ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

NARMA-L2 CONTROLLER DESIGN FOR NONLINEAR SYSTEMS USING ONLINE LSSVR

M.Sc. THESIS

Gökçen Devlet ŞEN

Department of Control and Automation Engineering

Control and Automation Engineering Programme

JUNE 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

NARMA-L2 CONTROLLER DESIGN FOR NONLINEAR SYSTEMS USING ONLINE LSSVR

M.Sc. THESIS

Gökçen Devlet ŞEN
(504161113)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Gülay ÖKE GÜNEL

JUNE 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DOĞRUSAL OLMAYAN SİSTEMLER İÇİN ÇEVRİMİÇİ EN KÜÇÜK KARELER DESTEK VEKTÖR REGRESYONU İLE NARMA-L2 KONTROLÖR TASARIMI

YÜKSEK LİSANS TEZİ

Gökçen Devlet ŞEN
(504161113)

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Gülay ÖKE GÜNEL

HAZİRAN 2019

Gökçen Devlet ŞEN, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504161113 successfully defended the thesis entitled "NARMA-L2 CONTROLLER DESIGN FOR NONLINEAR SYSTEMS USING ONLINE LSSVR", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Gülay ÖKE GÜNEL** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. İbrahim Eksin** Istanbul Technical University	.....
	Asst. Prof. Dr. Figen Özen Haliç University	.....

Date of Submission : **3 May 2019**
Date of Defense : **12 June 2019**

To my family and friends,

FOREWORD

First of all, I would like to express my sincere appreciation to my thesis advisor Assoc. Prof. Dr. Gülay ÖKE GÜNEL for her guidance and supportive words during this process.

I would like to thank Kemal UÇAK for always answering my questions gladly and being kind and helpful towards me.

Enormous thanks go to my friends Handan NAK and Erhan YUMUK for making my last one year full of art, sports, music and fun. I could have not made it without hearing cheerful laughter of Handan NAK and can not thank her enough for always being with me when I needed, relaxing me, motivating me and amazing me with her knowledge about almost everything. I would also like to thank Erhan YUMUK one more time for always supporting me in the academic field as well as in the sportive field. I could have not had the energy to finish this without our sportive trips.

Another huge thanks go to my friends Damla GÜLEN and Rabia HOZAR. I would like to thank Damla GÜLEN for being with me for more than 10 years, motivating and supporting me in every field. I would also like to thank Rabia HOZAR for being realistic and funny at the same time. Our trip to Switzerland is still the one that keeps me motivated. Thank you both for being such great friends.

I would like to thank all of my friends that I could have not mentioned here for making this experience enjoyable and memorable one.

Last but not least, special thanks go to my lovely family for their endless support, encouragement and love.

June 2019 Gökçen Devlet ŞEN

TABLE OF CONTENTS

Page

ABBREVIATIONS

ANFIS	**:** Adaptive Neuro Fuzzy Inference System
DOF	**:** Degree Of Freedom
LSSVR	**:** Least Square Support Vector Regression
NARMA	**:** Nonlinear Autoregressive Moving Average
NARX	**:** Nonlinear Autoregressive with Exogenous Inputs
PD	**:** Propotional Derivative
PID	**:** Propotional Integral Derivative
SISO	**:** Single Input Single Output
SVM	**:** Support Vector Machine
SVR	**:** Support Vector Regression
TITO	**:** Two Input Two Output

LIST OF TABLES

Page

LIST OF FIGURES

Page

NARMA-L2 CONTROLLER DESIGN FOR NONLINEAR SYSTEMS USING ONLINE LSSVR

SUMMARY

Mathematical modelling of the nonlinear systems is very challenging due to their complexity. Unknown parameters, changing conditions, environmental disturbances and noise make it almost impossible to model a nonlinear system precisely. Without a precise model, designing a controller is also difficult. In order to deal with this problem, adaptive control methods have been suggested. Adaptive control methods can be model free or model based. For the model based adaptive controllers, estimation of the model is very crucial to achieve a successful result. At this point, the utilization of the intelligent methods has been commonly considered. Three of these intelligent methods are artificial neural networks (ANN), adaptive neural-fuzzy inference systems (ANFIS) and support vector regression (SVR). In the literature, ANN and ANFIS have been popularly used for years. However, they are based on empirical risk minimization. As they learn with backpropagation algorithm and have nonconvex cost function, they have a problem of getting stuck at local extremum. Contrarily, SVR proposed by Vapnik, is based on structural risk minimization. Due to its convex cost function, it assures the global convergence. Support vectors can solve both classification and regression problems. For the regression problems, ε-SVR and LSSVR can be used. While estimating a function, ε-SVR ignores the errors as long as they are smaller than ε. If the error is larger than ε, it is punished. To solve an ε-SVR problem, it is needed to solve a quadratic programming problem. In LSSVR proposed by Suykens, squared error term is used and the constraints become equality constraints. To reach the global solution of LSSVR, it is required to solve a set of linear equations. Therefore, LSSVR is faster in terms of computation speed. LSSVR can also be reconstructed easily in the form of an online algorithm. Hence, it can be very useful for online system identification based methods.

Another important issue in model based controllers is representing systems as simple models. In that way, design of a controller becomes easier. As a known fact, nonlinear autoregressive moving average (NARMA) model of nonlinear systems provide the accurate representation of the input-output data in the neighbourhood of the equilibrium state. However, computing the control signal from the NARMA model is very demanding as the control input depends on the model nonlinearly. To deal with this issue, Narendra and Mukhopadhyay proposed a model called as NARMA-L2 model where the control term is detached from the nonlinearities in the model. This model includes two nonlinear functions. By substituting the reference value instead of system output in the model, the control input can be computed easily as a ratio of two time functions. Besides its several advantages, one of the biggest disadvantage of NARMA-L2 controller is the oscillatory control input. To overcome this problem, it has been suggested to add a linear feedback and filter.

In this thesis, a NARMA-L2 controller has been designed by using online LSSVR for two input two output (TITO) nonlinear systems. This method has been previously utilised by Uçak and Günel for single input single output (SISO) nonlinear systems. Here, the method is extended to TITO nonlinear systems and online LSSVR has been used instead of online ε-SVR. In this method, first of all, NARX models for both of the outputs of a TITO system are obtained by using online LSSVR. These models are achieved via current input-output data. Then, attained models are decomposed into NARMA-L2 submodels. The decomposition process is fulfilled by using adaptive parameters. These adaptive parameters are exploited to constitute a relation between LSSVR parameters of NARX models and NARMA-L2 submodels. Since there are two inputs and two outputs, eight parameters are required to accomplish the task. In every iteration, these eight parameters are optimised via Levenberg Marquard Algorithm. After acquiring NARMA-L2 submodels, it becomes straightforward to compute required control intputs for the next time step. The proposed control technique has been implemented to a three tank system. To get rid of the chattering in control signals, a linear feedback and a filter have been added. For the simulations, an asynchronous staircase reference and a sinusoidal reference have been applied to the system. The results show the success of the controller. Additionally, the robustness of the system has also been approved by adding a Gaussian measurement noise and by implementing uncertainty to a system parameter.

Secondly, in this thesis an online computed torque control has been designed by using the previously explained online LSSVR based NARMA-L2 controller. Previously, an offline computed torque control based on SVR was designed by Abdessemed. In computed torque control, the designed controller comprises two parts. The primary control is the inverse controller and the secondary controller is proportional- derivative (PD) controller. In this thesis, primary controller has been designed by adopting online LSSVR based NARMA-L2 controller. Firstly, the inverse model of the robot manipulator is considered as the NARMA-L2 model. Then, it is obtained via the formerly defined technique. Accordingly, the NARMA-L2 controller which is the primary controller in this case, is directly calculated. The secondary controller is again a PD controller. The proposed controller has been tested on a 2-degree of freedom (2-DOF) robot arm. Distinct sinusoidal reference trajectories are applied to the system. The simulations demonstrate that the system provides a good trajectory tracking performance.

DOĞRUSAL OLMAYAN SİSTEMLER İÇİN ÇEVRİMİÇİ EN KÜÇÜK KARELER DESTEK VEKTÖR REGRESYONU İLE NARMA-L2 KONTROLÖR TASARIMI

ÖZET

Gerçek dünyada sistemler son derece karmaşık ve doğrusal olmayan dinamiklere sahiptir. Bu karmaşık dinamikler sebebiyle doğrusal olmayan sistemlerin matematiksel olarak modellenmesi oldukça güçtür. Matematiksel modeldeki katsayıların kesin olarak saptanamama problemi vardır. Buna ek olarak sistemler çevredeki bozucu ve gürültülerden de etkilenirler. Tüm bu durumlar göz önüne alındığında doğrusal olmayan sistemlerin çevresel bozuculara ve parametre belirsizliklerine karşı dayanıklı olarak kontrol edilmesi zorlu bir süreçtir. Sabit parametreli kontrolör yapılarının bu değişken durumlu sistemlerin kontrolü için kullanılması pek efektif olmayacaktır. Çünkü her değişen durum için parametrelerinin yeniden ayarlanması gerekmektedir. Uyarlamalı kontrol yöntemleri bu gibi durumlarda daha etkili bir sonuç için tercih edilmesi gereken yöntemlerdir. Uyarlamalı kontrol yöntemleri model tabanlı veya modelden bağımsız olabilirler. Model tabanlı uyarlamalı kontrolörlerde modelin bir veya birkaç sonraki durum bilgisinin de kestirilmesi gerekmektedir. Bu kestirimlerin daha güvenilir bir şekilde yapılması için akıllı metotların kullanımı tercih edilebilir. Yapay sinir ağları (YSA), uyarlamalı sinirsel bulanık çıkarım sistemi (USBÇS) ve destek vektör regresyonu (DVR) yaygın olarak bilinen akıllı metotlardandır. YSA ve USBÇS literatürde sıklıkla ve başarıyla kullanılmış metotlardandır. Ancak bu yöntemlerin bir dezavantajı yerel uç noktalarda takılı kalıp, her zaman genel çözüme ulaşamamalarıdır. Bunun nedeni deneysel risk minimizasyonu ilkesiyle çalışmalarıdır. Bu yöntemlerde geri yayılım algoritması ile öğrenme sağlanır ve konveks olmayan amaç fonksiyonu kullanılır. Tüm bunlar genel çözüme ulaşmayı engelleyebilir. Destek vektör makineleri (DVM) Vapnik tarafından ortaya atılmış oldukça popüler makine öğrenmesi yöntemlerindendir. DVM, istatiksel öğrenme teorisi ve yapısal risk minimizasyonu ilkesi üstüne kurulmuştur. Bu sebeple YSA ve USBÇS'nin dezavantajı olan yerel uç noktalara takılma problemini ortadan kaldırmıştır. Konveks yapıda bir amaç ölçütüne sahip olduğundan genel çözüme ulaşmayı garantilemektedir. DVM, sınıflandırma ve regresyon problemlerinde kolayca uygulanabilir. Çekirdek fonksiyonu kullanılması özelliği sayesinde düşük boyutlu bir uzayda çözülmesi imkansız olan problemler daha yüksek boyutlu uzaylara taşınarak çözüm doğrusal olarak bulunabilir. Sistem tanıma problemleri regresyon problemi olarak düşünülebildiğinden, DVR sistem tanımada da kolaylıkla kullanılabilir. En çok bilinen DVR çeşitlerinden biri ε-DVR'dir. Bir fonksiyon kestirme problemi üzerinden düşünürsek ε-DVR'de amaç belli bir ε kadar hatayı aşmayacak şekilde fonksiyonun alacağı değeri tahmin etmeye çalışmaktır. Eğer ε kadar hata aşılırsa bu durum ne kadar aşıldığıyla orantılı olarak cezalandırılır, böylelikle en uygun ve genel çözüm bulunmuş olur. ε-DVR ile elde edilen problem karesel programlama problemidir. Suykens tarafından önerilmiş olan en küçük kareler destek vektör regresyonu (EKDVR), ε-DVR'nin geliştirilmiş bir versiyonu olan diğer bir SVR çeşididir. EKDVR'de hata

terimi en küçük karelerle belirtilmiştir ve kısıt fonksiyonunda eşitsizlik yerine eşitlik kullanılmıştır. Bu yöntemin en önemli özelliği genel çözümü karesel programlama problemi çözümü yerine doğrusal denklemler kümesinin çözümüne çevirmesidir. Böylelikle ε-SVR'nin sahip olduğu hesaplama yükü azalmıştır. Ayrıca EKDVR kolayca çevrimiçi çalışacak şekilde düzenlenip kontrol sistemlerinde uygulanabilir.

Model tabanlı kontrolörlerde diğer bir önemli konu sistemlerin basit modellerle ifade edilmesidir. Bu sayede kontrolör tasarımı kolaylaşacaktır. Türkçe'de doğrusal olmayan otoregresif hareketli ortalamalar modeli olarak ifade edilebilecek olan NARMA modeli sistem tanımada sık sık kullanılan zaman serisi kestirme yöntemlerinden biridir. Bilindiği üzere, belli şartlar altında denge noktasına yakın durumlarda NARMA modeli doğrusal olmayan sistemlerin giriş-çıkış ifadesini kesin olarak sağlamaktadır. Ancak kontrol işareti bu modele doğrusal olmayan bir şekilde bağımlıdır. Bu sebeple kontrol işaretini buradan elde etmek oldukça zordur. Narendra ve Mukhopadhyay bu zorluğu aşmak için NARMA-L2 modelini önermişlerdir. Bu modelde kontrol işareti sistem denklemlerinde doğrusal bir biçimde yer almaktadır. NARMA-L2 modeli iki alt fonksiyondan oluşmaktadır. Kontrol işareti bu modelin yeniden düzenlenmesiyle bulunmaktadır. Modelde sistem çıkışı yerine sistemin takip etmesi istenen referans bilgisi konularak gereken kontrol işareti yaklaşık olarak hesaplanır. Birçok avantajının yanında NARMA-L2 kontrolörün en büyük dezavantajı salınımlı kontrol işaretleri üretmesidir. Ancak bu da doğrusal bir geri beslemeyle ve filtreyle giderilebilmektedir.

Bu yüksek lisans tezinde ilk olarak çevrimiçi EKDVR tabanlı NARMA-L2 kontrolörü tasarlanmıştır. Bu kontrol metodu daha önce Uçak ve Günel tarafından tek giriş tek çıkış sistemler için çevrimiçi ε-DVR kullanılarak tasarlanmıştır. Bu tezde bu yöntem iki giriş iki çıkış sistemlerde uygulanmak üzere geliştirilmiştir ve farklı bir yaklaşım olarak çevrimiçi EKDVR kullanılmıştır. NARMA-L2 kontrolörünün salınımının giderilmesi için de yine filtreye ek olarak doğrusal geri besleme yöntemi kullanılmıştır. Bu kontrolör tasarlanırken ilk olarak iki girişli ve iki çıkışlı bir sistemin iki çıkışı için de NARX modeli oluşturulmuştur. Bu model oluşturulurken çevrimiçi EKDVR kullanılmıştır. O anki mevcut giriş çıkış verisi kullanılarak sistem çıkışları kestirilmiştir. Daha sonra bu NARX modellerindeki sistem dinamiklerinin bilgisi NARMA-L2 alt modellerine ayrıştırılmıştır. Bu ayrıştırma sürecinde NARX modeller ve NARMA-L2 alt modeller arasında bir ilişkinin olduğu varsayılmıştır ve bu ilişki uyarlanabilir parametrelerle ifade edilmiştir. İki girişli iki çıkışlı bir sistem ele alındığı için sekiz tane parametreye ihtiyaç duyulmuştur. Bu uyarlanabilir parametreler, NARX ve NARMA-L2 alt modellerinin içindeki EKDVR parametrelerinin arasında bir ilişki kurar ve bu ilişki sayesinde sistem NARMA-L2 modeli bulunur. Uyarlanabilir parametreler her iterasyonda Levenberg Marquard Algoritmasıyla optimize edilir. Sistem çıkışlarının NARMA-L2 modellerinin eldesinden sonra NARMA-L2 kontrolörünün hesaplanması artık oldukça basittir. Geliştirilen bu kontrol metodu iki giriş iki çıkışlı bir üç tanklı sıvı seviye kontrol sistemine uygulanmıştır. Sisteme referans olarak basamak ve sinüzoidal girişler verilmiştir. Simülasyon sonuçlarında sistemin bu referansları başarılı bir şekilde takip ettiği görülmüştür. Sistemin dayanıklılığını test etmek için sisteme Gauss tabanlı ölçüm gürültüsü eklenmiştir ve sistemin referansı takip etmeye devam ettiği görülmüştür. Ayrıca sistem parametrelerinden biri belirsiz yapılarak sistemin belirsizliğe karşı tepkisi ölçülmüştür.

