*\right)}{\partial x} \right] + \sum_{i=1}^{M} \left[n\overline{w}^i\left(x^*\right) \frac{\partial y^i\left(x^*\right)}{\partial x} + n y^i\left(x^*\right) \frac{\partial \overline{w}^i\left(x^*\right)}{\partial x} \right] \tag{5.50}$$

Since Δ depends only on the training/sampling points, the maximum allowable deviation in x is achieved when $|\Delta y| = |\Delta y_{des}|$, i.e.,

$$|\Delta x|_{max} = \frac{|\Delta y_{des}|}{|\Delta|} \tag{5.51}$$

(5.51) can be expressed as

$$\frac{|\Delta x|_{max}}{|\Delta y_{des}|} = \frac{1}{|\Delta|} \tag{5.52}$$

From (5.52) it is easy to see that $R = \frac{|\Delta x|_{max}}{|\Delta y_{des}|}$ is only a function of the sampling points. Hence, the robustness index, R, is constant for different desired output deviations.

To perform robustness analysis, different desired output deviations, ΔY_{des}, are considered and the results are compared for both T1 and IT2. Simulation results in terms of function approximation error, e (based on the root of sum of squares) and error performance improvement (EPI) are summarized in Table 5.2.

While both T1 and IT2 FLSs satisfy the required constraint on the maximum output deviation, IT2 reveals fewer output errors for different values of ΔY_{des}.

Table 5.2: T1 and IT2 performances for different desired output deviations- first example: Part A.

	$\Delta Y_{des} = 0.01$		$\Delta Y_{des} = 0.05$		$\Delta Y_{des} = 0.1$		$\Delta Y_{des} = 0.15$		$\Delta Y_{des} = 0.2$	
	e	EPI	e	EPI	e	PI	e	EPI	e	PI
T1	0.3617	–	0.3596	–	0.3587	–	0.3506	–	0.3302	–
IT2	0.0940	74.00%	0.1014	71.80%	0.0942	73.74%	0.0839	76.08%	0.0837	74.64%

Based on (5.27), the robustness indices of T1 and IT2 are computed as $R_{T1} = 0.5073$ and $R_{IT2} = 0.5915$, respectively. Hence, the robustness performance improvement of IT2 over T1 is calculated as $RPI = 16.60\%$. Thus, in this example, the IT2 proves to be a more robust FLS than the T1 FLS.

Part B

Similar to Part A, the robustness performance of T1 and IT2 TSK FLSs are compared in the approximation problem of a single-input function given by $f(x) = x^2 \ln(x)$. A sample training data of 75 I/O pairs, chosen randomly in $[1, 2]$, was used to develop a T1 TSK model with five Gaussian MFs. The parameters of the MFs of the T1 and IT2 TSK FLSs are shown in Table 5.3. The tuning parameters, m and n, are chosen as 0.007 and 0.29, respectively.

Table 5.3: Membership function parameters of T1 and IT2 TSK FLSs- first example: Part B.

	MF1	*MF2*	*MF3*	*MF4*	*MF5*
T1	$\mu = 1.0270$ $\sigma = 0.2340$	$\mu = 1.5160$ $\sigma = 0.2555$	$\mu = 2.0100$ $\sigma = 0.2655$	$\mu = 2.4990$ $\sigma = 0.2657$	$\mu = 2.9840$ $\sigma = 0.2417$
IT2	$\sigma_l = 0.2059$ $\sigma_u = 0.2363$	$\sigma_l = 0.2248$ $\sigma_u = 0.2581$	$\sigma_l = 0.2336$ $\sigma_u = 0.2682$	$\sigma_l = 0.2338$ $\sigma_u = 0.2684$	$\sigma_l = 0.2127$ $\sigma_u = 0.2241$

Next, robustness analysis for different desired output deviations, ΔY_{des}, is performed. The results for both T1 and IT2 are summarized in Table 5.4.

The robustness indices are $R_{T1} = 0.1098$ and $R_{IT2} = 0.1109$, respectively, and hence, $RPI = 1.02\%$. Results show that the robustness performance of the two FLSs are comparable.

Table 5.4: T1 and IT2 performances for different desired output deviations- first example: Part B.

	$\Delta Y_{des} = 0.01$		$\Delta Y_{des} = 0.05$		$\Delta Y_{des} = 0.1$		$\Delta Y_{des} = 0.15$		$\Delta Y_{des} = 0.2$	
	e	EPI	e	EPI	e	PI	e	EPI	e	PI
T1	0.0531	–	0.0534	–	0.0542	–	0.0556	–	0.0576	–
IT2	0.0481	9.34%	0.0485	9.13%	0.0497	8.35%	0.0517	7.03%	0.0546	5.23%

Table 5.5: T1 and IT2 performances for different desired output deviations- first example: Part B.

	$\Delta Y_{des} = 0.01$		$\Delta Y_{des} = 0.05$		$\Delta Y_{des} = 0.1$		$\Delta Y_{des} = 0.15$		$\Delta Y_{des} = 0.2$	
	e	EPI	e	EPI	e	PI	e	EPI	e	PI
T1	0.0436	–	0.0445	–	0.0463	–	0.0488	–	0.0521	–
IT2	0.0385	11.54%	0.0389	12.68%	0.0399	13.95%	0.0415	14.98%	0.0439	15.70%

Next, the results of T1 and IT2 FLSs ARE tabulated when the number of sampling points is increased to 100.

The robustness indices are $R_{T1} = 0.1093$ and $R_{IT2} = 0.1091$, respectively, and hence, $RPI = -0.22\%$. It is observed that, for this example, when the number of sampling points increases the robustness of both FLSs decreases. Intuitively, the reason lies in the increase in the number of constraints required to satisfy the robust criteria, which, in effect, will limit the tolerance of the system to input deviations.

Similar to the first example, both FLSs prove to be mapping the function accurately as well revealing robust behaviors.

Example 2: Function approximation-two inputs

This example investigates the robustness of a two-input system. Similar to the previous examples, two fuzzy models, T1 and IT2 TSK, for 100 I/O pairs are developed to approximate the function given by $f(x_1, x_2) = \frac{\sin(x1)}{x1}.\frac{\sin(x2)}{x2}$ for $x_1, x_2 \in [-10, 10]$ as shown in Figure 5.2. The MF parameters of the two FLSs are given in Table 5.6. For the GA parameters, a population size of 20 with a crossover rate of 0.8 is considered, and the tuning parameters are $m = 0.3$ and $n = 0.1$.

Next, the responses of T1 and IT2 TSK FLSs to different desired output deviations are investigated. Results are shown in Table 5.7, in which T1 and IT2 reveal very comparable performances. In some cases, T1 outputs a slightly improved robustness than IT2; however, the output error generated by IT2 is slightly smaller than T1.

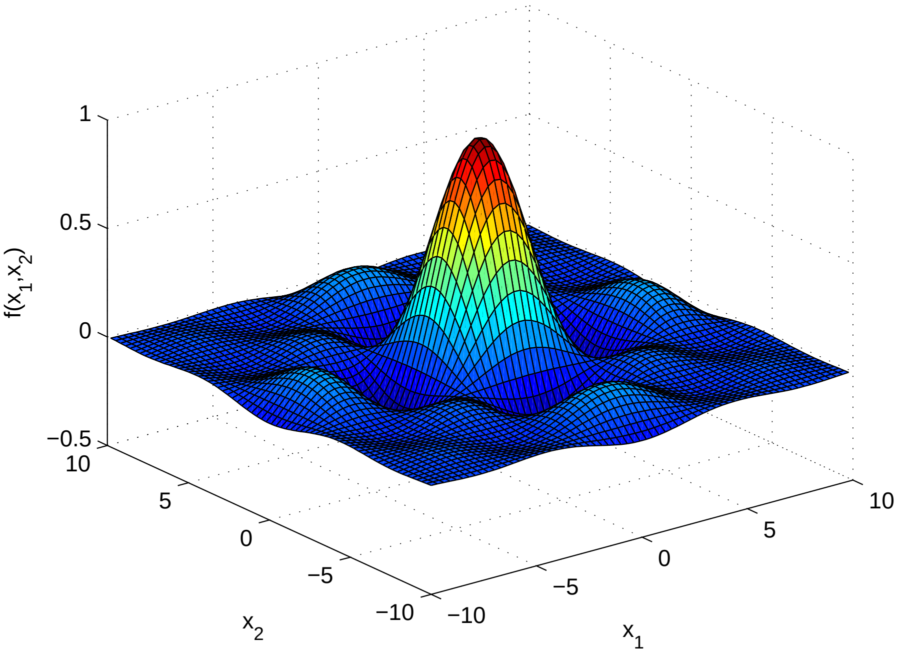


Figure 5.2: Plot of $f(x_1, x_2) = \frac{\sin(x_1)}{x_1}.\frac{\sin(x_2)}{x_2}$.

Table 5.6: Membership function parameters of T1 and IT2 TSK FLSs- second example.

	First Input				Second Input			
	MF1	*MF2*	*MF3*	*MF4*	*MF1*	*MF2*	*MF3*	*MF4*
T1	$\sigma = 2.9290$ $\mu = -9.7910$	$\sigma = 1.7180$ $\mu = -3.9740$	$\sigma = 2.2510$ $\mu = 1.1110$	$\sigma = 3.7520$ $\mu = 8.0220$	$\sigma = 3.0390$ $\mu = -9.3970$	$\sigma = 2.7360$ $\mu = -3.2000$	$\sigma = 2.8220$ $\mu = 2.8460$	$\sigma = 3.4400$ $\mu = 9.2040$
IT2	$\sigma_u = 2.9290$ $\sigma_l = 2.6361$	$\sigma_u = 1.7180$ $\sigma_l = 1.5462$	$\sigma_u = 2.2510$ $\sigma_l = 2.0259$	$\sigma_u = 3.7520$ $\sigma_l = 3.3768$	$\sigma_u = 3.0390$ $\sigma_l = 2.7351$	$\sigma = 2.7360$ $\sigma_l = 2.4624$	$\sigma_u = 2.8220$ $\sigma_l = 2.5398$	$\sigma_u = 3.4400$ $\sigma_l = 3.0960$

Example 3: Function approximation- three inputs

Two T1 and IT2 TSK FLSs are developed to approximate a nonlinear function given by $f(x, y, z) = (1 + x^{-0.5} + y^{-1} + z^{-1.5})^2$ for $x, y, z \in [1, 6]$. Each input has four MFs, resulting in a FLS with a total of 64 rules. As in the other examples, robustness analyses on T1 and IT2 TSK FLSs for different output deviations are performed. The MF parameters are shown in Tables 5.8 and 5.9. For the GA parameters, a population size of 20 with a crossover rate of 0.8 is considered (default values), and the optimized tuning parameters are $m = 0.1$ and $n = 0.1$.

Next the performance of the T1 and IT2 FLSs are compared for different desired output deviations. Results are tabulated in Table 5.10 for different output deviations.

Table 5.7: T1 and IT2 performances for different desired output deviations- second example.

$\Delta Y_{des} = 0.01$

	e	$EPI\%$	R_1	R_2	$RPI_1\%$	$RPI_2\%$
T1	0.1491	–	0.0114	0.0100	–	–
IT2	0.1476	1.0231%	0.0166	0.0100	45.61	0
$\Delta Y_{des} = 0.03$						
T1	0.1491	–	0.0034	0.0033	–	–
IT2	0.1449	2.81%	0.0034	0.0034	0	3.03
$\Delta Y_{des} = 0.05$						
T1	0.1499	–	0.0066	0.9400	–	–
IT2	0.1450	3.26%	0.0081	0.5120	44.26	-45.53
$\Delta Y_{des} = 0.07$						
T1	0.1615	–	0.0021	2.8471	–	–
IT2	0.1534	5.04%	0.0067	2.6029	> 100	-9.38
$\Delta Y_{des} = 0.1$						
T1	0.1763	–	1	3	–	–
IT2	0.1688	4.28%	1	3	0	0

Table 5.8: Membership function parameters for T1 and IT2 TSK FLSs- third example.