Model tabanlı kontrolörlerden bir diğeri ise ters dinamik kontrolü veya diğer bir deyişle hesaplamalı tork kontrolüdür. Bu yöntemde de yine model dinamiklerinin kesin olarak bilinmesi önem taşımaktadır. Daha önce de bahsedildiği gibi, akıllı algoritmaların bu problem için kullanılması hemen akla gelmektedir. Literatürde yapay sinir ağları tabanlı hesaplamalı tork kontrolörlerine sıklıkla rastlanmaktadır. Ancak daha önce de bahsedildiği gibi sinir ağlarının yerel uç noktalara takılma problemi vardır. Daha çok robot manipülatörlerine uygulanan bu yöntemde, sistemin genel bir çözüme ulaşamaması büyük sıkıntılara yol açabilir. Bu nedenle burada da yine genelleştirme başarısı göz önüne alınarak DVR'nin kullanılması tercih edilecektir.

İkinci olarak bu tezde çevrimiçi EKDVR tabanlı NARMA-L2 kontrolörü kullanılarak çevrimiçi hesaplamalı tork kontrolörü tasarlanmıştır. Abdessemed'in tasarladığı çevrimdışı DVR tabanlı hesaplamalı tork kontrolü metodu bu tezde çevrimiçi hale getirilmiştir. Bu metotta birincil ve ikincil olmak üzere iki tane kontrolör kullanılmıştır. Birincil kontrolör sistemin ters dinamiklerinden oluşan kontrolördür. Bunu yaparken öncelikle robotun ters dinamikleri NARMA-L2 model ile ifade edilmiştir. Daha sonra ilk anlatılan metotta olduğu gibi birincil kontrol işareti NARMA-L2 kontrolör kullanılarak hesaplanmıştır. İkincil kontrolör olarak oransal–türevsel (PD) kontrolör tasarlanmıştır. Bu kontrolörle birinci kontrolörle giderilemeyen model ve bozucu hatalarının giderilmesi amaçlanmıştır. Tasarlanan bu kontrol yapısı iki serbestlik dereceli doğrudan tahrikli bir robot koluna uygulanmıştır. Sistemin performansı sinüs biçimli referansla test edilmiştir. Sonuçlar sistemin yörünge takibini başarılı bir şekilde yerine getirdiğini göstermiştir.

1. INTRODUCTION

Real physical systems mostly have complex and highly nonlinear dynamics. Along with their complexity, the parameters of their dynamical models usually are not known precisely and also are affected by the environment. By taking all of these difficulties into consideration, designing a robust and adaptive controller that compels the system to behave as desired is a demanding issue. Representing the systems as simple models makes the process of computing an efficient controller straightforward. One of the most popular representations of nonlinear systems is the nonlinear autoregressive moving average (NARMA) model [1]. It is a known fact that NARMA models can provide the precise representation of nonlinear systems at the vicinity of the equilibrium state. However, obtaining control inputs from NARMA models is very challenging as control inputs depend on the model nonlinearly. To overcome this problem, Narendra and Mukhopadhyay [2] introduced one of the efficient model approximations called as NARMA-L2 model. In this model, two time functions generate the nonlinear dynamics of the system. The control input is separated from the nonlinear dynamics and appears linearly in the model. Therefore, the computation of the control term becomes a simple task as it is solely the ratio of two time functions. The NARMA-L2 controller has been utilised in many applications. Majstorovic et al. [3] used the NARMA-L2 control method to control a double tank system in an offline manner and compared it with a classical PID controller and a model reference controller. Four different nonlinear systems have been controlled by NARMA-L2 controller and its comparison with other two popular neural network controllers; the model predictive controller and the model reference controller has been provided in [4]. Uçak and Günel [5] proposed an online adaptive control technique for single-input single output (SISO) systems by utilising an online ε-SVR based NARMA-L2 control technique and tested the performance of this controller on a bioreactor system. Apart from its several advantages, the NARMA-L2 controller has a major disadvantage of producing oscillatory control inputs. In order to solve this problem, Pukrittayakamee,

De Jesús and Hagan [6] examined the effects of adding linear feedback to the NARMA-L2 controller and managed to reduce chattering in control action. Mokri and Shafie [7] exploited this chattering-free NARMA-L2 controller to control a manipulator as a real time implementation.

Model based control techniques require great accuracy in system models. To ensure this accuracy in modelling, researchers tend to use intelligent methods to obtain the system models. The precise estimation of the system behaviour relies on the soft computing tools that are used for system identification such as artificial neural networks (ANN), adaptive neuro fuzzy inference system (ANFIS) and support vector regression (SVR). In [8], neural network based identification techniques have been investigated and their success have been evaluated for a three degrees of freedom (3-DOF) robot manipulator. ANFIS based modelling and control have been discussed in [9]. In [10–12] SVR has been examined thoroughly from system identification point of view. The training of the network topologies like ANN and ANFIS is based on backpropagation algorithms. Therefore, they suffer from getting stuck at local minima. Vapnik's Support vector machine (SVM) algorithm is motivated by the statistical learning theory [13]. It is also based on structural risk minimization rather than empirical risk minimization, thus it can avoid getting stuck at local minima. Neural network based methods require the choice of the hidden units. In SVR, there are only a few parameters to adjust and the unknown parameters can be estimated by optimising a convex cost function. Due to its various advantages, SVR has been widely used in control applications. A SVR based generalized predictive control (GPC) has been proposed by Iplikci [14]. In this control structure, an unknown plant has been estimated by SVR and then the SVR model has been employed within the GPC scheme. Yao-Nan and Xiao-Fang [15] designed an internal model control (IMC) structure based on SVR approximations where SVR estimates the inverse controller and the uncertainty compensation in the IMC. Uçak and Günel utilised the online SVR as the controller in [16] and to estimate the parameters of the generalized controller in [17]. Suykens [18] introduced the least square support vector regression (LSSVR) as an extension to ε-SVR. This proposed technique attains the global solution by solving a set of linear equations, thus it saves from being concerned with a quadratic programming problem. Hence, LSSVR overcomes the computational burden of SVR by speeding

up the calculations. In [19], Wang and Hu compared LSSVR with SVR and showed that LSSVR is more efficient in terms of time and memory consumption. An online version of LSSVR has been proposed by Zhu and Mao [20] with the combination of the time window concept. Wanfeng, Shengdun and Yajing [21] proposed an online LSSVR identification based adaptive PID controller and compared its performance with a classical PID controller and an ANN based PID controller. Uçak and Öke [22] also proposed an adaptive PID controller by using online LSSVR and implemented it to a CSTR system. The performance of the proposed controller has been examined in regard to the changing values of adaptive Kernel parameters.

In this thesis, firstly a NARMA-L2 controller has been designed for two input two output (TITO) systems by employing an online LSSVR. A method of decomposing the nonlinear autoregressive with exogenous inputs (NARX) model into NARMA-L2 submodels that is employed in [5] has been extended for TITO systems. Moreover, online LSSVR has been utilized instead of SVR to compute the NARMA-L2 model and the linear feedback has been employed to the NARMA-L2 controller for smoothing the control signal. The proposed technique has been applied to a TITO three tank system and the simulations have been presented. The robustness of the method has been investigated by imposing Gaussian noise to the system. The system response for the case of uncertainty in system parameters has been also evaluated.

Another well-known model based control technique is the inverse dynamics control. This method is popularly used for controlling the robot manipulators. It eliminates the nonlinearities in the robot dynamics and forces the robot track the desired trajectory. However, precise knowledge about the dynamics of the manipulator is again a crucial part of designing this controller. The intelligent learning algorithms have been addressed again to deal with this issue. In [23], the identification of a robotic manipulator has been performed by four types of neural network structures and the results have been compared. Shuzhi, Hang and Woon [24] implemented an adaptive ANN controller for robot manipulators. Jin [25] designed a decentralized adaptive fuzzy controller for robot systems. Again owing to its success in generalization, SVR has been commonly used for controlling robot manipulators. In [26], an offline SVR based computed torque controller has been proposed for a robot manipulator. In this method, two controllers have been utilised, one of which is the inverse dynamics

controller and the other is PD controller. The robot dynamics have been estimated offline by SVR.

Secondly in this thesis, an online computed torque control has been proposed by using NARMA-L2 controller based on online LSSVR for 2-link manipulators. Motivated by the offline method in [26], an online estimation and control scheme has been designed. Inverse dynamics of a robot manipulator have been identified online by previously explained technique. Accordingly, the NARMA-L2 controller has been computed and utilised as a primary controller. PD controller is added as a secondary controller. The proposed method has been executed on a 2-DOF robot manipulator. The simulations have been demonstrated to evaluate the performance.

This thesis is organised as follows: Support Vector Machines are explained thoroughly in chapter 2. The proposed NARMA-L2 based control structure and its optimization method is given in chapter 3. The simulation results are illustrated at the end of chapter 3. In chapter 4, online computed torque method is described and the corresponding simulation performances are given at the end of chapter 4. The thesis is concluded in chapter 5.

2. SUPPORT VECTOR MACHINES

Support vector machines (SVM) are one of the commonly used machine learning techniques [13]. Vapnik's SVM algorithm adopts structural risk minimization, thus it eliminates the drawback of having bad generalization property of artificial neural networks (ANN) that is based on empirical risk minimization [27, 28]. Having a convex cost function and always being able to achieve the global solution make SVM favourable amongst other learning algortihms. SVM can easily be employed in classification and regression problems. In this chapter, SVM will be examined in detail. First of all, SVM structure for classification problems will be explained for both separable and nonseparable patterns. Then, the Kernel Trick concept will be introduced. Lastly, SVM structure for regression problems will be discussed by presenting ε-SVR and LSSVR.

2.1 Support Vector Machines for Classification

SVM are convenient for solving classification problems. In these problems, the aim is to decide the optimal hyperplane which separates these classes from each other. If a two-class problem is considered, there can be found several separating hyperplanes (Figure 2.1). However, the optimal hyperplane has to be determined to maximize the margin between the closest data points of each class to that hyperplane.

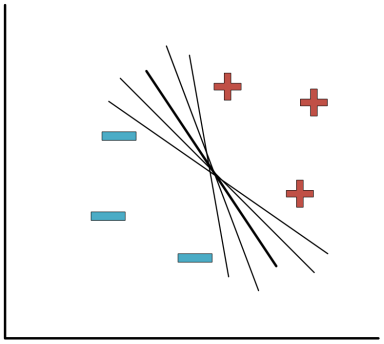

Figure 2.1 : Separating hyperplanes for two classes.

In order to construct a decision rule, it is assumed that the optimal hyperplane is found as given in Figure 2.2. Figure 2.2 illustrates an $\vec{x}$ vector that points to an unknown sample and a $\vec{w}$ vector of any length that is perpendicular to the optimal hyperplane.

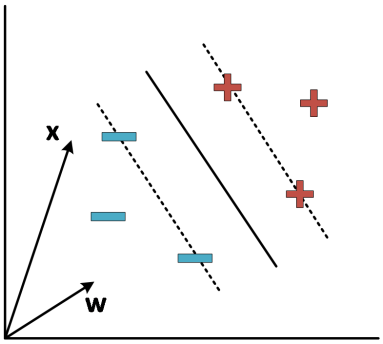

Figure 2.2 : Optimal hyperplane, weight vector and an unknown sample vector.

The decision rule is formulated as follows:

$$\vec{w}^T x + b \geq 0, \quad Then + \tag{2.1}$$

Here, b is a constant. For a $\vec{x}_+$ vector that points to a positive sample and a $\vec{x}_-$ vector that points to a negative sample, the decision rule is reconstructed.

$$\vec{w}^T \vec{x}_+ + b \geq 1 \tag{2.2}$$

$$\vec{w}^T \vec{x}_- + b \geq -1 \tag{2.3}$$

To generalize this formulation, y_i is introduced as follows:

$$y_i \left(\vec{w}^T \vec{x}_i + b\right) - 1 \geq 0, \quad for \ i = 1, 2, \ldots, N \tag{2.4}$$

Here, y_i is 1 for positive samples and -1 for negative samples. The optimal hyperplane and its margin can be described by an analogy of a "street". The samples that are located in the street can be also formulated as given below:

$$y_i(\vec{w}^T \vec{x}_i + b) - 1 = 0 \tag{2.5}$$

The margin of the optimal hyperplane or in other words, width of the street, can be calculated by subtracting a sample in one side of the street from a sample in the other side of the street as in Figure 2.3 and multiplying the solution by a unit vector that is perpendicular to the optimal hyperplane. This calculation is shown as follows:

$$Margin = (\vec{x}_+ - \vec{x}_-) \frac{\vec{w}}{||\vec{w}||} \tag{2.6}$$

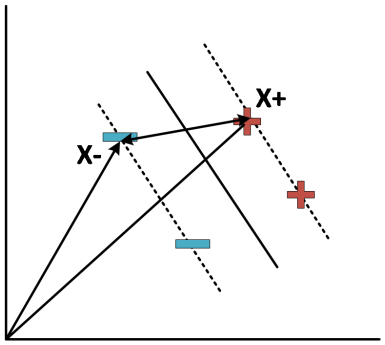

Figure 2.3 : Calculation of the width.

By using equation (2.5), the margin is found as,

$$Margin = \frac{(1-b-(-1-b))}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|} \tag{2.7}$$

Therefore, the maximum width can be found by maximizing $\frac{1}{\|\vec{w}\|}$ or minimizing $\|\vec{w}\|$. Now, the problem turns out to be an optimization problem.

$$\phi(\vec{w}) = \frac{1}{2}\vec{w}^T\vec{w} \tag{2.8}$$

subject to

$$y_i\left(\vec{w}^T\vec{x}_i + b\right) - 1 \geq 0, \quad i = 1,2,\ldots,N \tag{2.9}$$

In order to solve this problem, Lagrangian function is constructed.

$$L(w,b,\alpha) = \frac{1}{2}\vec{w}^T\vec{w} - \sum_{i=1}^{N}\alpha_i[y_i(\vec{w}^T\vec{x}_i + b) - 1] \tag{2.10}$$

Here, α_i are the Lagrange multipliers. The conditions for optimality are formulated as follows:

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{2.11}$$

$$\frac{\partial L}{\partial w} = 0 \rightarrow \vec{w} = \sum_{i=1}^{N}\alpha_i y_i \vec{x}_i \tag{2.12}$$

The solution of the primal problem can also be achieved by solving its dual problem. Transforming the primal problem to its dual makes the optimization problem simpler to solve. The dual problem is obtained by substituting equations (2.11) and (2.12) into (2.10).

$$Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \vec{x}_i^T\vec{x}_j \tag{2.13}$$

subject to

$$\sum_{i=1}^{N}\alpha_i y_i = 0 \tag{2.14}$$

$$\alpha_i \geq 0 \tag{2.15}$$

Minimizing the first Lagrangian or maximizing its dual problem gives the same solution. The optimal weight vector and the bias term are given as:

$$w^* = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i \tag{2.16}$$

$$b^* = -\frac{1}{2} < w^*, x_+ + x_- > \tag{2.17}$$

The corresponding Lagrange multipliers of some data points are nonzero. These points are called support vectors. In equation (2.17), $\vec{x}_+$ and $\vec{x}_-$ are any support vectors. The classifier is now given by equation (2.18).

$$y(x) = sign\left[\sum_{i=1}^{N} \alpha_i y_i \vec{x}_i^T \vec{x} + b\right] \tag{2.18}$$

As it can be seen from the above equation, there is no need to calculate $\vec{w}$. The nonzero Lagrange multipliers α_i and the matching data $\vec{x}_i$ are the support values and the support vectors, respectively.

2.2 Nonseparable Patterns

In the previous section, the calculations have been done by assuming that there is no sample on the optimal hyperplane or its margin. However, most of the time this is not possible as shown in Figure 2.4 and Figure 2.5. The data might be not linearly separable or separating it linearly makes the margin so small that it results in memorization. In this case, a nonnegative slack variable is added to the decision rule.

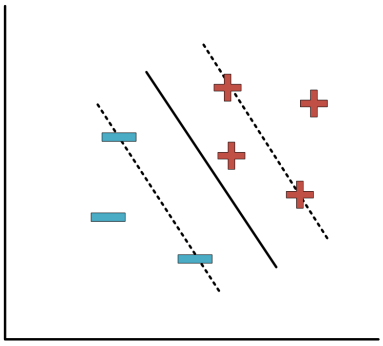

Figure 2.4 : Position of data points for the case $0 \leq \xi_i \leq 1$.

In this case, the optimization problem is given below:

$$\phi(w) = \frac{1}{2}\vec{w}^T\vec{w} + C\sum_{i}^{N} \xi_i \tag{2.19}$$

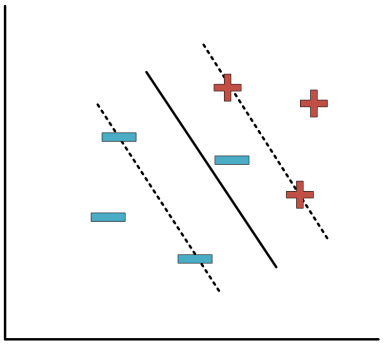

Figure 2.5 : Position of data points for the case $\xi_i > 1$.

subject to

$$y_i\left(\vec{w}^T\vec{x}_i + b\right) \geq 1 - \xi_i, \quad i = 1,2,\ldots,N \tag{2.20}$$

Here, ξ_i are the nonnegative slack variables and C is a positive parameter to be determined. If C is chosen very large, the emphasis on misclassification increases.

For this optimization problem, the Lagrangian is constructed in equation (2.21).

$$L(w,b,\alpha) = \frac{1}{2}\vec{w}^T\vec{w} + C\sum_i^N \xi_i - \sum_{i=1}^N \alpha_i\left[y_i\left(\vec{w}^T\vec{x}_i + b\right) - 1 + \xi_i\right] - \sum_i^N \beta_i\xi_i \tag{2.21}$$

Here, α_i and β_i are Lagrange multipliers. The optimality conditions are presented in equations (2.22), (2.23) and (2.24).

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \tag{2.22}$$

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^N \alpha_i y_i \vec{x}_i \tag{2.23}$$

$$\frac{\partial L}{\partial \xi} = 0 \rightarrow \alpha_i + \beta_i = C \tag{2.24}$$

The dual problem is calculated by substituting equations (2.22) and (2.23) into (2.21).

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j y_i y_j \vec{x}_i^T\vec{x}_j \tag{2.25}$$

subject to

$$\sum_{i=1}^N \alpha_i y_i = 0 \tag{2.26}$$

$$0 \leq \alpha_i \leq C, \quad i = 1,2,\ldots,N \tag{2.27}$$

The classifier is presented in equation (2.28).

$$y(x) = sign\left[\sum_{i=1}^N \alpha_i y_i \vec{x}_i^T\vec{x} + b\right] \tag{2.28}$$

2.3 Kernel Trick

In some cases, there is no way to find a linear classifier even if the misclassification errors are allowed or it is inappropriate to try to find one. However, there is always a linear solution if all of the data points map into a higher dimension. This method is named as 'Kernel Trick'. The Kernel function is presented in equation (2.29).

$$K\left(x_i, x_j\right) = \varphi\left(x_i\right)^T \varphi\left(x_j\right) \tag{2.29}$$

If this Kernel function is substituted into equation (2.28), the classifier becomes,

$$y(x) = sign\left[\sum_{i=1}^{N} \alpha_i y_i K\left(x, x_i\right) + b\right] \tag{2.30}$$

There are many functions to be used as a Kernel function, nonetheless, the commonly used ones are given below: [18, 27].

- Linear Function

$$K\left(x, x_i\right) = x_i^T x \tag{2.31}$$

- Polynomial Function of degree d

$$K\left(x, x_i\right) = \left(x_i^T x + 1\right)^d \tag{2.32}$$

- Gaussian Radial Basis Function

$$K\left(x, x_i\right) = exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \tag{2.33}$$

- Exponential Radial Basis Function

$$K\left(x, x_i\right) = exp\left(-\frac{\|x - x_i\|}{2\sigma^2}\right) \tag{2.34}$$

- Multilayer Perceptron

$$K\left(x, x_i\right) = tanh\left(\kappa x_i^T x + \theta\right) \tag{2.35}$$

2.4 ε-Sensitive Support Vector Regression

Regression problems can also be solved by using SVM. In these problems, the aim is to estimate a function that provides a mapping between given inputs and desired outputs. The term Support Vector Regression (SVR) represents SVM for regression. There are some loss functions that determine how good the function model will predict the outputs or in other words how much error the model will be allowed to make. One of them is ε-insensitive loss function. SVR structured on this loss function is named as ε-SVR.

A data set is given as $\{\vec{x}_i, y_i\}_{i=1}^{N}$ where $\vec{x}_i$ are the input vectors, y_i are the outputs, N is the number of data points. The idea of ε-SVR is that it is allowed to make errors but this is not important as long as the error is less than ε. If the error is larger than ε, then it is penalized. Figure 2.6 and 2.7 illustrate these cases.

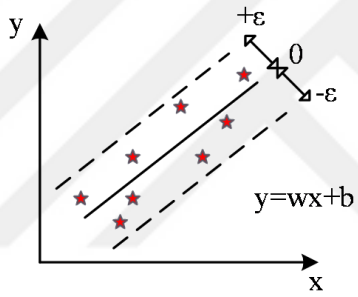

Figure 2.6 : Position of data points for the case where the error is ignored.

The function model is taken as:

$$f(x) = w^T \varphi(x) + b \tag{2.36}$$

ε-insensitive loss function given in Figure 2.8 is defined as

$$|\xi|_\varepsilon = \begin{cases} 0 & if\ |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & otherwise \end{cases} \tag{2.37}$$

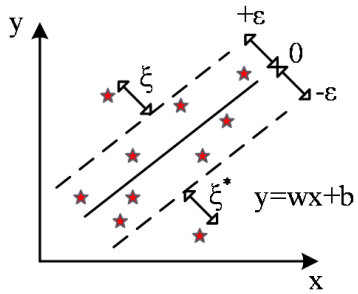

Figure 2.7 : Position of data points for the case where the error is penalised.

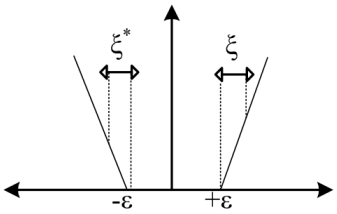

Figure 2.8 : ε-insensitive loss function.

The optimization problem is given below:

$$\phi\left(w,\xi_i,\xi_i^*\right)=\frac{1}{2}\vec{w}^T\vec{w}+C\sum_i^N\left(\xi_i+\xi_i^*\right) \tag{2.38}$$

subject to

$$y_i-\vec{w}^T\vec{x}_i-b\leq\varepsilon+\xi_i \tag{2.39}$$

$$\vec{w}^T\vec{x}_i+b-y_i\leq\varepsilon+\xi_i^* \tag{2.40}$$

$$\xi_i,\xi_i^*\geq 0,\ i=1,2,\ldots,N \tag{2.41}$$

The Lagrangian is constructed as,

$$\begin{aligned}L\left(w,b,\alpha\right)=\frac{1}{2}w^Tw+C\sum_i^N\left(\xi_i+\xi_i^*\right)-\sum_{i=1}^N\alpha_i\left[\varepsilon+\xi_i-y_i+w^T\varphi\left(x_i\right)+b\right]\\-\sum_{i=1}^N\alpha_i^*\left[\varepsilon+\xi_i^*+y_i-w^T\varphi\left(x_i\right)-b\right]-\sum_i^N\left(\eta_i\xi_i+\eta_i^*\xi_i^*\right)\end{aligned} \tag{2.42}$$

Here, α_i, α_i^*, η_i and η_i^* are Lagrange multipliers. The optimality conditions are formulated in equations (2.43), (2.44) and (2.45).

$$\frac{\partial L}{\partial b}=0\rightarrow\sum_{i=1}^N\left(\alpha_i-\alpha_i^*\right)=0 \tag{2.43}$$

$$\frac{\partial L}{\partial w}=0\rightarrow\vec{w}=\sum_{i=1}^N\left(\alpha_i-\alpha_i^*\right)\varphi\left(x_i\right) \tag{2.44}$$

$$\frac{\partial L}{\partial\xi^{(*)}}=0\rightarrow\alpha_i^{(*)}+\eta_i^{(*)}=C \tag{2.45}$$

The dual of the Lagrangian is calculated as in equation (2.46).

$$Q\left(\alpha\right)=\sum_{i=1}^N y_i\left(\alpha_i-\alpha_i^*\right)-\varepsilon\sum_{i=1}^N\left(\alpha_i+\alpha_i^*\right)-\frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N\left(\alpha_i-\alpha_i^*\right)\left(\alpha_j-\alpha_j^*\right)K\left(x_i,x_j\right) \tag{2.46}$$

subject to

$$\sum_{i=1}^N\left(\alpha_i-\alpha_i^*\right)=0 \tag{2.47}$$

$$0 \leq \alpha_i^{(*)} \leq C \tag{2.48}$$

With these calculations, the final function model becomes,

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(x, x_i) + b \tag{2.49}$$

As it can be seen again, there is no need to calculate $\vec{w}$ and $\varphi(\vec{x}_i)$ vectors.

2.5 Least Square Support Vector Regression

LSSVR has been proposed by Suykens [18] with two important modifications to the standard SVR. One of the modifications is that in this algorithm inequality constraints turn into equality constraints. Secondly, the squared error term appears in the primal objective function. The problem is considerably simplified by these two modifications, as it becomes a set of linear equations problem.

The optimization equation is constructed in equation (2.50).

$$\phi(w, e) = \frac{1}{2}\vec{w}^T\vec{w} + \frac{1}{2}C\sum_{i}^{N} e_i^2 \tag{2.50}$$

subject to

$$y_i = w^T\varphi(\vec{x}_i) + b + e_i, \quad i = 1, 2, \ldots, N \tag{2.51}$$

The corresponding Lagrangian function is calculated as follows:

$$L(w, b, e, \alpha) = \frac{1}{2}\vec{w}^T\vec{w} + \frac{1}{2}C\sum_{i}^{N} e_i^2 - \sum_{i=1}^{N} \alpha_i\{\vec{w}^T\varphi(\vec{x}_i) + b + e_i - y_i\} \tag{2.52}$$

Accordingly, the Karush-Kuhn-Tucker conditions are organised as below:

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0 \tag{2.53}$$

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i\varphi(\vec{x}_i) \tag{2.54}$$

$$\frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = Ce_i \tag{2.55}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow y_i = \vec{w}^T\varphi(\vec{x}_i) + b + e_i \tag{2.56}$$

The following linear equations are obtained as the solution.

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & \Omega + C^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \vec{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \vec{y} \end{bmatrix} \tag{2.57}$$

where , $\vec{y} = [y_1, y_2, \ldots, y_N]^T$, $\vec{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$, $\Omega_{i,j} = K(\vec{x}_i, \vec{x}_j)$ and $\vec{1} = [1, 1, \ldots, 1]^T$, $i, j = 1, 2, \ldots, N$. Finally, the estimated function model is given in equation (2.58).

$$f(x) = \sum_{i}^{N} \alpha_i K(\vec{x}, \vec{x}_i) + b \tag{2.58}$$

Another difference of LSSVR from clasical ε-SVR is that in LSSVR all of the Lagrange multipliers are non-zero. Therefore, the training data set includes support vectors.

Here, the available model has been an offline LSSVR model. In the next chapter, after giving a review of NARMA-L2 model and controller concept, online version of LSSVR will be described and will be used for online system identification. Then, the collocation of the both concepts will be proposed.

3. NARMA-L2 CONTROLLER DESIGN BY USING ONLINE LSSVR

In this section, the main control structure proposed in this thesis will be explained. This control method is based on the NARMA-L2 controller and is previously designed for SISO nonlinear systems in [5]. Here, it will be extended to TITO nonlinear systems. To demonstrate the method more apprehensibly, first of all NARMA-L2 model and controller approach will be shown for TITO systems. Then, the online system identification method based on online LSSVR will be explained for TITO systems. The principle method of decomposing a NARX model into a NARMA-L2 model in [5] will be revealed. After that another significant point which is optimization procedure of the adaptive parameters will be described. Finally the simulation results will be given in section 3.6.

3.1 NARMA-L2 Model and Controller for a TITO system

The nonlinear systems can be expressed by using NARMA models. The NARMA model can exactly represent the input-output mapping of nonlinear systems at the neighbourhood of the equilibrium state. The mathematical representation of the NARMA model is given in equation (3.1) [1].

$$y_{n+1} = F\left[u_n, \ldots, u_{n-n_u+1}, y_n, \ldots, y_{n-n_y+1}\right] \tag{3.1}$$

where n_u and n_y are the number of the past inputs and the outputs, respectively. F is a nonlinear mapping function. Even though this model gives the exact representation of a system, the nonlinear dependency of the output on the control input makes it almost impossible to obtain the control term separately. In order to simplify this problem, Narendra and Mukhopadhyay proposed the NARMA-L2 model by approximating the NARMA model with two nonlinear functions [2]. In this approximation, the control input appears linearly in the formulation, thus it can be computed easily. The general form of NARMA-L2 models is given in equation (3.2).

$$\hat{y}_{n+d} = \hat{f}_n + \hat{g}_n u_n \tag{3.2}$$

Here, $\hat{f}_n = F_n(\vec{x}_n)$ and $\hat{g}_n = G_n(\vec{x}_n)$ are the nonlinear dynamics of the system where $\vec{x}_n = \left[u_{n-1},\ldots,u_{n-n_u},y_n,\ldots,y_{n-n_y+1}\right]^T$ is the current state vector and d is the relative degree. The NARMA-L2 model can be extended for TITO systems and can be represented in matrix notation as given below:

$$\begin{bmatrix} y_{1_{n+1}} \\ y_{2_{n+1}} \end{bmatrix} = \begin{bmatrix} \hat{f}_{1_n} \\ \hat{f}_{2_n} \end{bmatrix} + \begin{bmatrix} \hat{g}_{11_n} & \hat{g}_{12_n} \\ \hat{g}_{21_n} & \hat{g}_{22_n} \end{bmatrix} \begin{bmatrix} u_{1_n} \\ u_{2_n} \end{bmatrix} \tag{3.3}$$

Here, f_i and g_{ij} are the nonlinear functions of the system dynamics. This time the current state vector in equation (3.4) is used as the input vector for these functions.

$$\vec{x}_n = \left[u_{1_n},\ldots,u_{1_{\left(n-n_{u_1}\right)}},u_{2_n},\ldots,u_{2_{\left(n-n_{u_2}\right)}},y_{1_n},\ldots,y_{1_{\left(n-n_{y_2}\right)}},y_{2_n},\ldots,y_{2_{\left(n-n_{y_2}\right)}}\right]^T \tag{3.4}$$

By substituting reference signals into equation (3.3) instead of system outputs, the control inputs at time $n+1$ can be computed approximately as follows:

$$\begin{bmatrix} u_{1_{n+1}} \\ u_{2_{n+1}} \end{bmatrix} \cong \begin{bmatrix} \hat{g}_{11_n} & \hat{g}_{12_n} \\ \hat{g}_{21_n} & \hat{g}_{22_n} \end{bmatrix}^{-1} \left(\begin{bmatrix} r_{1_n} \\ r_{2_n} \end{bmatrix} - \begin{bmatrix} \hat{f}_{1_n} \\ \hat{f}_{2_n} \end{bmatrix} \right) \tag{3.5}$$

The NARMA-L2 Controller has the great advantage of computational simplicity. Moreover, there is no need to implement an additional controller. Nevertheless, the biggest disadvantage of this controller arise from its oscillatory behaviour. This issue will be handled in the following sections. The block diagram illustrating the method proposed in this study is given in Figure 3.1.