	First Input				Second Input			
	MF1	*MF2*	*MF3*	*MF4*	*MF1*	*MF2*	*MF3*	*MF4*
T1	$\sigma = 0.6754$ $\mu = 1.125$	$\sigma = 0.6192$ $\mu = 2.796$	$\sigma = 0.5991$ $\mu = 4.346$	$\sigma = 0.6619$ $\mu = 5.968$	$\sigma = 0.0.6994$ $\mu = 1.063$	$\sigma = 0.6424$ $\mu = 2.701$	$\sigma = 0.6647$ $\mu = 4.299$	$\sigma = 0.6463$ $\mu = 5.91$
IT2	$\sigma_u = 0.8780$ $\sigma_l = 0.6079$	$\sigma_u = 0.8050$ $\sigma_l = 0.5573$	$\sigma_u = 0.7788$ $\sigma_l = 0.5392$	$\sigma_u = 0.8605$ $\sigma_l = 0.5957$	$\sigma_u = 0.9092$ $\sigma_l = 0.6295$	$\sigma_u = 0.8351$ $\sigma_l = 0.5782$	$\sigma_u = 0.8641$ $\sigma_l = 0.5982$	$\sigma_u = 0.8402$ $\sigma_l = 0.5817$

Table 5.9: Membership function parameters of the third input for T1 and IT2 TSK FLSs- third example.

	Third Input			
	MF1	*MF2*	*MF3*	*MF4*
T1	$\sigma = 0.9375$ $\mu = 4.299$	$\sigma = 2.635$ $\mu = 1.539$	$\sigma = 0.7054$ $\mu = 4.27$	$\sigma = 0.6386$ $\mu = 5.935$
IT2	$\sigma_u = 0.7066$ $\sigma_l = 0.4891$	$\sigma_u = 0.7703$ $\sigma_l = 0.5333$	$\sigma_u = 0.9170$ $\sigma_l = 0.6349$	$\sigma_u = 0.8302$ $\sigma_l = 0.5747$

Both T1 and IT2 approximate the function accurately and exhibit robust performances for various $\Delta Y_{desired}$. It is important to note that the IT2 consistently reveals smaller errors for all the given desired outputs. Moreover, the maximum output deviation for each input is larger then its T1 counterpart and, hence, it demonstrates a more robust behavior. For this example, when both minimization of the output error and robustness are considered concurrently, IT2 will be a better choice for modeling.

Table 5.10: T1 and IT2 performances for different desired output deviations: third example.

$\Delta Y_{des} = 0.01$

	e	$EPI\%$	R_1	R_2	R_3	$RPI_1\%$	$RPI_2\%$	$RPI_3\%$
T1	0.0015	–	0.0100	0.2600	0.0100	–	–	–
IT2	0.0012	15.56%	0.0105	0.2600	0.0125	5	0	25
$\Delta Y_{des} = 0.03$								
T1	0.0123	–	0.0027	0.2533	0.0025	–	–	–
IT2	0.0108	12.36%	0.0029	0.2600	0.0028	7.41	2.6451	12
$\Delta Y_{des} = 0.05$								
T1	0.0325	–	8.9848e-4	0.2480	0.0011	–	–	–
IT2	0.0285	12.43%	0.0014	0.2560	0.0014	55.82	3.23	27.27
$\Delta Y_{des} = 0.07$								
T1	0.0621	–	0.0014	0.2471	0.0010	–	–	–
IT2	0.0533	14.08%	0.0025	0.2514	0.0013	78.57	1.74	30
$\Delta Y_{des} = 0.1$								
T1	0.1222	–	0.0020	0.2440	0.0016	–	–	–
IT2	0.1040	14.90%	0.0027	0.2470	0.0051	35	1.23	>100

Example 4: Real-time identification for control (single input)

In many real applications, fuzzy logic has been used as an estimator or identifier of a portion of a plant [151] (the controller may or may not be a fuzzy controller). An FLS, used as an estimator, due to its universal approximation property, should be able to capture the dynamics of the plant relatively well. More importantly, the FLS must be robust to input deviations. In other words, small deviations in the FLS inputs should not cause a large deviation and hence error in the output. Significant deviations of the FLS output will change the closed-loop performance and, in severe cases, could even lead to an unstable system. In Examples 4 and 5, it is demonstrate how robustness of T1 and IT2 affect the performance of different nonlinear plants. Example 4 consists of two parts, and Example 5 deals with a more complex plant.

Part A

Consider the following plant whose dynamic model is given by

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + f(u(k)) \tag{5.53}$$

where $u(k)$ and $y(k)$ are input and output, respectively, and $f(u(k))$ is an unknown function of the following form

$$f(u(k)) = \sin(u(k)). \cos(u(k)) \tag{5.54}$$

The unknown function is meant to be identified by an FLS. Similar to other examples, for 50 I/O sampling points, two FLS models, T1 and IT2, are developed and have three MFs. The parameters of the MFs have been tabulated in Table 5.11. The tuning parameters of the IT2 TSK FLS are $m = 0.2$ and $n = 0.01$.

Table 5.11: Membership function parameters of T1 and IT2 TSK FLSs- fourth example: Part A.

	MF1	*MF2*	*MF3*
T1	$\sigma = 0.3172$ $\mu = 0.5511$	$\sigma = 0.3660$ $\mu = 1.2800$	$\sigma = 0.3592$ $\mu = 1.9280$
IT2	$\sigma_l = 0.2633$ $\sigma_u = 0.4758$	$\sigma_l = 0.3038$ $\sigma_u = 0.5490$	$\sigma_l = 0.2981$ $\sigma_u = 0.5388$

Table 5.12 provides the outputs of T1 and IT2 FLSs for different desired output deviations. It can be easily seen that the IT2 FLS produces outputs with smaller error than the T1 FLSs.

Table 5.12: T1 and IT2 performances for different desired output deviations- fourth example: Part A.

	$\Delta Y_{des} = 0.01$		$\Delta Y_{des} = 0.05$		$\Delta Y_{des} = 0.07$		$\Delta Y_{des} = 0.1$	
	e	EPI	e	EPI	e	EPI	e	EPI
T1	0.0032	–	0.0138	–	0.0181	–	0.0234	–
IT2	0.0004	77.08%	0.0032	76.59%	0.0042	76.70%	0.0055	76.59%

The robustness indices of T1 and IT2 are computed as $R_{T1} = 0.9984$ and $R_{IT2} = 1.0016$, respectively, and hence, the robustness performance improvement of IT2 over T1 is calculated as $RPI = 0.32\%$. Observe that the maximum output deviations of the two systems are very comparable.

Next, the performances of both FLSs are evaluated when the input, $u(k)$, is deviated by $0.01u(k)$, $0.05u(k)$, and $0.1u(k)$, respectively. This comparison is of great importance in identification of the plant since a robust identifier should ultimately capture the plant as accurately as possible, a crucial step for control design. To compare the output of the two TSK models, simulations were carried out for $k = 0$ to $k = 250$. Results are summarized in Table 5.13, where for all different deviations, $0.1\%\Delta u(k) - 10\%\Delta u(k)$, the IT2 FLS captures the dynamic term $f(u(k))$ more accurately than the T1 FLS. Therefore, in this example, the IT2 clearly proves to be a more robust system for identification of the plant.

Part B

Consider another plant with the following dynamics:

$$y(k+1) = 0.4y(k) + f(u(k)) \tag{5.55}$$

Table 5.13: T1 and IT2 performances for different deviated inputs- fourth example: Part A.

	$\Delta u(k) = 0.01$		$\Delta u(k) = 0.05$		$\Delta u(k) = 0.1$	
	e_1	EPI_1	e_2	EPI_2	e_3	EPI_3
T1	0.0037	–	0.0135	–	0.0368	–
IT2	0.0012	67.32%	0.0047	65.21%	0.0138	62.45%

where $u(k)$ and $y(k)$ are input and output, respectively, and $f(u(k))$ is an unknown function with the following form

$$f(u) = \frac{u(k)}{1 + u^3(k)} \tag{5.56}$$

Suppose the initial condition is $y(0) = 0.01$. The parameters of the MFs are tabulated in Table 5.14. For 75 I/O sampling points, T1 and IT2 TSK FLSs are designed. The tuning parameters of the IT2 TSK FLS are $m = 0.3$ and $n = 0.1$.

Table 5.14: Membership function parameters of T1 and IT2 TSK FLSs- fourth example: Part B.

	MF1	*MF2*	*MF3*	*MF4*
T1	$\mu = -0.9754$ $\sigma = 0.3573$	$\mu = -0.3357$ $\sigma = 0.3797$	$\mu = 0.3566$ $\sigma = 0.3908$	$\mu = 0.9632$ $\sigma = 0.3774$
IT2	$\sigma_l = 0.3287$ $\sigma_u = 0.3716$	$\sigma_l = 0.3493$ $\sigma_u = 0.3949$	$\sigma_l = 0.3595$ $\sigma_u = 0.4064$	$\sigma_l = 0.3472$ $\sigma_u = 0.3925$

Table 5.15 provides the outputs of T1 and IT2 FLSs for different desired output deviations and shows both FLSs are capable of estimating the unknown function relatively well.

Table 5.15: T1 and IT2 performances for different desired output deviations- fourth example: Part B.

	$\Delta Y_{des} = 0.05$		$\Delta Y_{des} = 0.1$		$\Delta Y_{des} = 0.15$	
	e	EPI	e	EPI	e	EPI
T1	0.0185	–	0.0220	–	0.0278	–
IT2	0.0145	16.61%	0.0184	16.57%	0.0231	16.83%

The robustness indices are $R_{T1} = 0.2945$ and $R_{IT2} = 0.2921$, respectively, and hence, $RPI = -0.81\%$, which show the comparable robust behaviors of T1 and IT2.

Similar to Part A, simulations are performed for $k = 0$ to $k = 100$ and results are summarized in Table 5.16. For all different deviations, $5\%\Delta u(k) - 15\%\Delta u(k)$, the IT2 FLS identifies $f(u(k))$ with a higher accuracy than the T1 FLS.

Table 5.16: T1 and IT2 performances for different deviated inputs- fourth example: Part B.

	$\Delta u(k) = 0.05$		$\Delta u(k) = 0.1$		$\Delta u(k) = 0.15$	
	e_1	EPI_1	e_2	EPI_2	e_3	EPI_3
T1	0.0029	–	0.0039	–	0.0055	–
IT2	0.0026	10.64%	0.0035	11.39%	0.0046	15.35%

For large input deviations, the estimation error is an important index and therefore designing an FLS that can better approximate the unknown variables/parts is essential, especially for real-time applications.

Example 5: Real-time identification for control- a more complex problem

This example presents a more complex identification problem than any previously presented. The plant to be identified is given by

$$y(k+1) = f(u_1(k)) + g(u_2(k)) + 0.2y(k) \tag{5.57}$$

where f and g are two unknown nonlinear functions (two nonlinear FLCs). These functions are of the form, $f = \frac{1}{1+u_1^2(k)}$ and $g = \exp(-u_2^2(k))$, respectively. Two FLSs are designed for identifying f and g. Similar to the previous examples, two T1 TSK FLSs for f and g for 100 I/O sampling points are generated. Subsequently, their corresponding IT2 TSK FLSs are developed. The MFs of the FLSs are given in Tables 5.17 and 5.18. The tuning parameters for f and g are $m_f = 0.023$, $n_f = 0.41$, $m_g = 0.001$, and $n_g = 0.5$, respectively.