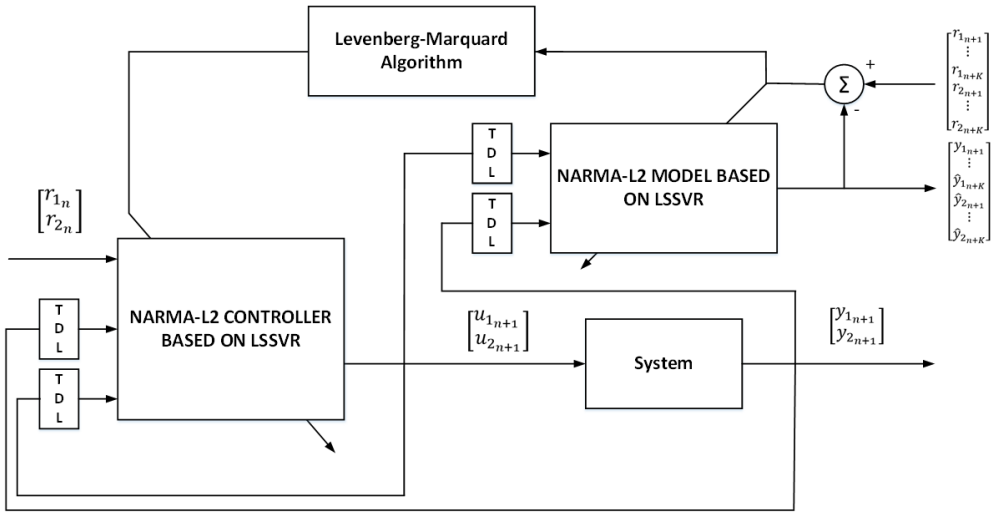

Figure 3.1 : Online LSSVR based NARMA-L2 control structure

In this control method, the NARX model is estimated by online LSSVR. Then, this model is decomposed into the NARMA-L2 model by assuming a relation between the

parameters of the NARX and the NARMA-L2 models. This relation is carried out by adaptive μ_{ij} parameters. In every iteration these adaptive parameters are updated via Levenberg-Marquard Algorithm. The NARMA-L2 controller is calculated by rearranging the NARMA-L2 model. In the next section, the system identification by using online LSSVR will be discussed for a TITO system.

3.2 System Identification of a TITO System via Online LSSVR

The NARX model is a commonly used time series model due to its simplicity [29]. In the case of TITO systems, the NARX model is given as,

$$y_{i_{n+1}} = f(\vec{x}_n), \quad i = 1,2 \tag{3.6}$$

Here, $\vec{x}_n$ is the current state vector shown as,

$$\vec{x}_n = \left[u_{1_n}, \ldots, u_{1_{\left(n-n_{u_1}\right)}}, u_{2_n}, \ldots, u_{2_{\left(n-n_{u_2}\right)}}, y_{1_n}, \ldots, y_{1_{\left(n-n_{y_2}\right)}}, y_{2_n}, \ldots, y_{2_{\left(n-n_{y_2}\right)}} \right]^T \tag{3.7}$$

where n_{u_i} and n_{y_i} are the number of past inputs and outputs.

In the previous chapter, the output of a function estimated by LSSVR has been given in equation (2.58). Now, if the current state vector $\vec{x}_n$ is selected as an input vector, the output of NARX model of a TITO system can be obtained as follows:

$$\vec{y}_{i_{n+1}} = \sum_{m=n-L}^{n-1} \alpha_{i_m} K(\vec{x}_n, \vec{x}_m) + b_i, \quad i = 1,2 \tag{3.8}$$

Here, L is the length of the sliding window that has to be determined by the user. It makes the training data set dynamic, thus, in each iteration the new data points are added to the training data set and the old data points are extricated. The dynamic training data set consists of the past state vectors defined as $X_n = [\vec{x}_{n-1}, \vec{x}_{n-2}, \ldots, \vec{x}_{n-L}]$. Similarly, the output vectors are given as $\vec{Y}_{i_n} = \left[y_{i_n}, y_{i_{n-1}}, \ldots, y_{i_{n-L+1}} \right]^T$ for $i = 1,2$. The dynamic training data sets for a TITO system are $\{X_n, \vec{Y}_{1_n}\}$ and $\{X_n, \vec{Y}_{2_n}\}$.

For the online computation of LSSVR, the Lagrange multipliers and the bias term given in equation (3.8) have to be calculated in every iteration. The solution of LSSVR that is a set of linear equations has been demonstrated in the matrix form in equation (2.57) in chapter 2. Now, it is written in the following form [20–22]:

$$\vec{1}^T \vec{\alpha}_{i_n} = 0 \tag{3.9}$$

$$\vec{Y}_{i_n} = U_n^{-1}\vec{\alpha}_{i_n} + \vec{1}b_{i_n} \tag{3.10}$$
$$i = 1,2$$

where

$$U_n = \left[\Omega_n + C^{-1}I\right]^{-1} \tag{3.11}$$

The equations (3.9) and (3.10) can be solved as follows.

$$U_n\vec{Y}_{i_n} = U_nU_n^{-1}\vec{\alpha}_{i_n} + U_n\vec{1}b_{i_n} \tag{3.12}$$

$$\vec{\alpha}_{i_n} = U_n\vec{Y}_{i_n} - U_n\vec{1}b_{i_n} \tag{3.13}$$

Equation (3.13) is substituted into equation (3.9), the following is obtained:

$$\vec{1}^T\vec{\alpha}_{i_n} = \vec{1}^TU_n\left[\vec{Y}_{i_n} - \vec{1}b_{i_n}\right] = 0 \tag{3.14}$$

The bias description can be calculated by rearranging the equation (3.14) as below:

$$b_{i_n} = \frac{\vec{1}^TU_n\vec{Y}_{i_n}}{\vec{1}^TU_n\vec{1}} \tag{3.15}$$

By substituting the bias into the equation (3.13), the Lagrange multipliers can also be calculated.

$$\vec{\alpha}_{i_n} = U_n\left[\vec{Y}_{i_n} - \vec{1}b_{i_n}\right] \tag{3.16}$$

These Lagrange multipliers and the bias term have been calculated at time index n by using the current training data sets $X_n = [\vec{x}_{n-1}, \vec{x}_{n-2}, \ldots, \vec{x}_{n-L}]$ and $\vec{Y}_{i_n} = \left[y_{i_n}, y_{i_{n-1}}, \ldots, y_{i_{n-L+1}}\right]^T$, for $i = 1,2$. In every iteration, they have to be calculated again. At time index $n+1$, the training data set becomes $X_{n+1} = [\vec{x}_n, \vec{x}_{n-1}, \ldots, \vec{x}_{n+1-L}]$ and the output sets are $\vec{Y}_{i_{n+1}} = \left[y_{i_{n+1}}, y_{i_n}, \ldots, y_{i_{n-L+2}}\right]^T$, for $i = 1,2$. Obviously, the most previous state vector and the outputs are removed, the new state vector and the outputs are added. The dimension of the training data sets do not change.

In the next section, the derivation of the NARMA-L2 model will be explained by using online LSSVR.

3.3 Decomposition of The NARX Model Into NARMA-L2 Model

The NARX models based on online LSSVR has been given for TITO systems in section 3.2. The NARMA-L2 model of a system can also be approximated by using online LSSVR. The dynamic functions of the NARMA-L2 model, $f(.)$ and $g(.)$ are expressed as separate LSSVR models. The decomposition of the NARX model into the NARMA-L2 model is carried out by equalizing the Lagrange multipliers and the bias terms of both models. In [5], this process has been proposed for single-input single-output (SISO) systems. Here, it will be extended to TITO systems.

The NARX model of the outputs that are identified by LSSVR at time step n are:

$$\hat{y}_{i_{NARX_n}} = \sum_{m=1}^{N} \alpha_{i_m} K(\vec{x}_m, \vec{x}_n) + b_i, \quad i = 1,2 \tag{3.17}$$

The nonlinear functions of NARMA-L2 models based on LSSVR are given by:

$$\hat{f}_{i_n} = F_{i_n}(\vec{x}_n) = \sum_{m=1}^{N} \beta_{i_m} K(\vec{x}_m, \vec{x}_n) + b_{f_i}, \quad i = 1,2 \tag{3.18}$$

$$\hat{g}_{ij_n} = G_{ij_n}(\vec{x}_n) = \sum_{m=1}^{N} \theta_{ij_m} K(\vec{x}_m, \vec{x}_n) + b_{g_{ij}}, \quad i,j = 1,2 \tag{3.19}$$

The NARX models are decomposed into the NARMA-L2 models as follows:

$$y_{i_{NARX}} = y_{i_{NARMA}}, \quad i = 1,2 \tag{3.20}$$

$$\sum_{m=1}^{N} \alpha_{i_m} K(\vec{x}_m, \vec{x}_n) + b_i = \sum_{m=1}^{N} [\beta_{i_m} + \theta_{i1_m} u_{1_n} + \theta_{i2_m} u_{2_n}] K(\vec{x}_m, \vec{x}_n) + b_{f_i} + b_{g_{i1}} + b_{g_{i2}} \tag{3.21}$$

Since the Kernel functions are already the same, equalizing Lagrange multipliers and bias terms are enough to make both sides of the equation the same.

$$\alpha_{i_m} = \beta_{i_m} + \theta_{i1_m} u_{1_n} + \theta_{i2_m} u_{2_n} \tag{3.22}$$

$$b_i = b_{f_i} + b_{g_{i1}} u_{1_n} + b_{g_{i2}} u_{2_n} \tag{3.23}$$

$$for \quad i = 1,2$$

The following relations between the Lagrange multipliers and the terms of the NARMA-L2 submodels are assumed:

$$\theta_{12_m} = \mu_{11}(.)\,\theta_{11_m},\ \beta_{1_m} = \mu_{12}(.)\,\theta_{12_m} \tag{3.24}$$

$$b_{g_{12}} = \mu_{13}(.)\, b_{g_{11}},\ b_{f_1} = \mu_{14}(.)\, b_{g_{12}} \tag{3.25}$$

$$\theta_{21_m} = \mu_{21}(.)\, \theta_{22_m},\ \beta_{2_m} = \mu_{22}(.)\, \theta_{21_m} \tag{3.26}$$

$$b_{g_{21}} = \mu_{23}(.)\, b_{g_{22}},\ b_{f_2} = \mu_{24}(.)\, b_{g_{21}} \tag{3.27}$$

The decomposition of the NARX model into the NARMA-L2 model can be performed easily with this relation.

$$\begin{aligned} \theta_{11_m} &= \frac{\alpha_{1_m}}{\mu_{11}(.)\,\mu_{12}(.) + \mu_{11}(.)\,u_{2_n} + u_{1_n}} \\ \theta_{12_m} &= \mu_{11}(.)\,\theta_{11_m} \\ \beta_{1_m} &= \mu_{11}(.)\,\mu_{12}(.)\,\theta_{11_m} \end{aligned} \tag{3.28}$$

$$\begin{aligned} b_{g_{11}} &= \frac{b_1}{\mu_{13}(.)\,\mu_{14}(.) + \mu_{13}(.)\,u_{2_n} + u_{1_n}} \\ b_{g_{12}} &= \mu_{13}(.)\,b_{g_{11}} \\ b_{f_1} &= \mu_{13}(.)\,\mu_{14}(.)\,b_{g_{11}} \end{aligned} \tag{3.29}$$

$$\begin{aligned} \theta_{22_m} &= \frac{\alpha_{2_m}}{\mu_{21}(.)\,\mu_{22}(.) + \mu_{21}(.)\,u_{1_n} + u_{2_n}} \\ \theta_{21_m} &= \mu_{21}(.)\,\theta_{22_m} \\ \beta_{2_m} &= \mu_{21}(.)\,\mu_{22}(.)\,\theta_{22_m} \end{aligned} \tag{3.30}$$

$$\begin{aligned} b_{g_{22}} &= \frac{b_2}{\mu_{21}(.)\,\mu_{22}(.) + \mu_{21}(.)\,u_{1_n} + u_{2_n}} \\ b_{g_{21}} &= \mu_{23}(.)\,b_{g_{22}} \\ b_{f_2} &= \mu_{23}(.)\,\mu_{24}(.)\,b_{g_{22}} \end{aligned} \tag{3.31}$$

3.4 Determination of $\mu_{ij}(.)$ Parameters for Optimizing Tracking Performance

The $\mu_{ij}(.)$ parameters that build a relationship between the Lagrange multipliers and the bias term have to be optimized for a good tracking performance. The objective function that ensures the minimization of the tracking error by predicting the system behavior has been employed for SISO systems in [5, 30]. Here, it will be augmented for TITO systems.

$$E(\mu) = \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{2}\hat{e}_{j_{n+k}} + \frac{1}{2}\lambda\sum_{j=1}^{2}\left[u_{j_{n+1}} - u_{j_n}\right]^2 \tag{3.32}$$

where $\hat{e}_{j_{n+k}} = r_{j_{n+k}} - \hat{y}_{j_{n+k}}$. Here, K determines how many steps will be predicted and λ penalises the control signal.

The adaptive $\mu_{ij}(.)$ parameters are optimized via Levenberg-Marquard Algorithm as follows:

$$[\mu_{new}]_{8x1} = [\mu_{old}]_{8x1} + \left[J^T J + \eta I\right]^{-1} J^T e \tag{3.33}$$

where

$$J = \begin{bmatrix} \frac{\partial \hat{e}_{1_{n+1}}}{\partial \hat{y}_{1_{n+1}}} \frac{\hat{y}_{1_{n+1}}}{\partial \mu_{11}} & \cdots & \frac{\partial \hat{e}_{1_{n+1}}}{\partial \hat{y}_{1_{n+1}}} \frac{\hat{y}_{1_{n+1}}}{\partial \mu_{24}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{e}_{1_{n+k}}}{\partial \hat{y}_{1_{n+k}}} \frac{\hat{y}_{1_{n+k}}}{\partial \mu_{11}} & \cdots & \frac{\partial \hat{e}_{1_{n+k}}}{\partial \hat{y}_{1_{n+k}}} \frac{\hat{y}_{1_{n+k}}}{\partial \mu_{24}} \\ \frac{\partial \hat{e}_{2_{n+1}}}{\partial \hat{y}_{2_{n+1}}} \frac{\hat{y}_{2_{n+1}}}{\partial \mu_{11}} & \cdots & \frac{\partial \hat{e}_{2_{n+1}}}{\partial \hat{y}_{2_{n+1}}} \frac{\hat{y}_{2_{n+1}}}{\partial \mu_{24}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \hat{e}_{2_{n+k}}}{\partial \hat{y}_{2_{n+k}}} \frac{\hat{y}_{2_{n+k}}}{\partial \mu_{11}} & \cdots & \frac{\partial \hat{e}_{2_{n+k}}}{\partial \hat{y}_{2_{n+k}}} \frac{\hat{y}_{2_{n+k}}}{\partial \mu_{24}} \\ \frac{\partial \sqrt{\lambda} \Delta u_{1_{n+1}}}{\partial \mu_{11}} & \cdots & \frac{\partial \sqrt{\lambda} \Delta u_{1_{n+1}}}{\partial \mu_{24}} \\ \frac{\partial \sqrt{\lambda} \Delta u_{2_{n+1}}}{\partial \mu_{11}} & \cdots & \frac{\partial \sqrt{\lambda} \Delta u_{2_{n+1}}}{\partial \mu_{24}} \end{bmatrix} = \tag{3.34}$$

$$= \begin{bmatrix} -\frac{\partial y_{1_{n+1}}}{\partial u_{1_{n+1}}} & -\frac{\partial y_{1_{n+1}}}{\partial u_{2_{n+1}}} \\ \vdots & \vdots \\ -\frac{\partial y_{1_{n+k}}}{\partial u_{1_{n+1}}} & -\frac{\partial y_{1_{n+k}}}{\partial u_{2_{n+1}}} \\ -\frac{\partial y_{2_{n+1}}}{\partial u_{1_{n+1}}} & -\frac{\partial y_{2_{n+1}}}{\partial u_{2_{n+1}}} \\ \vdots & \vdots \\ -\frac{\partial y_{2_{n+k}}}{\partial u_{1_{n+1}}} & -\frac{\partial y_{2_{n+k}}}{\partial u_{2_{n+1}}} \\ \sqrt{\lambda} & \sqrt{\lambda} \\ \sqrt{\lambda} & \sqrt{\lambda} \end{bmatrix}_{(2K+2)x2} \begin{bmatrix} \frac{\partial u_{1_{n+1}}}{\partial \mu_{11}} & \cdots & \frac{\partial u_{1_{n+1}}}{\partial \mu_{24}} \\ \frac{\partial u_{2_{n+1}}}{\partial \mu_{11}} & \cdots & \frac{\partial u_{2_{n+1}}}{\partial \mu_{24}} \end{bmatrix}_{2x8}$$

$$e = \left[\hat{e}_{1_{n+1}}, \ldots, \hat{e}_{1_{n+k}}, \hat{e}_{2_{n+1}}, \ldots, \hat{e}_{2_{n+k}}, \sqrt{\lambda}\Delta u_{1_{n+1}}, \sqrt{\lambda}\Delta u_{2_{n+1}}\right]^T \tag{3.35}$$

η is computed by using Golden Section algorithm.