Table 5.17: Membership function parameters of T1 and IT2 TSK FLSs models estimating the function f- fifth example.

	MF1	*MF2*	*MF3*	*MF4*	*MF5*	*MF6*
T1	$\sigma = 0.2533$ $\mu = 0.0475$	$\sigma = 0.2666$ $\mu = 0.4030$	$\sigma = 0.2665$ $\mu = 0.8036$	$\sigma = 0.2483$ $\mu = 1.1980$	$\sigma = 0.2248$ $\mu = 1.5920$	$\sigma = 0.1953$ $\mu = 1.9870$
IT2	$\sigma_l = 0.2280$ $\sigma_u = 0.3293$	$\sigma_l = 0.2399$ $\sigma_u = 0.3466$	$\sigma_l = 0.2399$ $\sigma_u = 0.3465$	$\sigma_l = 0.2235$ $\sigma_u = 0.3228$	$\sigma_l = 0.2023$ $\sigma_u = 0.2922$	$\sigma_l = 0.1758$ $\sigma_u = 0.2539$

Next, performances of the FLSs are evaluated in the identification of the two functions, f and g. To make the comparison of the two FLSs easier, the results are tabulated as shown in Table 5.19 in which Δf_{des} and Δg_{des} denote the desired output deviation of each corresponding function.

The robustness indices and RPI of the T1 and IT2 corresponding to the function f are calculated as $R_{T1} = 1.5382$, $R_{IT2} = 1.5396$, and $RPI = 0.1\%$, respectively. The

Table 5.18: Membership function parameters of T1 and IT2 TSK FLSs models estimating the function g- fifth example.

	MF1	*MF2*	*MF3*	*MF4*	*MF5*	*MF6*
T1	$\sigma = 0.1041$ $\mu = 0.0318$	$\sigma = 0.1222$ $\mu = 0.2252$	$\sigma = 0.1325$ $\mu = 0.4130$	$\sigma = 0.1303$ $\mu = 0.6070$	$\sigma = 0.1137$ $\mu = 0.7935$	$\sigma = 0.0946$ $\mu = 0.9922$
IT2	$\sigma_l = 0.0989$ $\sigma_u = 0.1457$	$\sigma_l = 0.1161$ $\sigma_u = 0.1711$	$\sigma_l = 0.1259$ $\sigma_u = 0.1855$	$\sigma_l = 0.1238$ $\sigma_u = 0.1824$	$\sigma_l = 0.1080$ $\sigma_u = 0.1592$	$\sigma_l = 0.0899$ $\sigma_u = 0.1324$

Table 5.19: T1 and IT2 performance for different desired output deviations of function f: fifth example.

	$\Delta Y_{des} = 0.05$		$\Delta Y_{des} = 0.1$		$\Delta Y_{des} = 0.15$	
	e	EPI	e	EPI	e	EPI
T1	0.0012	–	0.0033	–	0.0073	–
IT2	1.7380e-4	85.20%	6.4067e-4	80.40%	0.0016	77.88%

Table 5.20: T1 and IT2 performance for different desired output deviations of function g: fifth example.

	$\Delta Y_{des} = 0.05$		$\Delta Y_{des} = 0.1$		$\Delta Y_{des} = 0.15$	
	e	EPI	e	EPI	e	EPI
T1	0.0015	–	0.0057	–	0.0201	–
IT2	5.3622e-4	63.63%	0.0028	50.95%	0.0088	59.94%

robustness indices and RPI of T1 and IT2 for the function g are computed as $R_{T1} = 1.1573$, $R_{IT2} = 1.1655$, and $RPI = 1\%$, respectively, and hence the maximum allowable input deviations of both FLSs are comparable for all ΔY_{des}. However, as the results indicate, the IT2 FLS demonstrates its superior capability in modeling both functions. In specific terms, the smaller the deviations, the better the IT2 performance.

Finally, simulations are performed to compare the output of the plant, y, given by (5.57) when the inputs u_1 and u_2 are deviated by $0.1\% - 15\%$. The simulation results are provided in Table 5.21. Both T1 and IT2 successfully reveal robust behavior in allowing the input deviations, while IT2 appears to be a more robust FLS for this problem.

Table 5.21: T1 and IT2 performances for different input deviations- fifth example.

	$\Delta u_1(k), \Delta u_2(k) = 0.01$		$\Delta u_1(k), \Delta u_2(k) = 0.1$		$\Delta u_1(k), \Delta u_2(k) = 0.15$	
	e_1	EPI_1	e_2	EPI_2	e_3	EPI_3
T1	0.5297	–	0.6896	–	0.8007	–
IT2	0.3236	38.91%	0.6210	9.95%	0.5576	30.36%

In summary, the examples presented in this section demonstrate the potential of both

T1 and IT2 FLSs in the design of robust static and dynamic systems. More importantly, the case studies conducted herein prove that IT2 FLSs, due to their more flexible structure compared to T1 FLSs, produce relatively small approximation errors. For larger sampling data, the robustness of T1 and IT2 usually decreases, and both tend to reveal comparable robust behavior. However, this observation is not always the case. Depending on the nature of a nonlinear function, T1 or IT2 might perform more robustly than to the other.

5.6 Conclusion

This chapter presented a rigorous mathematical analysis of the robustness of T1 and IT2 FLSs. The robustness of FLSs was formulated for the case when the sampling points are subjected to small deviations (through first order Taylor approximation). An efficient procedure was introduced that can be used for the design of robust FLSs. Several examples verified the effectiveness of the proposed methodologies, and it was concluded that, because of their flexible structures, IT2 FLSs reveal a great potential that can be exploited in the design of robust FLSs. In future work, it would be interesting to investigate the effect of higher order approximations on the robustness of FLSs and thus to provide a metric to measure the trade-off between accuracy and computational complexity. In addition, other MFs, not only Gaussian MFs, are suggested for further investigation in the design of robust FLSs.

Chapter 6

Conclusions and Future Work

This chapter presents conclusions and future work of the thesis in two sections. In each section, for the ease of reading, the contribution of each chapter is provided in a separate subsection.

6.1 Conclusions

6.1.1 Stability Analysis

In Chapter 3, five novel inference engines for the design and analyses of IT2 FLSs were proposed. The inference mechanisms all have closed-forms and thus facilitate the analysis of IT2 TSK FLCs. It was shown that, among the proposed inference engines, the most general one is derivable from WM UBs. Using the general inference engine, this dissertation focused on the following control applications when (1) both plant and controller use A2-C0 TSK models, and (2) the plant uses T1 TS and the controller uses IT2 TS models. In both cases, sufficient conditions for the stability of the closed-loop system were derived. Furthermore, novel linear matrix inequalities-based algorithms were developed. These results further enable the stability analysis and the design of stable IT2 TS (or TSK) FLCs that are of interest to control engineers. To validate the effectiveness of the new inference engines, numerical analyses were presented, and it was concluded that an IT2 TS FLCS using the proposed inference engine has the potential to outperform its T1 TSK counterpart. In addition, because of the simple nature of the proposed inference engines, they are easy to implement in real-time control systems.

6.1.2 MRR Control

Chapter 4 presented the design and implementation of a novel IT2 TSK fuzzy logic control strategy for tracking applications in MRRs. It was shown that the proposed controller can be effectively applied to MRRs with guaranteed UUB stability. Furthermore, due to the adjustment law, a minimal effort is required to design the IT2 TSK FLC parameters. It was also demonstrated that the developed controller can considerably outperform a well-known linear controller in terms of tracking performances. Furthermore, the varying-parameter controller demonstrated its ability to handle the dynamic coupling between the joints of two serial MRR configurations. Therefore, the proposed control methodology can be easily adopted for MRRs with uncertain and varying dynamic parameters.

6.1.3 Robustness

A rigorous mathematical analysis of the robustness of T1 and IT2 FLSs was presented in Chapter 5. The robustness of FLSs was formulated as a constraint multi-objective optimization problem. Subsequently, an efficient procedure was introduced to facilitate the design of robust FLSs. Several examples demonstrate the effectiveness of the proposed methodologies. It was shown that both T1 and IT2 FLSs reveal robust behaviors, and preference one or the other, in general, is application-dependant. However, through several case studies, it was concluded that IT2 FLSs, because of their flexible structures, reveal a great potential for approximating static and dynamic functions. For larger sampling data, a decrease in T1 and IT2 FLSs robustness is expected, making both FLSs exhibit comparable robust behavior. However, this observation is problem dependent. Based on the nature of a nonlinear function, T1 or IT2 might demonstrate comparably improved robustness. The developed approach is of practical value when FLSs are used for modeling and control applications.

6.2 Future Work

6.2.1 Stability

The developed stability conditions for IT2 TSK FLCs are sufficient. Thus, for problems in which a common Lyapunov function is not found, no conclusion can be made about the stability of the closed-loop system. Further investigation into relaxed stability conditions might facilitate the design of stable controllers. In addition, it would be interesting to perform a stability analysis of MIMO IT2 TS FLCSs when both plant and controller are IT2 TS. Finally, future work can be also geared toward the development of uncertainty bounds, not considered in the proposed inference engines, but which could further facilitate the use of the proposed inference engines in the design of IT2 TSK FLCSs.

6.2.2 MRR

A more systematic design methodology for tuning the parameters of PD and PD-like IT2 TSK FLCs should be developed to enhance facilitating their use in control applications. In addition, a more efficient algorithm is suggested to be developed to investigate the effect of IT2 TSK FLC with more rules and find the trade off between the number of rules and improvement in the tracking performance of MRRs. Furthermore, to validate the proposed control methodology on more complex MRRs, involving high dynamic couplings among joints, it would be useful if future experiments considered a system with more degrees of freedom. The noticeable fluctuations observed in the controller efforts of the second joint of the MRR, in severe cases, excite some modes of the systems; thus, resulting in undesirable motions. As for future work, it is important to consider the chattering effects especially when a heavy payload is used in the tracking applications. It would be also interesting to consider developing an adaptive controller to handle varying masses of the payload. Lastly, parallel processing of lower and upper portions of the control effort of the IT2 TSK FLC will reduce the computational time required to implement the controller. A code enabling parallel computing should be developed and its performance improvement observed.

6.2.3 Robustness

A First-order Taylor's series was used to approximate the output of a deviated FLS. Although robust IT2 TSK FLSs can be designed using this assumption, it would be interesting to investigate the second-order derivative approximations and compare their results with the ones presented in this dissertation. In addition, to find the maximum allowable input deviations, a systematic algorithm should be developed to design the parameters of IT2 FLSs, i.e., MFs, as well as the tuning parameters, hence further enabling the design of robust mechanisms that can be used for various applications.

Other MFs, not considered in this dissertation, are suggested for further investigation of FLS robustness. In addition, the robustness of FLSs can be investigated with different t-norm operators. Finally, examination and comparison of the performance of FLSs with different fuzzy operators might be a fruitful topic for investigation.