3.5 Smoothing The Control Signal

As it is mentioned earlier, the disadvantage of the NARMA-L2 Controller is that it has a chattering problem. In this thesis the technique that reduces the oscillatory behaviour proposed in [6] has been utilized. In [6] this technique has been described for SISO systems. Here, it will be enhanced for TITO systems.

The computation of the NARMA-L2 Controller for TITO systems has been given in equation (2.15). Now, the linear feedback term will be added to this equation as

follows.

$$\begin{bmatrix} u_{1_{n+1}} \\ u_{2_{n+1}} \end{bmatrix} = \begin{bmatrix} \hat{g}_{11_n} & \hat{g}_{12_n} \\ \hat{g}_{21_n} & \hat{g}_{22_n} \end{bmatrix}^{-1} \left(\begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix} \begin{bmatrix} r_{1_n} \\ r_{2_n} \end{bmatrix} - \begin{bmatrix} \hat{f}_{1_n} \\ \hat{f}_{2_n} \end{bmatrix} - \begin{bmatrix} \vec{d}_1^T & \vec{0}^T \\ \vec{0}^T & \vec{d}_2^T \end{bmatrix} \vec{Y}_p \right) \tag{3.36}$$

where $d_i = [d_{i1}, \ldots, d_{ip}]^T$ for $i = 1,2$, $\vec{0} = [0, \ldots, 0]^T$ and $\vec{Y}_p$ is a vector of the previous system outputs.

If the nonlinear functions of the NARMA-L2 model are approximated accurately, the equality becomes:

$$\begin{bmatrix} y_{1_{n+1}} \\ y_{2_{n+1}} \end{bmatrix} + \begin{bmatrix} \vec{d}_1^T & \vec{0}^T \\ \vec{0}^T & \vec{d}_2^T \end{bmatrix} \begin{bmatrix} y_{1_{n-1}} \\ \vdots \\ y_{1_{n-p}} \\ y_{2_{n-1}} \\ \vdots \\ y_{2_{n-p}} \end{bmatrix} = \begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix} \begin{bmatrix} r_{1_n} \\ r_{2_n} \end{bmatrix} \tag{3.37}$$

If Z-transform is taken,

$$\begin{bmatrix} D_1(z) & 0 \\ 0 & D_2(z) \end{bmatrix} \begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix} \begin{bmatrix} R_1(z) \\ R_2(z) \end{bmatrix} \tag{3.38}$$

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} \frac{c_1 R_1(z)}{D_1(z)} \\ \frac{c_2 R_2(z)}{D_2(z)} \end{bmatrix} \tag{3.39}$$

$$D_i(z) = 1 + d_{i1} z^{-1} + \ldots + d_{ip} z^{-p} \quad for\ i = 1,2 \tag{3.40}$$

To eliminate the oscillations in the system response, the roots of $D_1(z)$ and $D_2(z)$ can be selected to be inside the unit circle and close to 1. The closer to 1 the roots are chosen, the smoother the system response will be. However, there is a risk of having a worse response if the approximation of the model is not correct and the roots are chosen very close to 1.

3.6 Simulations Results of The NARMA-L2 Controller on a Three Tank System

The proposed controller is tested on a three tank system that is demonstrated in Figure 3.2. The dynamical equations of the system are given as follows:

$$\begin{aligned} y_1(t) &= \frac{1}{S}\left[u_1(t) - Q_{13}(t) - Q_{10}(t)\right] \\ y_2(t) &= \frac{1}{S}\left[u_2(t) + Q_{32}(t) - Q_{20}(t)\right] \\ y_3(t) &= \frac{1}{S}\left[Q_{13}(t) - Q_{32}(t) - Q_{30}(t)\right] \end{aligned} \tag{3.41}$$

where

$$
\begin{aligned}
Q_{13}(t) &= az_{13} S_p sgn(y_1(t) - y_3(t)) \sqrt{2g|y_1(t) - y_3(t)|} \\
Q_{32}(t) &= az_{32} S_p sgn(y_3(t) - y_2(t)) \sqrt{2g|y_3(t) - y_2(t)|} \\
Q_{10}(t) &= az_{10} S_p \sqrt{2g|y_1(t)|} \\
Q_{20}(t) &= az_{20} S_p \sqrt{2g|y_2(t)|} \\
Q_{30}(t) &= az_{30} S_p \sqrt{2g|y_3(t)|}
\end{aligned}
\tag{3.42}
$$

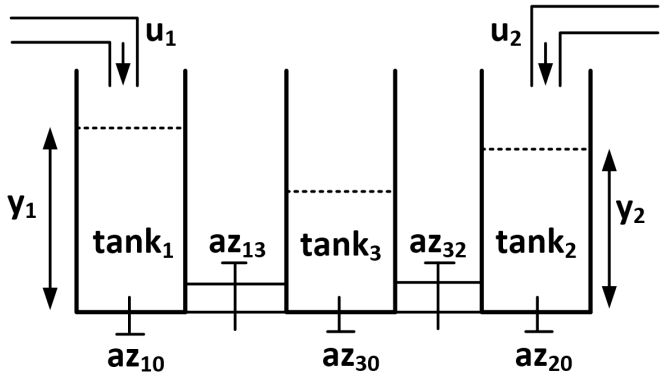

Figure 3.2 : Three tank system

The system parameters are listed in Table 3.1 [31, 32].

Table 3.1 : The system parameters

Parameters	Value
$y_i(t)$: height of the liquid level for tank_i	m
$u_i(t)$: inflow rate of i^{th} pump	m^3/s
az_{13}: the leakage parameter between tank 1 and tank 3	0.22
az_{32}: the leakage parameter between tank 3 and tank 2	0.28
az_{10}: the leakage parameter from tank 1 to reservoir	0.40
az_{20}: the leakage parameter from tank 2 to reservoir	0.27
az_{30}: the leakage parameter from tank 3 to reservoir	0.27
S: cross sectional area	$0.0154m^2$
S_p: section of connection pipe n	$5x10^-5m^2$
g: gravitation coefficient	$9.81m/s^2$

Here, y_1 and y_2 represent the system outputs. u_1 and u_2 are the control signals that are restricted between 0 and 10^{-4}. The aim The Fourth Order Runge Kutta (FORK) integration method is used for the simulations. Sampling time of the simulations has been selected as $T_s = 1s$. As it has been mentioned earlier, the NARX model arises from the nonlinear relationship of the previous inputs and previous outputs. The

number of previous inputs and outputs n_{u_1}, n_{u_2}, n_{y_1} and n_{y_2} given in equation (3.4) are all selected as 3. Since there is a huge difference in the scale of the input and the output values, it is necessary to normalize all the data within a uniform range to prevent overriding. The normalization procedure is carried out as follows.

$$z_{normal} = s_1 + (s_2 - s_1)\left(\frac{z - z_{min}}{z_{max} - z_{min}}\right) \tag{3.43}$$

where s_1 and s_2 are the endpoints of the desired normalization interval. z_{normal} is the normalized value of z in the interval $[s_1, s_2]$. z_{max} and z_{min} are the maximum and the minimum values of the data set that includes z. As it is suggested in [33], the normalization interval is chosen as $[0.1, 0.9]$ in the simulations. The Kernel function of online LSSVR is adopted as Exponential Radial Basis Function. The parameters of the Kernel are fixed at $\sigma_1 = \sigma_2 = 1$. The regularization parameter of online LSSVR is also kept fixed at C=1000 as it is better to select a large value. The sliding window of the online LSSVR has been determined as $L = 25$. The prediction horizon and the penalty parameter of the objective function for the tracking performance are set to $K = 5$ and $\lambda = 10^{-3}$ respectively.

Furthermore, as it is explained in section 3.6, in order to prevent chattering, linear feedback is added to the equation. In simulations, a first order linear feedback term is added and the corresponding coefficient has been selected as $p = 1$. The roots are chosen as $D_1 = D_2 = 0.75$. To eliminate steady state error, c_1 and c_2 are set to 0.25. Additionally, the low pass filter given in equation (3.44) has been applied to both of the control signals.

$$H(z) = \frac{0.09516z^{-1}}{1 - 0.9048z^{-1}} \tag{3.44}$$

The system performance has been tested for two different reference trajectories. Figures 3.3 and 3.4 illustrate the performance for the noiseless case. As it is observed from these figures, the outputs of both Tank 1 and Tank 2 follow the given staircase references without any oscillations. These reference signals change sometimes synchronously and sometimes asynchronously. In this way, the adaptation of the controller to coupling effects between the tanks has been observed. The control signals stay in the restricted interval and produce no oscillations. The changes in the adaptive control parameters $\mu_{ij}(.)$ are also given in Figures 3.5, 3.6, 3.7 and 3.8.

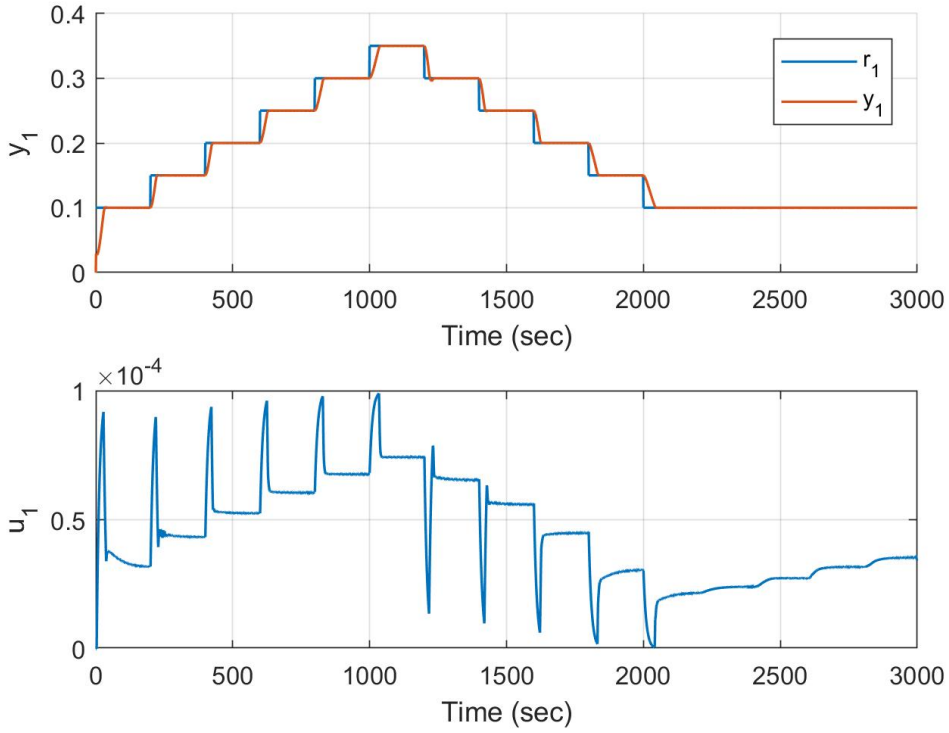

Figure 3.3 : The reference tracking of tank 1 and the inflow rate of pump 1 for the nominal case.

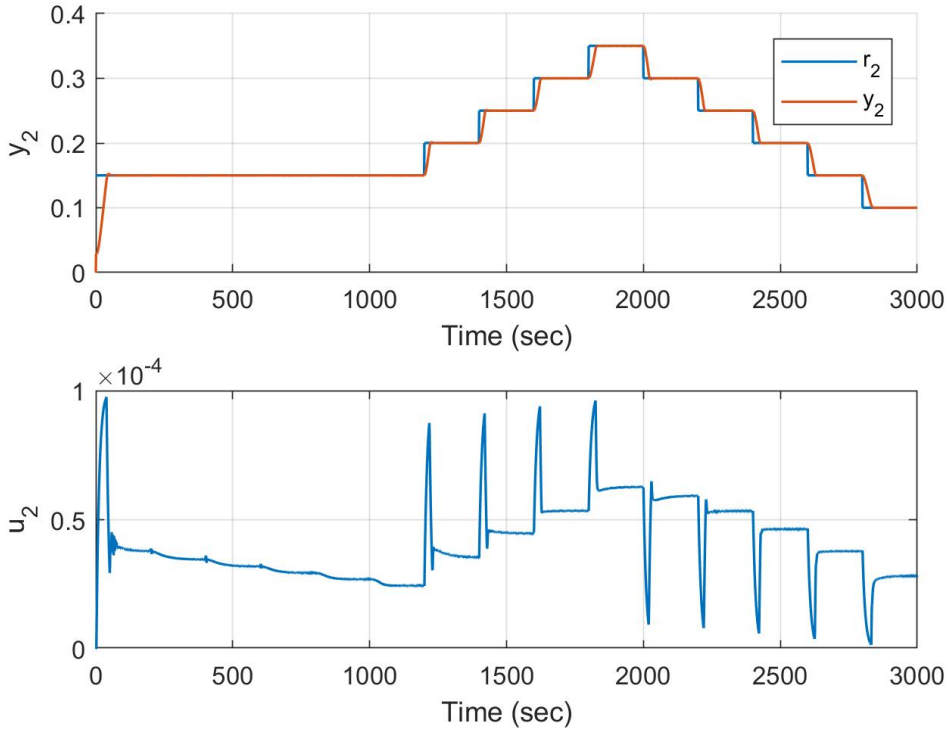

Figure 3.4 : The reference tracking of tank 2 and the inflow rate of pump 2 for the nominal case.

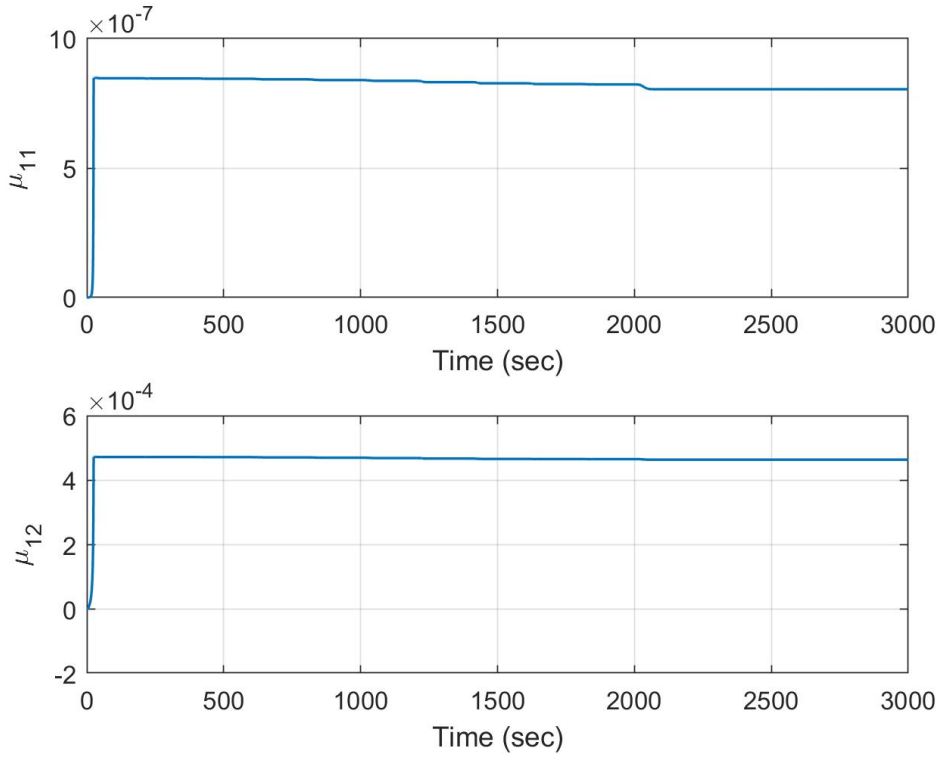

Figure 3.5 : The alteration of adaptive $\mu_{11}(.)$ and $\mu_{12}(.)$ parameters for the nominal case.