APPENDICES

Appendix A: Supplementary details of stability derivations

A.1 Components of $\boldsymbol{Z}$ in (3.73)

Components of $\boldsymbol{Z}$, namely, $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, and $\boldsymbol{Z}_3$ are as follows:

$$
\begin{aligned}
\boldsymbol{Z}_1 = & \left[\frac{m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{m\sum_{j=1}^{M}\underline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]+\left[\frac{m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{n\sum_{j=1}^{M}\overline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] \\
& +\left[\frac{n\sum_{i=1}^{M}\overline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\overline{f}^i}\boldsymbol{P}\frac{m\sum_{j=1}^{M}\underline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]+\left[\frac{n\sum_{i=1}^{M}\overline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\overline{f}^i}\boldsymbol{P}\frac{n\sum_{j=1}^{M}\overline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
\boldsymbol{Z}_2 = {} & \left[\frac{m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mm'\sum_{j=1}^{M}\sum_{l=1}^{Q}\underline{f}^j\underline{v}^l\boldsymbol{B}_{j,l}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] \\
& + \left[\frac{m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mn'\sum_{j=1}^{M}\sum_{l=1}^{Q}\underline{f}^j\overline{v}^l\boldsymbol{B}_{j,l}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] \\
& + \left[\frac{m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nm'\sum_{j=1}^{M}\sum_{l=1}^{Q}\overline{f}^j\underline{v}^l\boldsymbol{B}_{j,l}}{k_2\sum_{i=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] + \left[\frac{m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nn'\sum_{j=1}^{M}\sum_{l=1}^{Q}\overline{f}^j\overline{v}^l\boldsymbol{B}_{j,l}}{k_2\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] \\
& + \left[\frac{n\sum_{i=1}^{M}\overline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\overline{f}^i}\boldsymbol{P}\frac{mm'\sum_{j=1}^{M}\sum_{l=1}^{Q}\underline{f}^j\underline{v}^l\boldsymbol{B}_{j,l}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] + \left[\frac{n\sum_{i=1}^{M}\overline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\overline{f}^i}\boldsymbol{P}\frac{mn'\sum_{j=1}^{M}\sum_{l=1}^{Q}\underline{f}^j\overline{v}^l\boldsymbol{B}_{j,l}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] \\
& + \left[\frac{n\sum_{i=1}^{M}\overline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\overline{f}^i}\boldsymbol{P}\frac{nm'\sum_{j=1}^{M}\sum_{l=1}^{Q}\overline{f}^j\underline{v}^l\boldsymbol{B}_{j,l}}{k_2\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] + \left[\frac{n\sum_{i=1}^{M}\overline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\overline{f}^i}\boldsymbol{P}\frac{nn'\sum_{j=1}^{M}\sum_{l=1}^{Q}\overline{f}^j\overline{v}^l\boldsymbol{B}_{j,l}}{k_2\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] \\
& + \left[mm'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{m\sum_{j=1}^{M}\underline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] + \left[mm'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{n\sum_{j=1}^{M}\overline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] \\
& + \left[\frac{mn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{m\sum_{j=1}^{M}\underline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] + \left[\frac{mn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{n\sum_{j=1}^{M}\overline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] \\
& + \left[\frac{nm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{m\sum_{j=1}^{M}\underline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] + \left[\frac{nm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{n\sum_{j=1}^{M}\overline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] \\
& + \left[\frac{nn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{m\sum_{j=1}^{M}\underline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right] + \left[\frac{nn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{n\sum_{j=1}^{M}\overline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
\boldsymbol{Z}_3 = &\left[\frac{mm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\underline{v}^q\boldsymbol{B}_{j,q}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{mm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mn'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\overline{v}^q\boldsymbol{B}_{j,q}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{mm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\overline{f}^j\underline{v}^q\boldsymbol{B}_{j,q}}{k_2\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{mm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nn'\sum_{j=1}^{M}\sum_{q=1}^{Q}\overline{f}^j\overline{v}^q\boldsymbol{B}_{j,q}}{k_2\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{mn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\underline{v}^q\boldsymbol{B}_{j,q}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{mn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mn'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\overline{v}^q\boldsymbol{B}_{j,q}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{mn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\overline{f}^j\underline{v}^q\boldsymbol{B}_{j,q}}{k_2\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{mn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nn'\sum_{j=1}^{M}\sum_{q=1}^{Q}\overline{f}^j\overline{v}^q\boldsymbol{B}_{i,q}}{k_2\sum_{j=1}^{M}\overline{f}^i}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{nm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\underline{v}^q\boldsymbol{B}_{i,q}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{nm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mn'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\overline{v}^q\boldsymbol{B}_{j,q}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{nm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\overline{f}^j\underline{v}^q\boldsymbol{B}_{j,q}}{k_2\sum_{i=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{nm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nn'\sum_{j=1}^{M}\sum_{q=1}^{Q}\overline{f}^j\overline{v}^q\boldsymbol{B}_{j,q}}{k_2\sum_{i=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{nn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\underline{v}^q\boldsymbol{B}_{j,q}}{k_1\sum_{j=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{nn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mn'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\overline{v}^q\boldsymbol{B}_{j,q}}{k_1\sum_{i=1}^{M}\underline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{nn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\overline{f}^j\underline{v}^q\boldsymbol{B}_{j,q}}{k_2\sum_{i=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]\\
&+\left[\frac{nn'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\overline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{nn'\sum_{j=1}^{M}\sum_{q=1}^{Q}\overline{f}^j\overline{v}^q\boldsymbol{B}_{j,q}}{k_2\sum_{j=1}^{M}\overline{f}^j}-\frac{1}{36}\boldsymbol{P}\right]
\end{aligned}
\tag{A.3}
$$

A.2 Details of stability derivations

This section presents additional details of the stability conditions (3.79) and (3.81). Consider the first bracketed term of $\boldsymbol{Z}_2$ in (A.2) and denote it as $\boldsymbol{Z}_{2,1}$, i.e.,

$$\boldsymbol{Z}_{2,1} \equiv \frac{m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i^T}{\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\underline{v}^q\boldsymbol{B}_{j,q}}{k_1\sum_{j=1}^{M}\underline{f}^j} - \frac{1}{36}\boldsymbol{P} \tag{A.4}$$

Similarly, denote the ninth bracketed term of $\boldsymbol{Z}_2$ as $\boldsymbol{Z}_{2,9}$

$$\boldsymbol{Z}_{2,9} \equiv mm'\frac{\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{m\sum_{j=1}^{M}\underline{f}^j\boldsymbol{A}_j}{\sum_{j=1}^{M}\underline{f}^j} - \frac{1}{36}\boldsymbol{P} \tag{A.5}$$

Multiply both sides of (A.4) and (A.5) by $k_1\sum_{i=1}^{M}\underline{f}^i\sum_{j=1}^{M}\underline{f}^j$. It is easy to see that

$$\begin{aligned}
(\boldsymbol{Z}_{2,1}+\boldsymbol{Z}_{2,9})\left(k_1\sum_{i=1}^{M}\underline{f}^i\sum_{j=1}^{M}\underline{f}^j\right) &= m\sum_{i=1}^{M}\underline{f}^i\boldsymbol{A}_i^T\boldsymbol{P}mm'\sum_{j=1}^{M}\sum_{l=1}^{Q}\underline{f}^j\underline{v}^l\boldsymbol{B}_{j,l} \\
&\quad + mm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}\boldsymbol{P}m\sum_{j=1}^{M}\underline{f}^j\boldsymbol{A}_j^T - \frac{1}{18}\boldsymbol{P}\left(k_1\sum_{i=1}^{M}\underline{f}^i\sum_{j=1}^{M}\underline{f}^j\right) \\
&= \sum_{i,j=1}^{M}\underline{f}^i\underline{f}^j\left[\sum_{l=1}^{Q}m^2m'\left[\underline{v}^l\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{B}_{j,l}+\underline{v}^l\boldsymbol{B}_{j,l}\boldsymbol{P}\boldsymbol{A}_i^T\right]-\sum_{l=1}^{Q}\underline{v}^l\frac{1}{18}\boldsymbol{P}\right] \\
&= \sum_{i,j=1}^{M}\underline{f}^i\underline{f}^j\left\{\sum_{l=1}^{Q}\underline{v}^l\left[m^2m'\left(\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{B}_{j,l}+\boldsymbol{B}_{j,l}\boldsymbol{P}\boldsymbol{A}_i^T\right)-\frac{1}{18}\boldsymbol{P}\right]\right\}
\end{aligned} \tag{A.6}$$

It can be shown that, for the remaining terms of $\boldsymbol{Z}_2$, similar expressions can be obtained. Hence, to make all terms of $\boldsymbol{Z}_2$ negative definite, the following criteria must be satisfied:

$$b\boldsymbol{A}_i^T\boldsymbol{P}\boldsymbol{B}_{j,l}+b\boldsymbol{B}_{i,l}^T\boldsymbol{P}\boldsymbol{A}_j-\frac{1}{18}\boldsymbol{P}<\boldsymbol{0} \tag{A.7}$$

where $i,j=1,2,...,M$, and $l=1,2,...,Q$ and

$$b=\{m^2m', mnm', m^2n', mnn', n^2m', n^2n'\} \tag{A.8}$$

Next, consider the first term of $\boldsymbol{Z}_3$ in (A.3) and denote it as $\boldsymbol{Z}_{3,1}$, i.e.,

$$\boldsymbol{Z}_{3,1} \equiv \frac{mm'\sum_{i=1}^{M}\sum_{l=1}^{Q}\underline{f}^i\underline{v}^l\boldsymbol{B}_{i,l}^T}{k_1\sum_{i=1}^{M}\underline{f}^i}\boldsymbol{P}\frac{mm'\sum_{j=1}^{M}\sum_{q=1}^{Q}\underline{f}^j\underline{v}^q\boldsymbol{B}_{j,q}}{k_1\sum_{j=1}^{M}\underline{f}^j} - \frac{1}{36}\boldsymbol{P} \tag{A.9}$$

Multiply both sides of (A.9) by $\left(k_1^2 \sum_{i=1}^M \underline{f}^i \sum_{j=1}^M \underline{f}^j\right)$. It is straightforward to show [using (3.59)]

$$\left(k_1^2\sum_{i=1}^M \underline{f}^i \sum_{j=1}^M \underline{f}^j\right)\boldsymbol{Z}_{3,1} = mm'\sum_{i=1}^M\sum_{l=1}^Q \underline{f}^i\underline{v}^l \boldsymbol{B}_{i,l}^T \boldsymbol{P} mm' \sum_{j=1}^M\sum_{q=1}^Q \underline{f}^j\underline{v}^q\boldsymbol{B}_{i,l} - \frac{1}{36}\boldsymbol{P}\left(k_1^2 \sum_{i=1}^M \underline{f}^i\sum_{j=1}^M\underline{f}^j\right)$$

$$= \sum_{i,j=1}^M \underline{f}^i\underline{f}^j\left[\sum_{l,q=1}^Q m^2 {m'}^2 \underline{v}^l\underline{v}^q \boldsymbol{B}_{i,l}^T\boldsymbol{P}\boldsymbol{B}_{j,q} - \sum_{l,q=1}^Q \underline{v}^l\underline{v}^q\frac{1}{36}\boldsymbol{P}\right] =$$

$$\sum_{i,j=1}^M \underline{f}^i\underline{f}^j\left\{\sum_{l,q=1}^Q \underline{v}^l\underline{v}^q\left[m^2 {m'}^2 \boldsymbol{B}_{i,l}^T\boldsymbol{P}\boldsymbol{B}_{j,q} - \frac{1}{36}\boldsymbol{P}\right]\right\} \quad \text{(A.10)}$$

In a similar fashion, it can be shown that, for the remaining terms of $\boldsymbol{Z}_3$, similar expressions can be obtained. Hence, to make all terms of $\boldsymbol{Z}_3$ negative definite, the following criteria must be satisfied:

$$c\boldsymbol{B}_{i,l}^T\boldsymbol{P}\boldsymbol{B}_{j,q} - \frac{1}{36}\boldsymbol{P} < \boldsymbol{0} \quad \text{(A.11)}$$

where $i, j = 1, 2, ..., M$, and $l, q = 1, 2, ..., Q$, and,

$$c = \left\{m^2m'^2, n^2n'^2, m^2m'n', mnm'n', m^2n'^2, mnn'^2, mnm'^2, n^2m'^2, n^2m'n'\right\} \quad \text{(A.12)}$$

Appendix B: Error trajectories of different controllers tested on the MRR

B.1. First configuration

This appendix contains the position error trajectories of each controller for the first configuration. Figures 1 and 2 demonstrate the real-time position errors for different controllers. The difference in the output error for T1 and IT2 controllers are noticeable (for quantitative comparison please refer to Table 4.3).