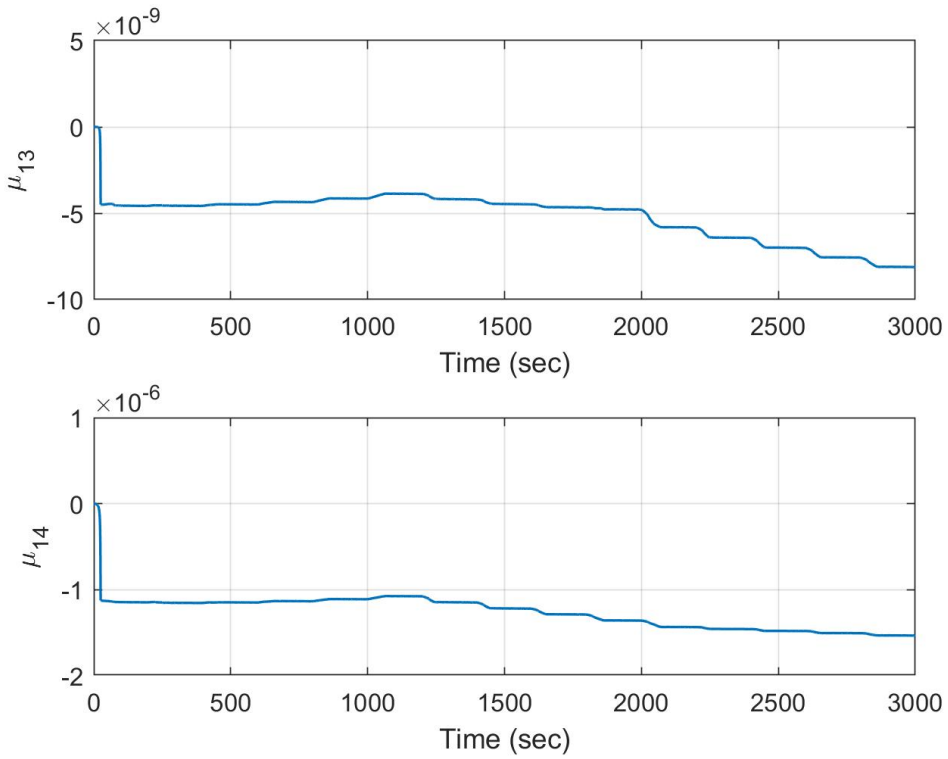

Figure 3.6 : The alteration of adaptive $\mu_{13}(.)$ and $\mu_{14}(.)$ parameters for the nominal case.

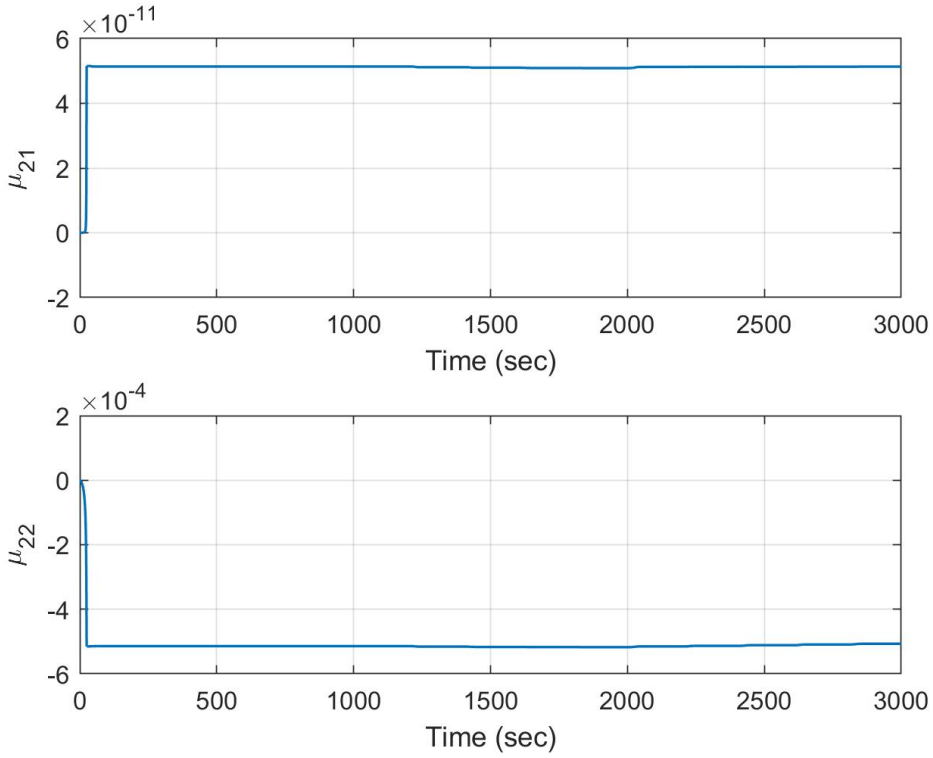

Figure 3.7 : The alteration of adaptive $\mu_{21}(.)$ and $\mu_{22}(.)$ parameters for the nominal case.

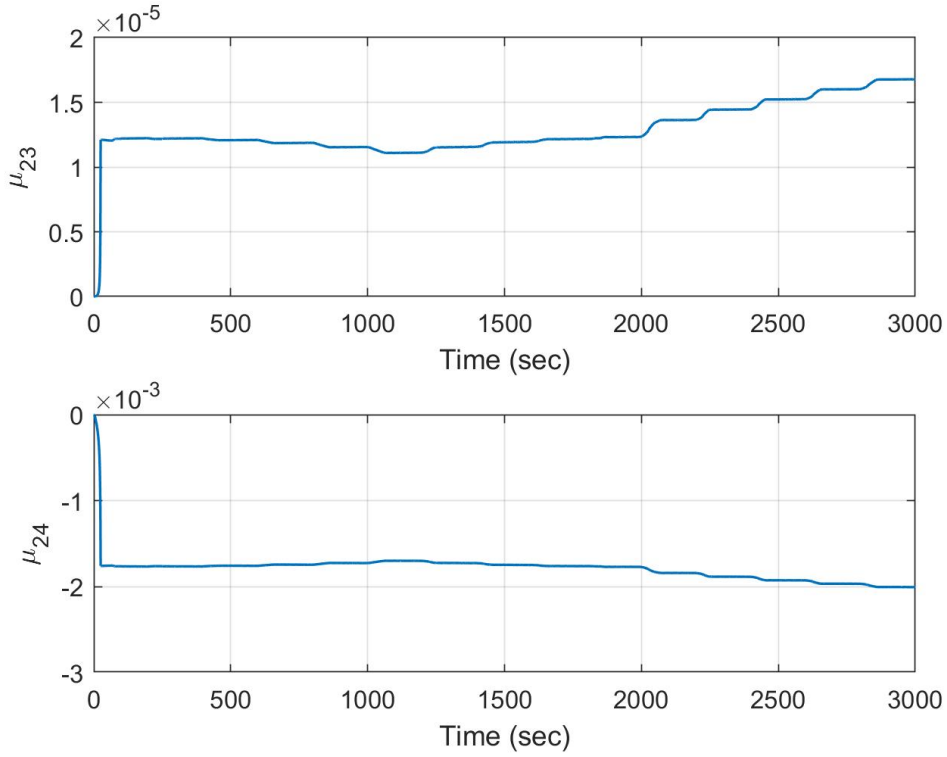

Figure 3.8 : The alteration of adaptive $\mu_{23}(.)$ and $\mu_{24}(.)$ parameters for the nominal case.

The system performance has also been tested for sinusoidal references for the noiseless case. Figure 3.9 and 3.10 show the success of the system responses to the sinusoidal inputs. It can be seen in these figures that while the reference signal for one of the tanks is sinusoidal, the reference signal for the other one is kept constant. Even though the reference signal is constant, the inflow rate of the corresponding pump has to change to achieve the task. The changes in adaptive parameters are depicted in Figures 3.11, 3.12, 3.13 and 3.14.

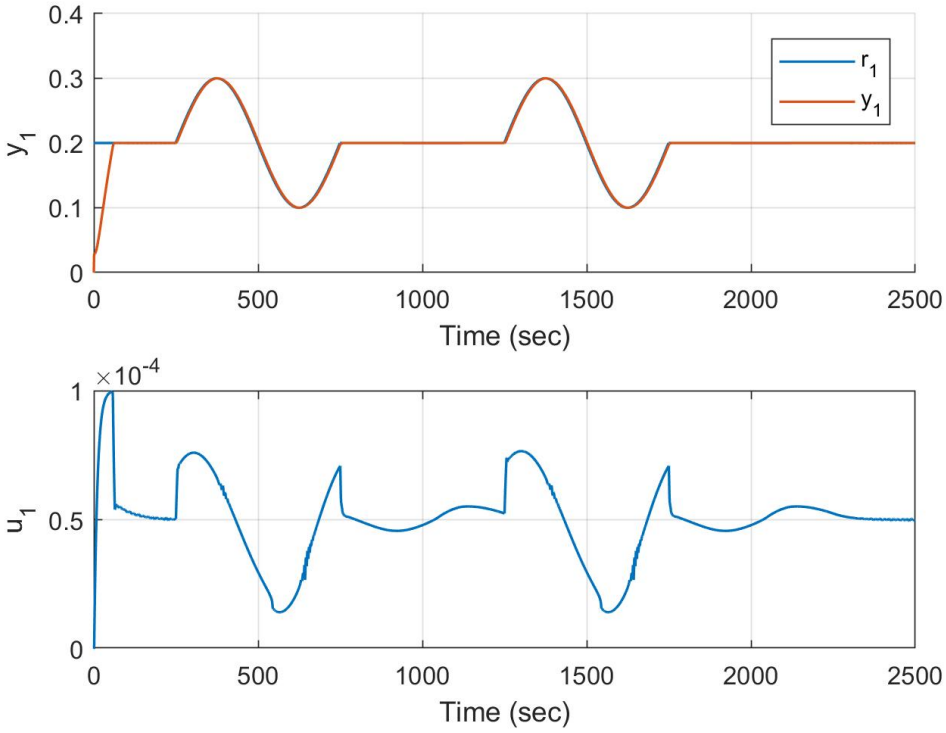

Figure 3.9 : The reference tracking of Tank 1 and the inflow rate of Pump 1 for the sinusoidal input.

In order to test the robustness of the system, a zero mean Gaussian noise with $50dB$ of signal-to-noise ratio (SNR) given in equation (3.45) has been applied to controlled outputs of the system as a measurement error [30].

$$SNR = 10log_{10}\left(\frac{\sigma_y^2}{\sigma_v^2}\right) dB \tag{3.45}$$

where σ_y and σ_v are the variances of the measured output and the noise, respectively. For the case with Gaussian noise added, the system has been forced to track the same staircase reference inputs applied in the nominal case.

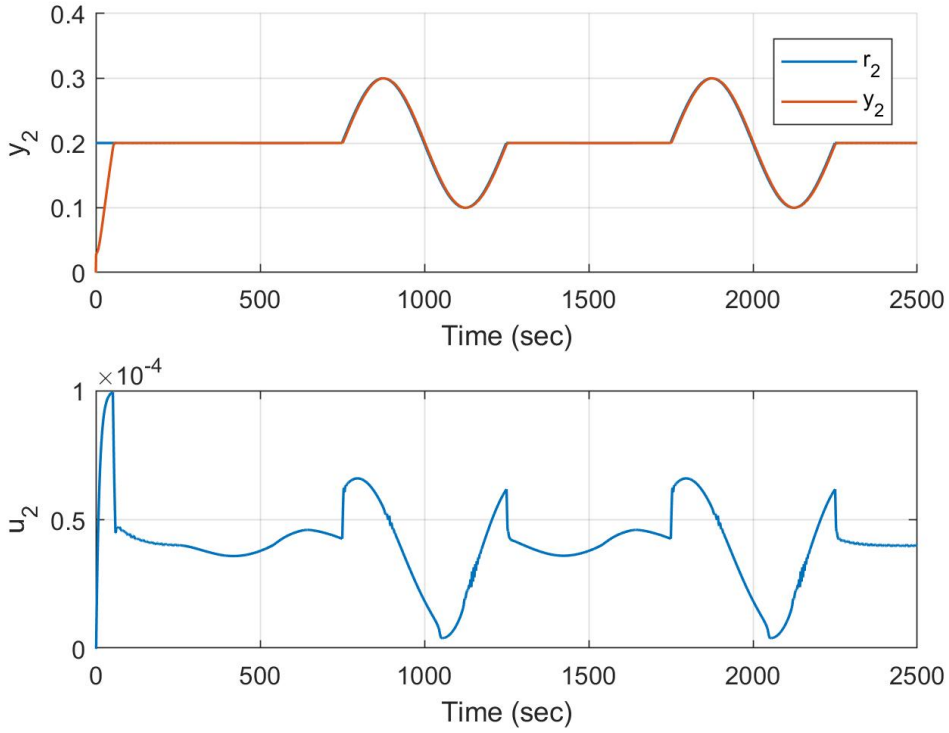

Figure 3.10 : The reference tracking of Tank 2 and the inflow rate of Pump 2 for the sinusoidal input.

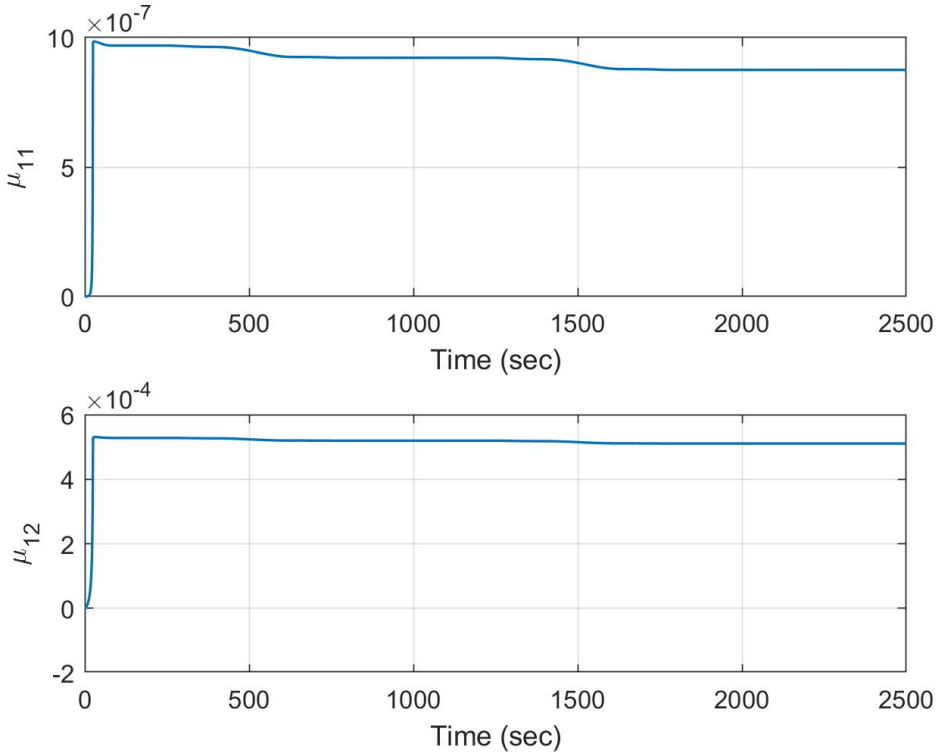

Figure 3.11 : The alteration of adaptive $\mu_{11}(.)$ and $\mu_{12}(.)$ parameters for the sinusoidal input.