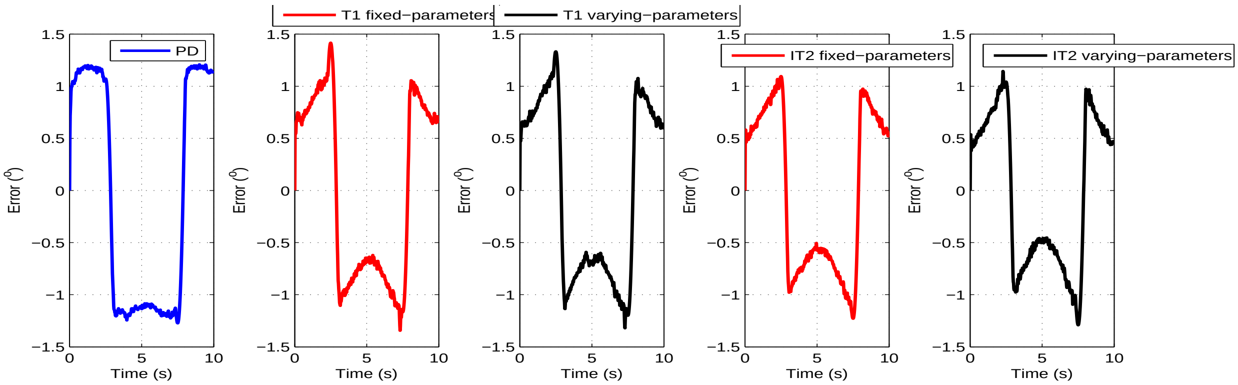


Figure 1: Position errors of different controllers for joint 1: first configuration.

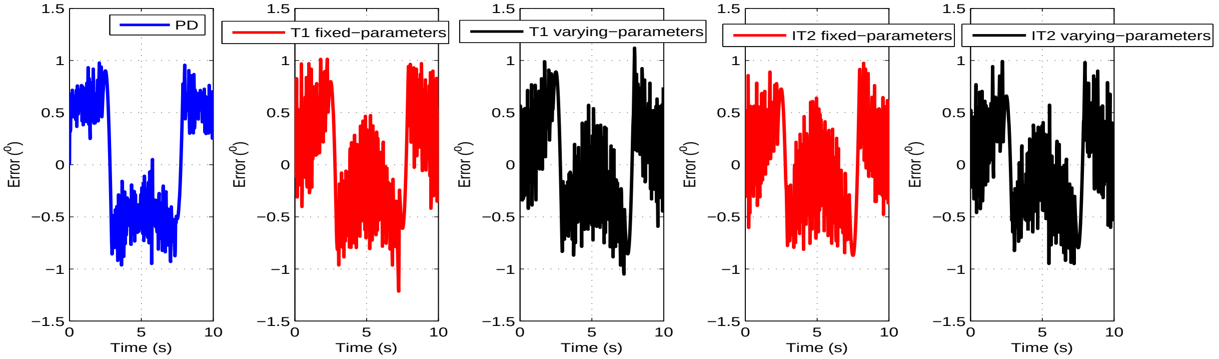


Figure 2: Position errors of different controllers for joint 2: first configuration.

B.2. Second configuration

Here the trajectories of error for the second configuration are shown. Figures 3 and 4 compare the control efforts of the different controllers, for the first and second joints, respectively.

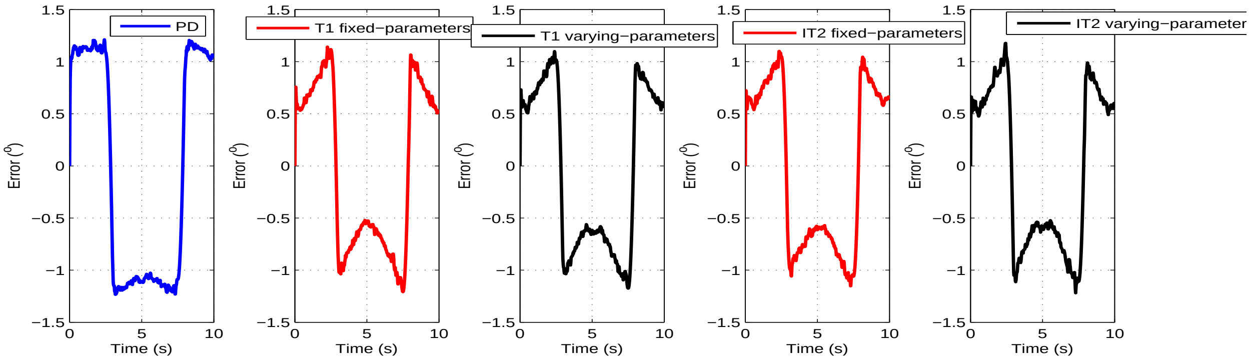


Figure 3: Position errors of different controllers for joint 1: second configuration.

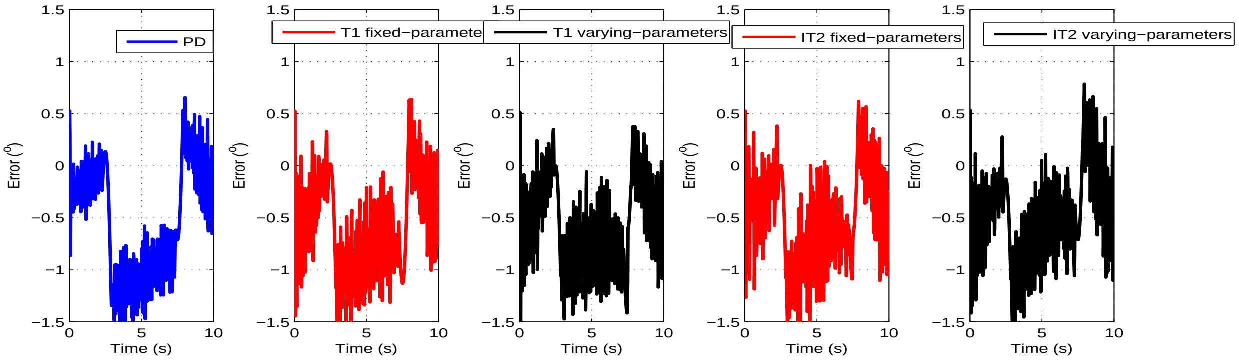


Figure 4: Position errors of different controllers for joint 2: second configuration.

Appendix C: Details of derivation of Chapter5

In this appendix, to simplify the notations, $\underline{w}^i$, $\overline{w}^i$, $\underline{f}^i$, $\overline{f}^i$, and y^i are short for $\underline{w}^i(\boldsymbol{x})$, $\overline{w}^i(\boldsymbol{x})$, $\underline{f}^i(\boldsymbol{x})$, $\overline{f}^i(\boldsymbol{x})$, and $y^i(\boldsymbol{x})$.

C.1. Derivation of $\frac{\partial \underline{w}^i}{\partial x_j}$

In the following, $\frac{\partial \underline{w}^i}{\partial x_j}$ is analytically expressed in terms of the FLS's parameters. First, $\frac{\partial \underline{f}^i}{\partial x_j}$ is derived.

$$\frac{\partial \underline{f}^i}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\underline{\mu}^i(x_1) \star \cdots \star \underline{\mu}^i(x_p)\right] \tag{C.1}$$

where $\star$ is a t-norm. Using algebraic product as the t-norm, (C.1) is calculated as

$$\begin{aligned}\frac{\partial \underline{f}^i}{\partial x_j} &= \underline{\mu}_1^i(x_1)\cdots\underline{\mu}_{j-1}^i(x_{j-1}).\frac{\partial \underline{\mu}_j^i(x_j)}{\partial x_j}.\underline{\mu}_{j+1}^i(x_{j+1})\cdots\underline{\mu}_p^i(x_p) \\ &= \left[\prod_{\substack{k=1\\k\neq j}}^{p-1}\underline{\mu}_k^i(x_k)\right].\frac{\partial \underline{\mu}_j^i(x_j)}{\partial x_j}\end{aligned} \tag{C.2}$$

Now $\frac{\partial}{\partial x_j}\left[\sum_{i=1}^M \underline{f}^i\right]$ is found

$$\frac{\partial}{\partial x_j}\left[\sum_{i=1}^M \underline{f}^i\right] = \frac{\partial}{\partial x_j}\left[\sum_{l=1}^M \underline{f}^l\right] = \sum_{l=1}^M\left(\prod_{\substack{k=1\\k\neq j}}^{p-1}\underline{\mu}_k^l(x_k)\right).\frac{\partial \underline{\mu}_j^l(x_j)}{\partial x_j} \tag{C.3}$$

Using (C.2) and (C.3), $\frac{\partial \underline{w}^i}{\partial x_j}$ is given as follows:

$$\begin{aligned}\frac{\partial \underline{w}^i}{\partial x_j} &= \frac{\partial}{\partial x_j}\left[\frac{\underline{f}^i(\boldsymbol{x}^*)}{\sum_{l=1}^M \underline{f}^l}\right] = \frac{\frac{\partial \underline{f}^i}{\partial x_j}.\left[\sum_{l=1}^M \underline{f}^l\right] - \underline{f}^i.\left(\frac{\partial}{\partial x_j}\left[\sum_{l=1}^M \underline{f}^l\right]\right)}{\left[\sum_{l=1}^M \underline{f}^l\right]^2} \\ &= \frac{\left(\prod_{\substack{k=1\\k\neq j}}^{p-1}\underline{\mu}_k^i(x_k)\right)\frac{\partial \underline{\mu}_j^i(x_j)}{\partial x_j}\left[\sum_{l=1}^M \underline{f}^l\right] - \underline{f}^i.\sum_{l=1}^M\left[\left(\prod_{\substack{k=1\\k\neq j}}^{p-1}\underline{\mu}_k^l(x_k)\right).\frac{\partial \underline{\mu}_j^l(x_j)}{\partial x_j}\right]}{\left[\sum_{l=1}^M \underline{f}^l\right]^2}\end{aligned} \tag{C.4}$$

Similarly,

$$\frac{\partial \overline{w}^i}{\partial x_j} = \frac{\left(\prod_{\substack{k=1\\k\neq j}}^{p-1} \overline{\mu}_k^i(x_k)\right) \frac{\partial \overline{\mu}_j^i(x_j)}{\partial x_j} \left[\sum_{l=1}^{M} \overline{f}^l\right] - \overline{f}^i \cdot \sum_{l=1}^{M} \left[\left(\prod_{\substack{k=1\\k\neq j}}^{p-1} \overline{\mu}_k^l(x_k)\right) \cdot \frac{\partial \overline{\mu}_j^l(x_j)}{\partial x_j}\right]}{\left[\sum_{l=1}^{M} \overline{f}^l\right]^2} \tag{C.5}$$

C.2. Derivation of robustness for higher order derivatives

This section provides the formulation for robustness analysis for FLSs with a single and two inputs.

- For a single input system, (5.18) is modified to

$$\begin{aligned}
\Delta Y &= \Delta x \sum_{i=1}^{M} \left[m\underline{w}^i \frac{\partial y^i}{\partial x} + my^i \frac{\partial \underline{w}^i}{\partial x}\right] + \Delta x \sum_{i=1}^{M} \left[n\overline{w}^i \frac{\partial y^i}{\partial x} + ny^i \frac{\partial \overline{w}^i}{\partial x}\right] \\
&+ (\Delta x)^2 \sum_{i=1}^{M} \left[\frac{m}{2}\underline{w}^i \frac{\partial^2 y^i}{\partial x^2} + \frac{m}{2} y^i \frac{\partial^2 \underline{w}^i}{\partial x^2} + \frac{m}{4} \frac{\partial y^i}{\partial x} \frac{\partial \underline{w}^i}{\partial x}\right] \\
&+ (\Delta x)^2 \sum_{i=1}^{M} \left[\frac{n}{2}\overline{w}^i \frac{\partial^2 y^i}{\partial x^2} + \frac{n}{2} y^i \frac{\partial^2 \overline{w}^i}{\partial x^2} + \frac{n}{4} \frac{\partial y^i}{\partial x} \frac{\partial \overline{w}^i}{\partial x}\right] \qquad \text{(C.6)}
\end{aligned}$$

Define

$$X_1 \equiv \sum_{i=1}^{M} \left[m\underline{w}^i \frac{\partial y^i}{\partial x} + my^i \frac{\partial \underline{w}^i}{\partial x}\right] + \sum_{i=1}^{M} \left[n\overline{w}^i \frac{\partial y^i}{\partial x} + ny^i \frac{\partial \overline{w}^i}{\partial x}\right] \tag{C.7}$$

$$\begin{aligned}
X_2 &\equiv \sum_{i=1}^{M} \left[\frac{m}{2}\underline{w}^i \frac{\partial^2 y^i}{\partial x^2} + \frac{m}{2} y^i \frac{\partial^2 \underline{w}^i}{\partial x^2} + \frac{m}{4} \frac{\partial y^i}{\partial x} \frac{\partial \underline{w}^i}{\partial x}\right] \\
&+ \sum_{i=1}^{M} \left[\frac{n}{2}\overline{w}^i \frac{\partial^2 y^i}{\partial x^2} + \frac{n}{2} y^i \frac{\partial^2 \overline{w}^i}{\partial x^2} + \frac{n}{4} \frac{\partial y^i}{\partial x} \frac{\partial \overline{w}^i}{\partial x}\right] \qquad \text{(C.8)}
\end{aligned}$$

Thus, Using (C.7) and (C.8), (C.6) can be expressed as

$$X_1(\Delta x)^2 + X_2 \Delta x = \Delta Y \tag{C.9}$$

To satisfy the robustness criteria, $|\Delta Y| \leq \Delta Y_{desired}$, is it required that

$$\left|X_1(\Delta x)^2 + X_2 \Delta x\right| \leq \Delta Y_{desired} \tag{C.10}$$

- The simplified expression for ΔY for a two-input system is given by

$$\begin{aligned}
\Delta Y &= \Delta x_1 \sum_{i=1}^{M} \left[m\underline{w}^i \frac{\partial y^i}{\partial x_1} + m y^i \frac{\partial \underline{w}^i}{\partial x_1} + n\overline{w}^i \frac{\partial y^i}{\partial x_1} + n y^i \frac{\partial \overline{w}^i}{\partial x_1} \right] \\
&+ \Delta x_2 \sum_{i=1}^{M} \left[m\underline{w}^i \frac{\partial y^i}{\partial x_2} + m y^i \frac{\partial \underline{w}^i}{\partial x_2} + n\overline{w}^i \frac{\partial y^i}{\partial x_2} + n y^i \frac{\partial \overline{w}^i}{\partial x_2} \right] \\
&+ \Delta x_1 \Delta x_2 \sum_{i=1}^{M} \left[m\underline{w}^i \frac{\partial^2 y^i}{\partial x_1 \partial x_2} + m y^i \frac{\partial^2 \underline{w}^i}{\partial x_1 \partial x_2} + n\overline{w}^i \frac{\partial^2 y^i}{\partial x_1 \partial x_2} + n y^i \frac{\partial^2 \overline{w}^i}{\partial x_1 \partial x_2} \right] \\
&+ (\Delta x_1)^2 \sum_{i=1}^{M} \left[\frac{m}{2}\underline{w}^i \frac{\partial^2 y^i}{\partial x_1^2} + \frac{m}{2} y^i \frac{\partial^2 \underline{w}^i}{\partial x_1^2} + \frac{n}{2}\overline{w}^i \frac{\partial^2 y^i}{\partial x_1^2} + \frac{n}{2} y^i \frac{\partial^2 \overline{w}^i}{\partial x_1^2} \right] \\
&+ (\Delta x_2)^2 \sum_{i=1}^{M} \left[\frac{m}{2}\underline{w}^i \frac{\partial^2 y^i}{\partial x_2^2} + \frac{m}{2} y^i \frac{\partial^2 \underline{w}^i}{\partial x_2^2} + \frac{n}{2}\overline{w}^i \frac{\partial^2 y^i}{\partial x_2^2} + \frac{n}{2} y^i \frac{\partial^2 \overline{w}^i}{\partial x_2^2} \right] \qquad \text{(C.11)}
\end{aligned}$$

where the right-hand side is calculated at the point x_1^* and x_2^*. Equation (C.11) can be turned into an inequality as

$$\left| X_1(\Delta x_1)^2 + X_2(\Delta x_2)^2 + X_3 \Delta x_1 x_2 + X_4 \Delta x_1 + X_5 \Delta x_1 \right| \leq \Delta Y_{desired} \qquad \text{(C.12)}$$

where X_i's ($i = 1, \cdots, 5$) are the expressions given by each summation in (C.11), respectively. Matlab can be used to solve (C.12) before which the second-derivatives of $\underline{w}^i$ and $\overline{w}^i$ must be computed.

References

[1] J. M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice-Hall, Upper Saddle River, NJ, first edition, 2001. xi, 1, 3, 4, 5, 10, 11, 15, 16, 17, 18, 20, 64

[2] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965. 1

[3] F. Karry and C. De Silva. *Fuzzy Control and Modeling: Analytical Foundations and Applications*. IEEE press, first edition. 1

[4] R.I. John and S. Coupland. Type-2 fuzzy logic a historical view. *IEEE Computational Intelligence Magazine*, 2(1):57–62, 2007. 1

[5] E. Mamdani. Fuzzy control–a misconception of theory and application. *IEEE Expert*, 9(4):27–28, 1994. 1, 5

[6] J. M. Mendel. Advances in type-2 fuzzy sets and systems. *Information Sciences*, 177(1):84–110, 2007. 1, 4, 5, 10, 22, 23

[7] J. M. Mendel. Type-2 fuzzy sets and systems: An overview. *IEEE Computational Intelligence Magazine*, 2(1):20–29, 2007. 1, 3, 4

[8] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 8(3):199–249, 1975. 3

[9] G. J. Klir and T.A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, Englewood Cliffs, NJ, second edition, 1988. 3

[10] M. Mizumoto and K. Tanaka. Some properties of fuzzy sets of type 2. *Information and Control*, 31:312–340, 1976. 3

[11] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic, first edition, 1980. 3

[12] N. N. Karnik and J. M. Mendel. Centroid of a type-2 fuzzy set. *Information Science*, 132:195–220, 2001. 3, 4

[13] J. M. Mendel, R. I. John, and F. Liu. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, 14(6):808–818, 2006. 4, 16

[14] Q. Liang and J. M. Mendel. Interval type-2 fuzzy logic systems: theory and design. *IEEE Transactions on Fuzzy Systems*, 8(5):535–550, 2000. 4, 16

[15] J. M. Mendel. Computing derivatives in interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 12(1):84–98, 2004. 4

[16] J. Zeng and Z-Q Liu. Type-2 fuzzy sets for pattern classification: A review. In *Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence*, pages 193–200, Honolulu, Hawaii, April 2007. 4

[17] H. Acostaa, D. Wu, and B. M. Forrestc. Fuzzy experts on recreational vessels, a risk modelling approach for marine invasions. *Ecological Modelling*, 221(5):850–863, 2010. 4

[18] H. B. Mitchell. Pattern recognition using type-2 fuzzy sets. *Information Sciences*, 170:409–418, 2005. 4

[19] Q. Liao, N. Li, and S. Li. Type-II T-S fuzzy model-based predictive control. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, China, December 2009. 4

[20] M.H. Fazel Zarandi, B. Rezaeea, I.B. Turksen, and E. Neshata. A type-2 fuzzy rule-based expert system model for stock price analysis. *Expert Systems with Applications*, 36(1):139–154, 2009. 4

[21] J. M. Mendel and F. Liu. Super-exponential convergence of the Karnik–Mendel algorithms for computing the centroid of an interval type-2 fuzzy set. *IEEE Transactions on Fuzzy Systems*, 15(2):309–320, 2007. 4

[22] A. Niewiadomski, J. Kacprzyk, J. Ochelska, and P.S. Szczepaniak. Interval-valued linguistic summaries of databases. *Control and Cybernetics*, 35(2):415–443, 2006. 4

[23] S. Coupland and R. John. A fast geometric method for defuzzification of type-2 fuzzy sets. 2008. IEEE Transactions on Fuzzy Systems, accepted. 4

[24] H. Wu and J. M. Mendel. Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 10(5):622–639, 2002. 4, 8, 22

[25] D. Wu and J. M. Mendel. Enhanced Karnik-Mendel algorithms for interval type-2 fuzzy sets and systems. In *North American Fuzzy Information Processing Society (NAFIPS)*, pages 184–189, San Diego, CA, June 2007. 5

[26] Miguel Melgarejo. A fast recursive method to compute the generalized centroid of an interval type-2 fuzzy set. In *North American Fuzzy Processing Society (NAFIPS)*, pages 190–194, San Diego, CA, June 2007. 5

[27] D. Wu and W.W. Tan. Computationally efficient type-reduction strategies for a type-2 fuzzy logic controller. In *Proceedings of IEEE FUZZ Conference*, pages 353–358, Reno, NV, May 2005. 5

[28] H. Ying. General interval type-2 Mamdani fuzzy systems are universal approximators. In *Proceedings of North American Fuzzy Information Processing Society (NAFIPS)*, pages 1–6, New York City, USA, May 2008. 5

[29] H. Ying. Interval type-2 Takagi-Sugeno fuzzy systems with linear rule consequent are universal approximators. In *Proceedings of North American Fuzzy Information Processing Society (NAFIPS)*, Cincinnati, Ohio, USA, June 2009. 5

[30] J. M. Mendel, F. Liu, and D. Zhai. α-plane representation for type-2 fuzzy sets: theory and applications. *IEEE Transactions on Fuzzy Systems*, 17(5):1189–1207, 2009. 5

[31] J. M. Mendel. Comments on α-plane representation for type-2 fuzzy sets: theory and applications. *IEEE Transactions on Fuzzy Systems*, 18(1):229–230, 2010. 5

[32] S-M. Zhou, R. I. John, F. Chiclana, and J. M. Garibaldi. On aggregating uncertain information by type-2 OWA operators for soft decision making. *International Journal of Intelligent Systems*, 25(6):540–558, 2010. 5

[33] J. M. Mendel. On answering the question"Where do I start in order to solve a new problem involving interval type-2 fuzzy sets?". *Information Sciences*, 179(19):3418–3431, 2009. 5, 11