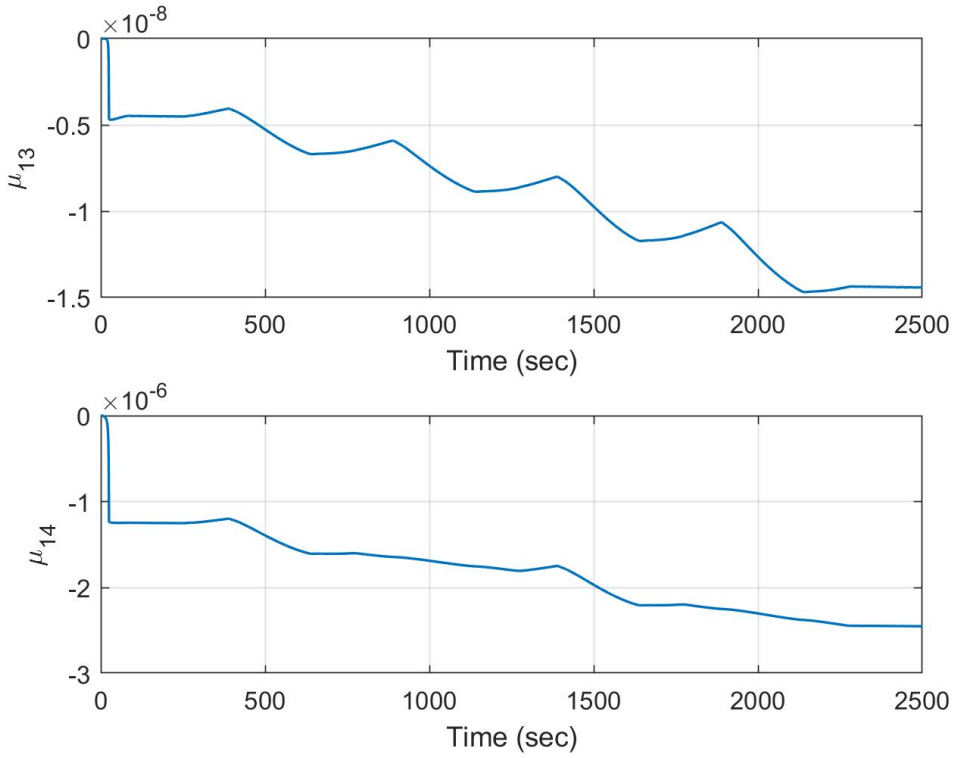

Figure 3.12 : The alteration of adaptive $\mu_{13}(.)$ and $\mu_{14}(.)$ parameters for the sinusoidal input.

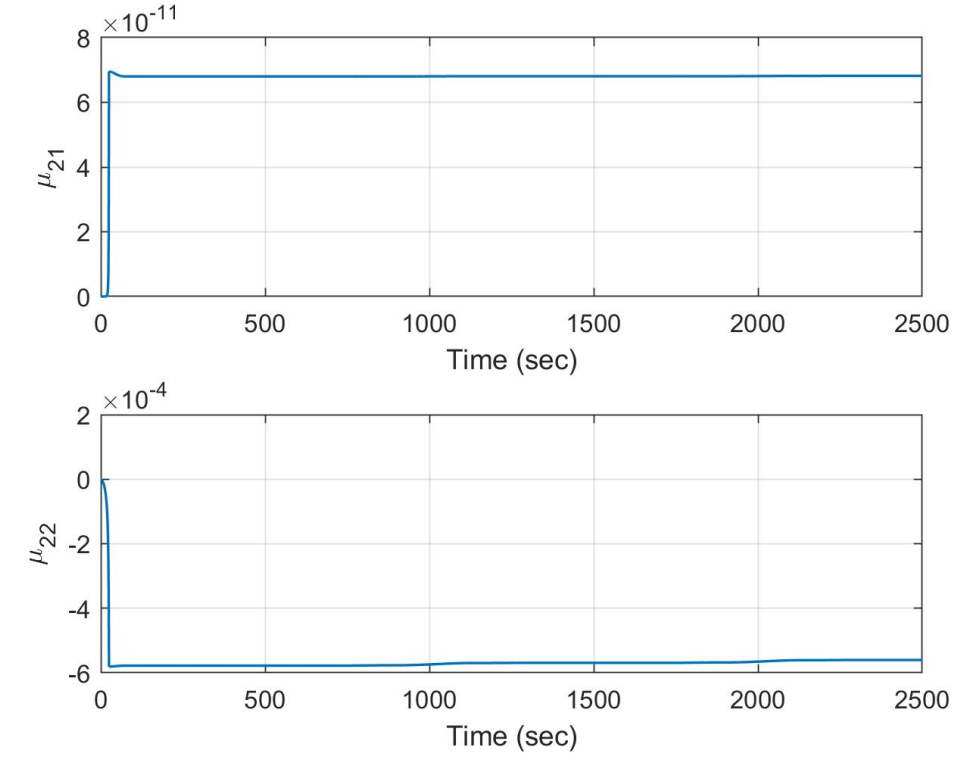

Figure 3.13 : The alteration of adaptive $\mu_{21}(.)$ and $\mu_{22}(.)$ parameters for the sinusoidal input.

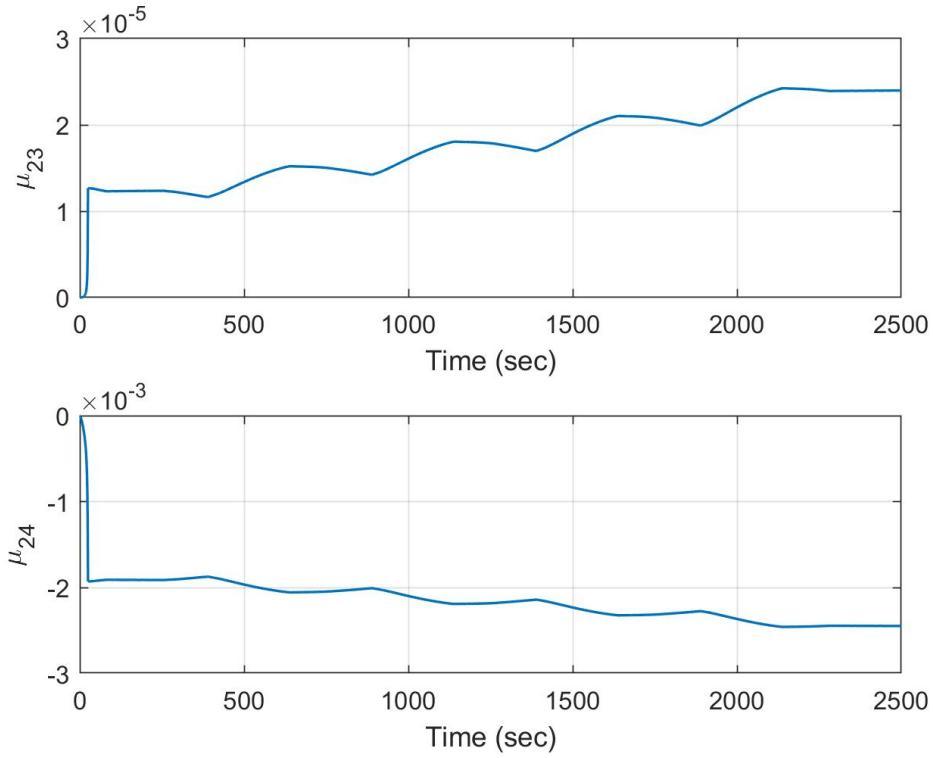

Figure 3.14 : The alteration of adaptive $\mu_{23}(.)$ and $\mu_{24}(.)$ parameters for the sinusoidal input.

The simulation results for the noisy case are depicted in Figures 3.15 and 3.16. Since Gaussian noise is added to the controlled outputs, there are small steady state errors in the result. However, the system can still track the staircase trajectory. The alteration of the adaptive control parameters $\mu_{ij}(.)$ for the noisy case is given in Figure 3.17, 3.18, 3.19 and 3.20.

As it is mentioned earlier, the system parameters have to be known accurately in model based control techniques. Nonetheless, there are always inaccuracies in modelling. Since the control method proposed in this thesis is based on estimation of system models by using an intelligent modelling technique, it should not be affected by modelling inaccuracies. Therefore, it is expected that this method has to be robust for the case of parameter uncertainty.

To examine the parameter estimation and the tracking performance of the system for the case with parametric uncertainty, the outflow parameter az_{13} has been selected as an uncertain parameter. Figure 3.21 illustrates its behaviour which varies as $az_{13} = 0.22 + 0.2sin(0.0133\pi t)$.

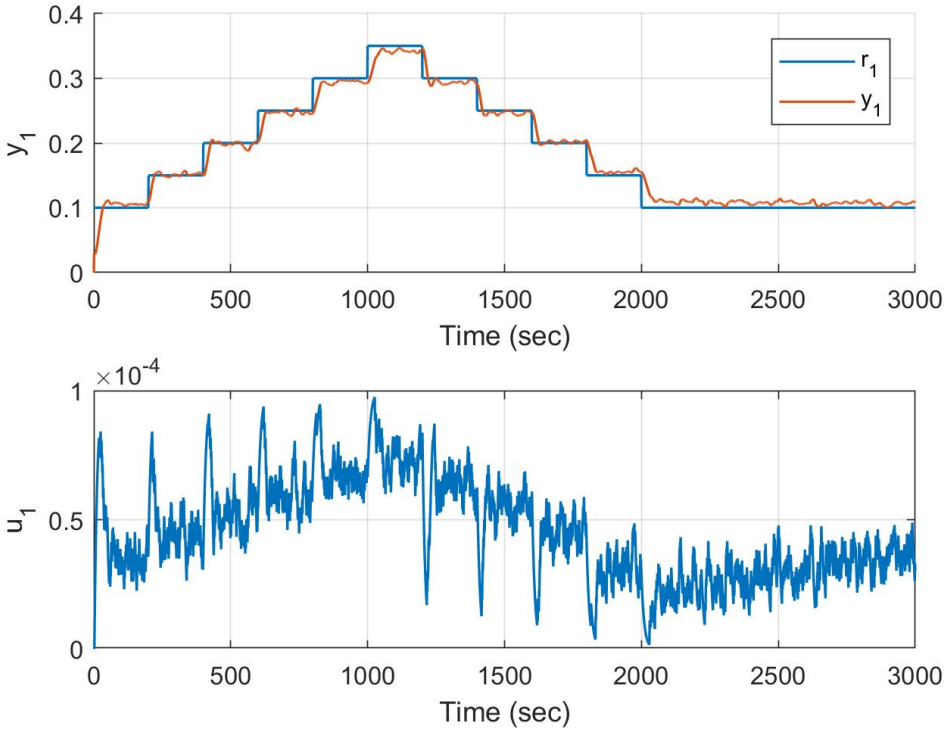

Figure 3.15 : The reference tracking of tank 1 and the inflow rate of pump 1 for the case with Gaussian noise.

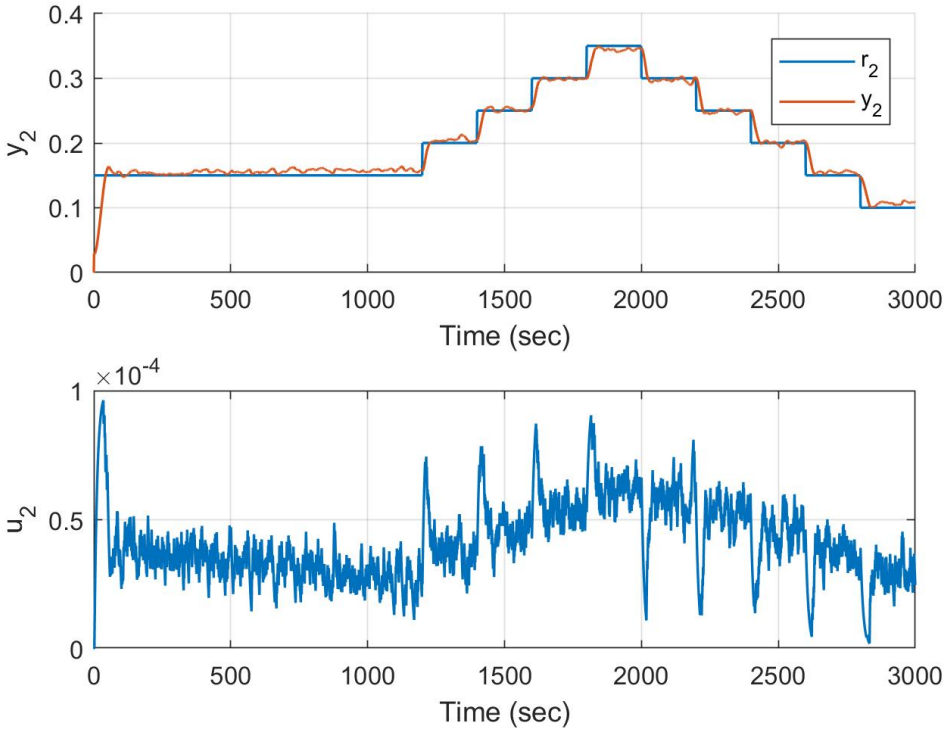

Figure 3.16 : The reference tracking of tank 2 and the inflow rate of pump 2 for the case with Gaussian noise.

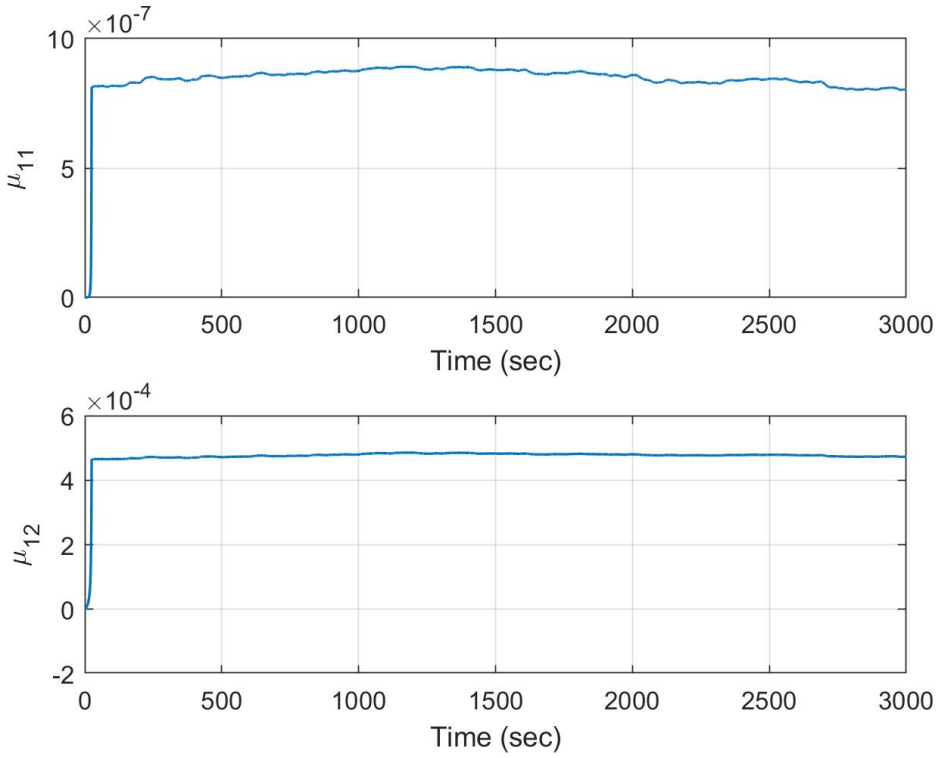

Figure 3.17 : The alteration of adaptive $\mu_{11}(.)$ and $\mu_{12}(.)$ parameters for the case with Gaussian noise.

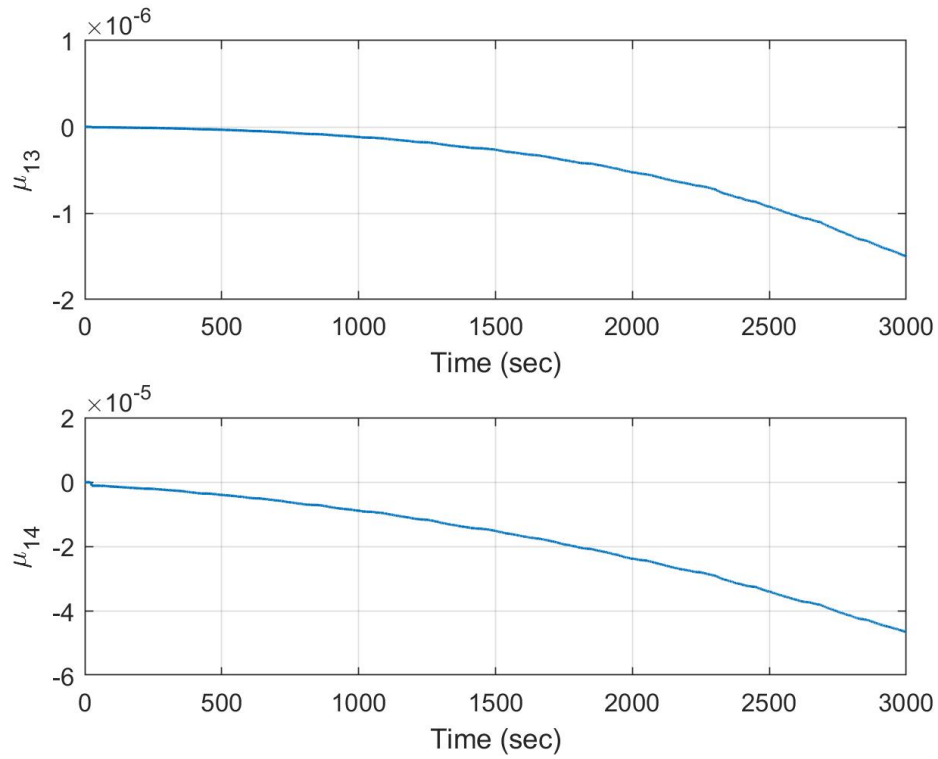

Figure 3.18 : The alteration of adaptive $\mu_{13}(.)$ and $\mu_{14}(.)$ parameters for case with Gaussian noise.