[34] J. M. Mendel and D. Wu. *Perceptual Computing: Aiding People in Making Subjective Judgments*. John Wiley & IEEE Press, 2010. 5

[35] E. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Machine Studies*, 7(1):1–13, 1975. 5

[36] G. Feng. A survey on analysis and design of model-based fuzzy control systems. *IEEE Transactions on Fuzzy Systems*, 14(5):676–697, 2006. 5, 6

[37] K. Tanaka, H. Yoshida, H. Ohtake, and H. O. Wang. A sum-of-squares approach to modeling and control of nonlinear dynamical systems with polynomial fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 17(4):911–922, 2009. 5, 6

[38] Y. Xia, H. Yang, P. Shi, and M. Fu. Constrained infinite-horizon model predictive control for fuzzy discrete-time systems. *IEEE Transactions on Fuzzy Systems*, 18(2):429–436, 2010. 5

[39] X. Jia, H. Zhang, X. Hao, and N. Zheng. Fuzzy H∞ tracking control for nonlinear networked control systems in T-S fuzzy model. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 39(4):1073–1079, 2009. 5

[40] E. A. Jammeh, F. Martin, C. Wagner, H. Hagras, and M. Ghanbari. Interval type-2 fuzzy logic congestion control for video streaming across IP networks. *IEEE Transactions on Fuzzy Systems*, 17(5):1123–1142, 2009. 5, 7

[41] L-X. Wang. *A Course in Fuzzy Systems and Control*. Prentice-Hall, NJ, 1996. 5

[42] H. Hagras. Type-2 fuzzy control. *IEEE Computational Intelligence Magazine*, 2(1):30–43, 2007. 5, 7, 10

[43] C. Lynch, H. Hagras, and V. Callaghan. Embedded type-2 FLC for real-time speed control of marine and traction diesel engines. In *Proceedings of IEEE FUZZ Conference*, pages 347–352, Reno, NV, May 2005. 5

[44] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems Man and Cybernetics*, 15:116–132, 1985. 6

[45] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1):15–33, 1988. 6

[46] T-S. Li, S-Ch. Tong, and G. Feng. A novel robust adaptive-fuzzy-tracking control for a class of nonlinear multi-input/multi-output systems. *IEEE Transactions on Fuzzy Systems*, 18(1):150–160, 2010. 6

[47] Guang Ren and Zhi-Hong Xiu. Analytical design of Takagi-Sugeno fuzzy control systems. In *American Control Conference*, pages 1733–1738, Portland, OR, June 2005. 6

[48] Chung-Shi Tseng. Model reference output feedback fuzzy tracking control design for nonlinear discrete-time systems with time-delay. *IEEE Transactions on Fuzzy Systems*, 14(1):58–70, 2006. 6

[49] H.K. Lam and F.H.F. Leung. Stability analysis of discrete-time fuzzy-model-based control systems with time delay: Time delay-independent approach. *Fuzzy Sets and Systems*, 159(8):990–1000, 2008. 6

[50] M. Sugeno and T. Taniguchi. On improvement of stability conditions for continuous Mamdani-like fuzzy systems. *IEEE Transactions on Systems Man and Cybernetics –Part B: Cybernetics*, 34(1):120–131, 2004. 6

[51] K. Tanaka and M. Sugeno. Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems*, 45(2):135–156, 1992. 6

[52] H. O. Wang, K. Tanaka, and M. F. Griffin. An appraoch to fuzzy control of nonlinear systems: Stability and design issues. *IEEE Transaction on Fuzzy System*, 4(1):14–23, 1996. 6

[53] S. P. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, first edition, 1994. 6, 36

[54] K. Tanaka and H. O. Wang. *Fuzzy Control Systems Designs and Analysis*. John Wiley and Sons, first edition, 2001. 6, 47

[55] J. Joh, Y. Chen, and R. Langari. On the stability issues of linear Takagi-Sugeno fuzzy models. *IEEE Transaction on Fuzzy System*, 6(3):402–410, 1998. 6

[56] H. K. Lam and F. H. F. Leung. Stability analysis of fuzzy control systems subject to uncertain grades of membership. *IEEE Transaction on Systems Man and Cybernetics*, 26(6):1322–1325, 2005. 6

[57] W.W. Tan and J. Lai. Development of a type-2 fuzzy proportional controller. In *IEEE FUZZ Conference*, pages 1305–1310, Budapest, Hungary, July 2004. 6

[58] L. Baron Q. Ren and M. Balazinski. Type-2 Takagi-Sugeno-Kang fuzzy logic modeling using subtractive clustering. In *Proceedings of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 120–125, Montreal, June 2006. 6

[59] H. Bernal, K. Duran, and M. Melgarejo. A comparative study between two algorithms for computing the generalized centroid of an interval type-2 fuzzy set. In *Proceedings of IEEE FUZZ Conference*, Hong Kong, China, June 2008. 6

[60] C-F. Juang, R-B. Huang, and Y-Y. Lin. A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing. *IEEE Transactions on Fuzzy Systems*, 17(5):1092–1105, 2009. 6

[61] M. E. Yksel and M. Borlu. Accurate segmentation of dermoscopic images by image thresholding based on type-2 fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 17(4):976–982, 2009. 6

[62] D. Wu and W.W. Tan. A type-2 fuzzy logic controller for the liquid-level process. In *IEEE FUZZ Conference*, pages 953–958, Budapest, Hungary, July 2004. 6

[63] H. Hagras. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on Fuzzy Systems*, 2(4):524–539, 2004. 6, 7

[64] E. Tunstel, H. Danny, T. Lippincott, and M. Jamshidi. Adaptive fuzzy-behavior hierarchy for autonomous navigation. In *IEEE International Conference on Robotics and Automation*, pages 829–834, New Mexico, April 1997. 7

[65] A. El Hajjaji. Fuzzy control of industrial mobile robot. *International Journal of Smart Engineering System Design*, 5(2):111–117, 2003. 7

[66] B. Li and C. Zhang. Adaptive fuzzy control for mobile robot obstacle avoidance based on virtual line path tracking. In *IEEE International Conference on Robotics and Biomimetics*, pages 1454–1458, Kunming, China, December 2006. 7

[67] K. C. Wu. Fuzzy interval control of mobile robots. *Computers & Electrical Engineering*, 22(3):211–229, 1996. 7

[68] C. Wagner and H. Hagras. A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots. In *Proceedings of FUZZ-IEEE*, pages 193–198, London, July 2007. 7, 26

[69] S. L. Crdenas, O. Castillo, L. T. Aguilar, and N. Czarez. *Tracking Control for a Unicycle Mobile Robot Using a Fuzzy Logic Controller*. 7

[70] O. Castillo, L. T. Aguilar, N. R. Cazarez-Castro, and S. Cardenas. Systematic design of a stable type-2 fuzzy logic controller. *Journal of Applied Soft Computing*, 8(3):1274–1279, 2008. 7

[71] H. K. Lam and L. D. Seneviratne. Stability analysis of interval type-2 fuzzy-model-based control systems. *IEEE Transactions on Systems Man and Cybernetics*, 38(3):617–628, 2008. 7

[72] H. Hagras. Developing a type-2 FLC through embedded type-1 FLCs. In *Proceedings of IEEE FUZZ Conference*, Hong Kong, June 2008. 7

[73] O. Castillo, P. Melin, O. Montiel, A. Rodriguez-Diaz, and R. Sepulveda. Handling uncertainty in controllers using type-2 fuzzy logic. *Journal of Intelligent Systems*, 14(3):237–262, 2005. 7

[74] P. Lin, C. Hsu, and T. Lee. Type-2 fuzzy logic controller design for buck dc-dc converters. In *Proceedings of IEEE FUZZ Conference*, pages 365–370, Reno, NV, May 2005. 7

[75] C. Lynch, H. Hagras, and V. Callaghan. Using uncertainty bounds in the design of an embedded real-time type-2 neuro-fuzzy speed controller for marine diesel engines. In *Proceedings of IEEE FUZZ Conference*, pages 7217–7224, Vancouver, Canada, July 2006. 7

[76] H. Chaoui and W. Gueaieb. Type-2 fuzzy logic control of a flexible-joint manipulator. *Journal of Intelligent and Robotic Systems*, 51(2):159–186, 2008. 7, 10

[77] M. Y. Hsiao, T. H. Li, J. Z. Lee, C. H. Chao, and S. H. Tsai. Design of interval type-2 fuzzy sliding-mode controller. *Information Sciences*, 178(6):1696–1716, 2008. 7

[78] H. Zhou, H. Ying, and J. Duan. Adaptive control using type-2 fuzzy logic. In *Proceedings of FUZZ-IEEE*, Jeju Island, Korea, August 2009. 7

[79] C-F. Juang and Ch-H. Hsu. Reinforcement interval type-2 fuzzy controller design by online rule generation and Q-value-aided ant colony optimization. *IEEE Transactions on Systems, Man, and Cybernetics– Part B: Cybernetics*, 39(6):1528–1542, 2009. 7

[80] O. Castillo, R. Martnez-Marroqun, P. Melin, F. Valdez, and J.Soria. Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot (to appear). *Information Sciences*, 2010. 7

[81] X. Du and H. Ying. Derivation and analysis of the analytical structures of the interval type-2 fuzzy PI and PD controllers (to appear). *IEEE Transactions on Fuzzy Systems*, 2010. 7

[82] A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita, and S. Kokaji. Automatic locomotion design and experiments for a modular robotic system. *IEEE/ASME Transactions on Mechatronics*, 10(3):314–325, 2005. 8

[83] S. Hara. A smooth switching from power-assist control to automatic transfer control and its application to a transfer machine. *IEEE Transactions on Industrial Electronics*, 54(1):638–650, 2007. 8

[84] J. Miyata, Y. Kaida, and T. Murakami. v-$\dot{\phi}$-coordinate-based power-assist control of electric wheelchair for a caregiver. *IEEE Transactions on Industrial Electronics*, 55(6):2517–2524, 2007. 8

[85] A. Pamecha, I. Ebert-Uphoff, and G. Chirikjian. Useful metrics for modular robot motion planning. *IEEE Transactions on Robotics and Automation*, 13(4):531–545, 1997. 8

[86] I-M. Chen. *Theory and applications of modular reconfigurable robotic system.* PhD thesis, California Institute of Technology, 1994. 8

[87] M. Yim, W. M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian. Modular self-reconfigurable robot systems. *IEEE Robotics and Automation Magazine*, 14(1):43–52, 2007. 8

[88] Pradeep Khosla D. Schmitz and Takeo Kanade. The CMU reconfigurable modular manipulator system. Technical Report CMU-RI-TR-88-07, Robotics Institute, Pittsburgh, PA, May 1988. 8

[89] S. Y. Lim, D. M. Dawson, J. Hu, and M. S. de Queiroz. An adaptive link position tracking controller for rigid-link flexible-joint robots without velocity measurements. *IEEE Transactions on Systems, Man, and Cybernetics– Part B: Cybernetics*, 27(3):412–427, 1997. 9

[90] A. D Luca, B. Siciliano, and L. Zollo. PD control with on-line gravity compensation for robots with elastic joints: Theory and experiments. *Robotica*, 41(10):1809–1819, 2005. 9

[91] M. Tarokh. Decoupled nonlinear three-term controllers for robot trajectory tracking. *IEEE Transactions on Robotics and Automation*, 15(2):369–380, 1999. 9

[92] Z. Li, W. Melek, and C. Clark. Distributed control of modular and reconfigurable robots manipulators. *Robotica*, 27(3):291–302, 2008. 9

[93] J. Kasac, B. Novakovic, D. Majetic, and D. Brezak. Passive finite-dimensional repetitive control of robot manipulators. *IEEE Transactions on Industrial Electronics*, 16(3):570–576, 2008. 9

[94] D. Sun, S. Hu, Shao X, and C. Liu. Global stability of a saturated nonlinear PID controller for robot manipulators. *IEEE Transactions on Industrial Electronics*, 17(4):892–899, 2009. 9

[95] A. Visioli and G. Legnani. On the trajectory tracking control of industrial SCARA robot manipulators. *IEEE Transactions on Industrial Electronics*, 49(1):224–232, 2002. 9

[96] P. Voglewede, A. H. C. Smith, and A. Monti. Dynamic performance of a SCARA robot manipulator with uncertainty using polynomial chaos theory. *IEEE Transactions on Robotics*, 25(1):206–210, 2009. 9

[97] Y.-W. Liang, S.-D. Xu, D.-C. Liaw, and C-C. Chen. A study of TS model-based SMC scheme with application to robot control. *IEEE Transactions on Industrial Electronics*, 55(11):3964–3971, 2008. 9

[98] M. Jin, S. H. Kang, and P. H. Chang. Robust compliant motion control of robot with nonlinear friction using time-delay estimation. *IEEE Transactions on Industrial Electronics*, 55(1):258–269, 2008. 9

[99] L. Mostefai, M. Dena, O. Sehoon, and Y. Hori. Optimal control design for robust fuzzy friction compensation in a robot joint. *IEEE Transactions on Industrial Electronics*, 56(10):3832–3839, 2009. 9

[100] F. L. Lewis, K. Liu, and A. Yesildirek. Neural net robot controller with guaranteed tracking performance. *IEEE Transactions on Neural Networks*, 6(3):703–715, 1995. 9, 68, 72

[101] F. L. Lewis, S. Jagannathan, and A. Yesildirek. *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor and Francis, 1999. 9, 68, 70, 71

[102] A. Ishiguro, T. Furuhashi, S. Okuma, and Y. Uchikawa. A neural network compensator for uncertainties of robotics manipulators. *IEEE Transactions on Industrial Electronics*, 39(6):565–570, 1992. 9

[103] M.-J. Lee and Y.-K. Choi. An adaptive neurocontroller using RBFN for robot manipulators. *IEEE Transactions on Industrial Electronics*, 51(3):711–717, 2004. 9

[104] F. L. Lewis. Neural network feedback control: Work at UTA's automation and robotics research institute. *Intelligent and Robotic Systems*, 48(4):513–523, 2007. 9

[105] X. Ren, F. L. Lewis, and J. Jang. Neural network compensation control for mechanical systems with disturbances. *Automatica*, 45(5):1221–1226, 2009. 9

[106] S. Jung and S. S. Kim. Hardware implementation of a real-time neural network controller with a DSP and an FPGA for nonlinear systems. *IEEE Transactions on Industrial Electronics*, 54(1):265–271, 2007. 9

[107] S. J. Yoo, J. B. Park, and Y. H. Choi. Adaptive output feedback control of flexible-joint robots using neural networks: Dynamic surface design approach. *IEEE Transactions on Neural Networks*, 19(10):1712–1725, 2008. 9

[108] H. Chaoui, P. Sicard, and W. Gueaieb. ANN-based adaptive control of robotic manipulators with friction and joint elasticity. *IEEE Transactions on Industrial Electronics*, 56(8):3174–3187, 2009. 9

[109] L. Wang, T. Chai, and L. Zhai. Neural-network-based terminal sliding-mode control of robotic manipulators including actuator dynamics. *IEEE Transactions on Industrial Electronics*, 56(9):3296–3304, 2009. 9

[110] H. Hu and P. Y. Woo. Fuzzy supervisory sliding-mode and neural network control for robotic manipulators. *IEEE Transactions on Industrial Electronics*, 53(3):929–940, 2006. 9

[111] R.-J. Wai and P.-C. Chen. Robust neural-fuzzy-network control for robot manipulator including actuator dynamics. *IEEE Transactions on Industrial Electronics*, 53(4):1328–1349, 2006. 9

[112] C-S. Chen. Dynamic structure neural-fuzzy networks for robust adaptive control of robot manipulators. *IEEE Transactions on Industrial Electronics*, 55(9):3402–3414, 2008. 9

[113] F. Jatta, G. Legnani, and A. Visioli. An adaptive neurocontroller using rbfn for robot manipulators. *IEEE Transactions on Industrial Electronics*, 53(2):604–613, 2006. 9

[114] G. Casalino and A. Turetta. A computationally distributed self-organizing algorithm for the control of manipulators in the operational space. In *IEEE International Conference on Robotics and Automation*, volume 18, pages 4050–4055, Barcelona, Spain, April 2005. 9

[115] C. J. J. Paredis, H. G. Brown, and P. K. Khosla. A rapidly deployable manipulator system. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1434–1439, Minneapolis, MN, April 1996. 9

[116] G. Liu, S. Abdul, and A. A. Goldenberg. Decentralized robust control of modular and reconfigurable robots with harmonic drive. *Robotica*, 26(1):75–84, 2008. 9

[117] H.F. Ho, Y.K. Wong, and A.B. Rad. Robust fuzzy tracking control for robotic manipulators. *Simulation Modelling Practice and Theory*, 15(7):801–816, 2007. 10

[118] A. Chatterjee, R. Chatterjee, F. Matsuno, and T. Endo. Augmented stable fuzzy control for flexible robotic arm using LMI approach and neuro-fuzzy state space modeling. *IEEE Transactions on Industrial Electronics*, 55(3):1256–1270, 2008. 10

[119] W. W. Melek and A. A. Goldenberg. The development of a robust fuzzy inference mechanism. *International Journal of Approximate Reasoning*, 39(1):29–47, 2005. 10, 11

[120] R. Sepulveda, O. Castillo, P. Melin, A. Rodriguez-Diaz, and O. Montiel. Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic. *Information Sciences*, 177(10):2023–2048, 2007. 10

[121] H. Hagras. Type-2 fuzzy control: A new generation of fuzzy controllers. *IEEE Computational Intelligence Magazine*, 2(1):30–43, 2007. 10

[122] H. Ying. *Fuzzy Control and Modeling: Analytical Foundations and Applications.* IEEE press, first edition. 11, 39

[123] H.T. Nguyen, V. Kreinovich, and D. Tolbert. A measure of average sensitivity for fuzzy logics. *International Journal of Uncertainty, Fuzinnes, and Knowledge-Based Systems*, 2(4):361–375, 1994. 11

[124] S. Ovchinnikov. On robust fuzzy systems. In *New Frontiers in Fuzzy Logic and Soft Computing Biennial Conference of the North American Fuzzy Information Processing Society NAFIPS*, pages 182–183, New York,NY, June 1996. 11

[125] H.T. Nguyen, V. Kreinovich, and D. Tolbert. On robustness of fuzzy logics. In *Second IEEE International Conference on Fuzzy Systems*, pages 543–547, Reno, NV, March-April 1993. 11

[126] M. Ying. Perturbation of fuzzy reasoning. *IEEE Transactions on Fuzzy Systems*, 7(5):625–629, 1999. 11

[127] K-Y. Cai. Robustness of fuzzy reasoning and δ-equalities of fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 9(5):738–750, 2001. 11

[128] Y. Li, D. Li, W. Pedrycz, and J. Wu. An approach to measure the robustness of fuzzy reasoning. *International Journal of Intelligent Systems*, 20(4):393 – 413, 2005. 11

[129] Y. Li, D. Li, W. Pedrycz, and J. Wu. Approximation and robustness of fuzzy finite automata. *International Journal of Approximate Reasoning*, 47(2):247–257, 2008. 11

[130] Z. Zheng, W. Liu, and K-Y. Cai. Robustness of fuzzy operators in environments with random perturbations (to appear). *Journal of Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 2010. Online available at: http://www.springerlink.com/content/gl2m754583t65j43. 12

[131] J. M. Mendel, H. Hagras, and R. I. John. Standard background material about interval type-2 fuzzy logic systems that can be used by all authors. 19

[132] L. Wang and R. Langari. Complex systems modelling via fuzzy logic. volume 26, pages 100–106, 1996. 29

[133] M. Biglar Begian, W. W. Melek, and J. M. Mendel. Parametric design of stable type-2 TSK fuzzy systems. In *Proceedings of North American Fuzzy Information Processing Society*, pages 1–6, New York, May 2008. 30

[134] S. Keshav. A control-theoretic approach to flow control. *ACM SIGCOMM Computer Communication Review*, 25(1):188–201, 1995. 31

[135] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong. A control theoretic analysis of RED. In *Proceedings of the IEEE Infocom Conference*, pages 1510–1519, Anchorage, Alaska, April 2001. 31

[136] K. Tanaka, T. Hori, and H. O. Wang. A multiple Lyapunov function approach to stabilization of fuzzy control systems. *IEEE Transactions on Fuzzy Systems*, 11(4):582–589, 2003. 35

[137] J-D. Hwang, Z-R. Tsai, and C-J. Chang. A fuzzy Lyapunov function approach to stabilize uncertain nonlinear systems using improved random search method. In *Proceedings of IEEE Conference on Systems, Man, and Cybernetics*, pages 3091–3096, Taipei, Taiwan, October 2006. 35

[138] K. H. Khalil. *Nonlinear Systems*. Englewood Cliffs, NJ, Prentice-Hall, second edition, 1996. 35

[139] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming (web page and software). *http://stanford.edu/ boyd/cvx*, 2009. 37

[140] M. Grant and S. Boyd. *Graph implementations for nonsmooth convex programs.* Recent Advances in Learning and Control (a tribute to M. Vidyasagar), 2008. 37

[141] K.Tanaka and M. Sano. Trajectory stabilization of a model car via fuzzy control. *Fuzzy Sets and Systems*, 70(2-3):155–170, 1995. 39, 40, 42, 49, 51, 53

[142] W. T. Baumann and W. J. Rugh. Feedback control of nonlinear systems by extended linearization. *IEEE Transactions on Automatic Control*, 31(1):40–46, 1986. 47

[143] L. Chua, M. Komuro, and T. Matsumoto. The double scroll family. *IEEE Transactions on Circuits and Systems*, 33(11):1073–1118, 1986. 57, 58

[144] H. O. Wang and K. Tanaka. An LMI-based stable fuzzy control of nonlinear systems and its application to control of chaos. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, pages 1433–1438, New Orleans, September 1996. 57, 58, 59

[145] L. Chua, C. W. Wu, A. Hunang, and G. Q. Zhong. A universal circuit for studying and generating chaos- part I: Routes to chaos. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 40(10):732–744, 1993. 58

[146] M. Biglar Begian, W. W. Melek, and J. M. Mendel. Stability analysis of type-2 fuzzy systems. In *IEEE World Congress on Computational Intelligence*, pages 1305–1310, Hong Kong, June 2008. 63, 65

[147] R. R. Yager and D. P. Filev. *Essentials of Fuzzy Modeling and Control.* New York, Wiley, first edition, 1994. 64, 65

[148] S. Chopra, R. Mitra, and V. Kumar. Fuzzy controller: choosing an appropriate and smallest rule set. *International Journal of Computational Cognition*, 3(4):73–79, 2005. 65

[149] P. J. MacVicar-Whelan. Fuzzy sets for man-machine interactions. *International Journal of Man-Machine Studies*, 8:687–697, 1977. 65

[150] S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice-Hall, second edition, 1999. 77

[151] L-X. Wang. *Adaptive Fuzzy Systems and Control: Design and Stability Analysis.* Prentice-Hall, NJ, 1994. 105