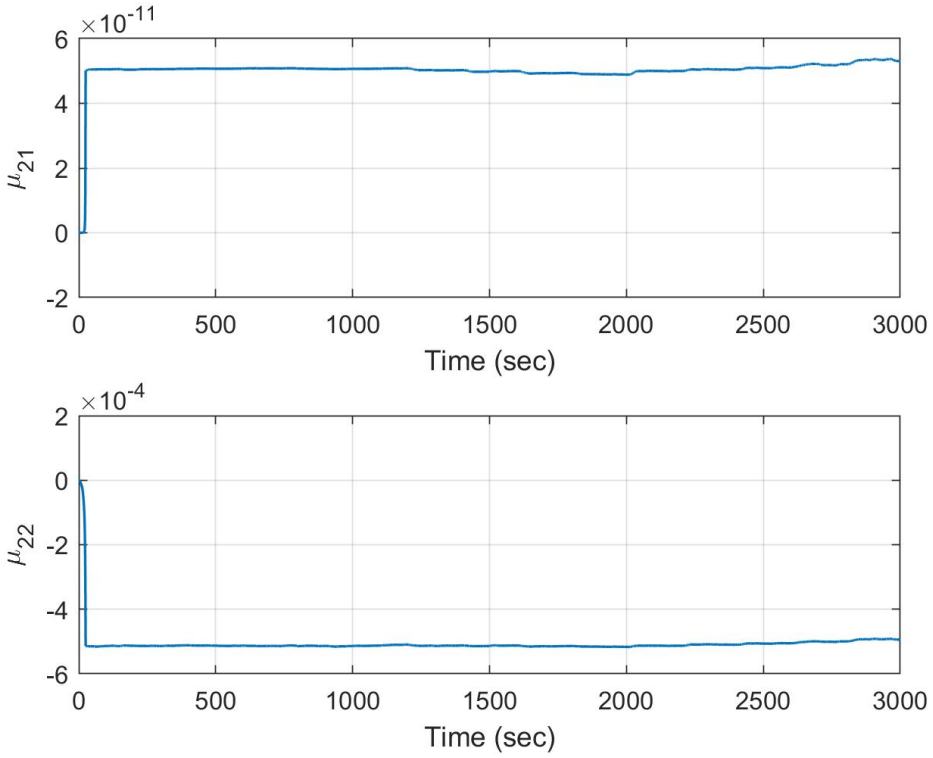

Figure 3.19 : The alteration of adaptive $\mu_{21}(.)$ and $\mu_{22}(.)$ parameters for case with Gaussian noise.

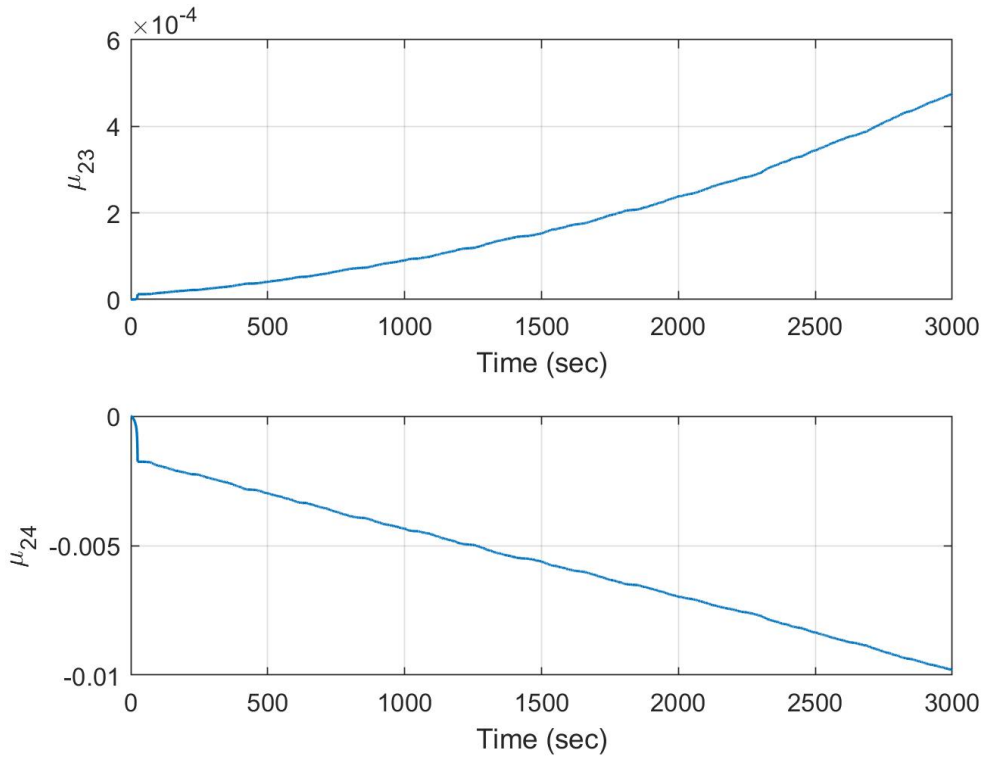

Figure 3.20 : The alteration of adaptive $\mu_{23}(.)$ and $\mu_{24}(.)$ parameters for case with Gaussian noise.

The simulations for the case with parameter uncertainty have been carried out by setting the reference signal for Tank 1 to 0.25 and for Tank 2 to 0.2. The system responses and the control signals for this case can be observed in Figures 3.22 and 3.23.

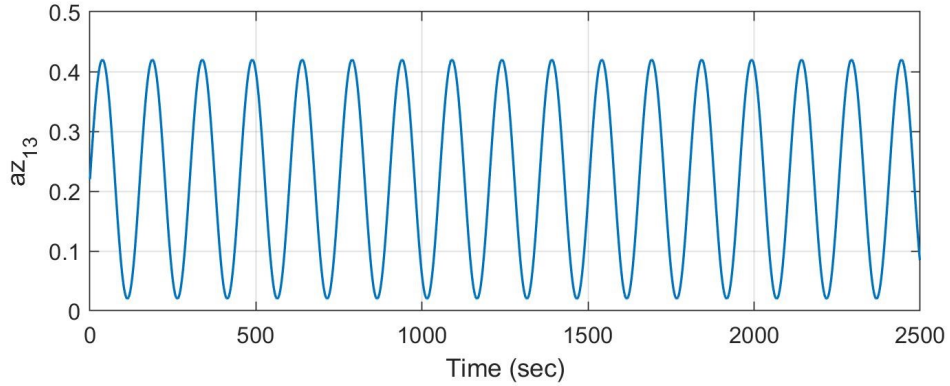

Figure 3.21 : Uncertain outflow parameter $az_{13}(t)$.

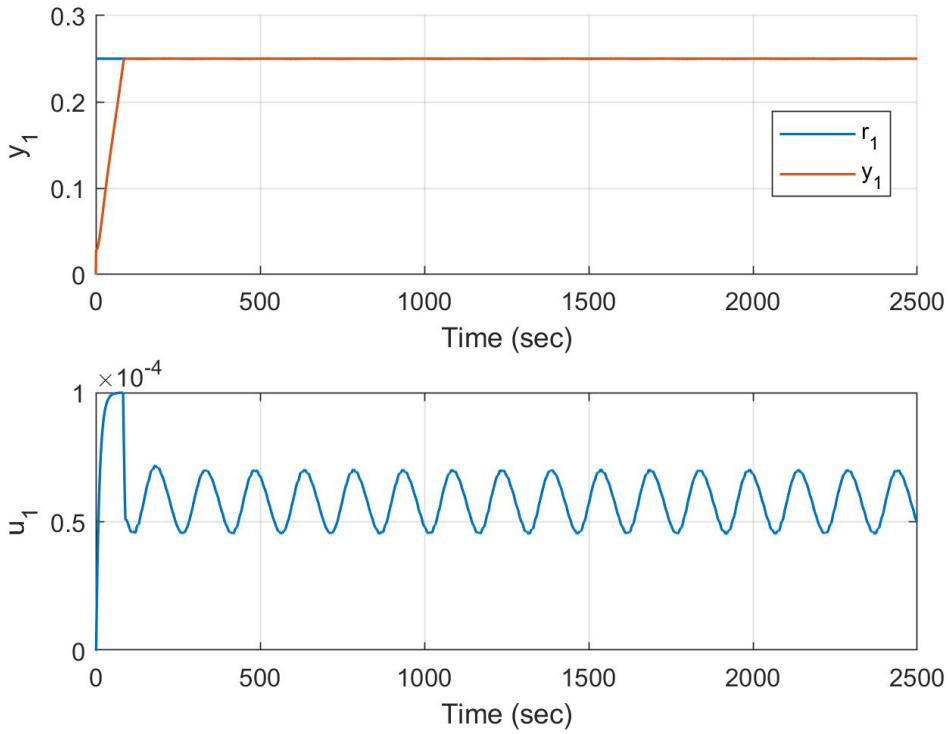

Figure 3.22 : The reference tracking of Tank 1 and the inflow rate of Pump 1 for the case with parametric uncertainty

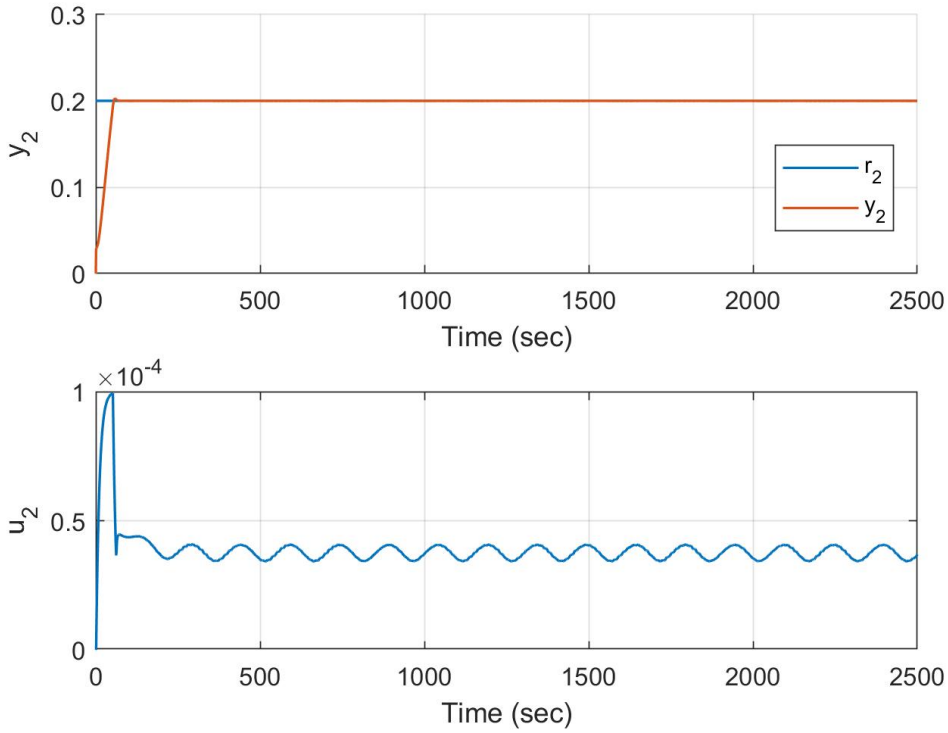

Figure 3.23 : The reference tracking of Tank 2 and the inflow rate of Pump 2 for the case with parametric uncertainty.

4. COMPUTED TORQUE CONTROL USING ONLINE LSSVR

The computed torque control or also called inverse dynamics control method is one of the well-known and commonly used robot control techniques that can be applied to many robot manipulators as well as to other nonlinear systems. The purpose of this technique is to cancel the nonlinear dynamics of the system and to provide a good trajectory tracking. However, this method requires that the system dynamics are known precisely. Practically, the robot manipulators are coupled and highly nonlinear systems. Therefore, having the exact knowledge about the dynamic model is very troublesome. There will always be imprecision in the model that may cause troubles in control. To solve this problem, intelligent approaches have been used to estimate the dynamic model more precisely. One of them has been utilised in [26]. In [26], SVR based computed torque control has been designed for a 2-DOF robot manipulator as an offline control. In this thesis, computed torque control method has been combined with the online LSSVR based NARMA-L2 controller explained in chapter 3 and an online computed torque control has been proposed.

4.1 Inverse Dynamics Control

The dynamical equation of a robot manipulator is given in equation (4.1).

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(q) \tag{4.1}$$

where $q \in R^n$ is the position vector, $M(q) \in R^{nxn}$ is the inertia matrix, $C(q,\dot{q})\dot{q} \in R^n$ is the centripetal and Coriolis forces, $G(q) \in R^n$ is the gravitational forces, $F(q) \in R^n$ is the frictional forces and $\tau \in R^n$ is the vector of applied torques.

The aim of computed torque control is to calculate a control input τ that compels the robot to track the desired trajectory. Figure 4.1 illustrates the control method that will be discussed in this section [26].

As it can be observed in Figure 4.1, the control input τ is made up of two parts. The inverse dynamics which is the primary controller eliminates the nonlinearities. Yet,

the inverse dynamics cannot be known exactly, the secondary controller is designed for removing the errors and improving the response.

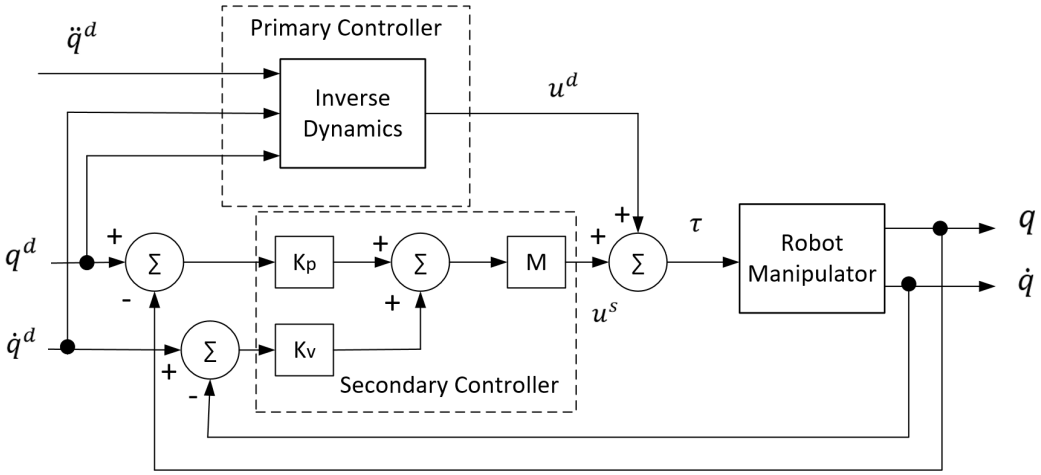

Figure 4.1 : An offline computed torque control structure

Equation (4.1) can be rewritten as,

$$\tau = M(q)\ddot{q} + V(q,\dot{q}) \tag{4.2}$$

where

$$V(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q) + F(q) \tag{4.3}$$

The control input is calculated with the summation of two distinct controllers as follows:

$$\tau = u^d + u^s \tag{4.4}$$

where

$$u^d = M(q^d)q^d + V(q^d, q^d) \tag{4.5}$$

$$u^s = M(q^d)(K_p e + K_v e) \tag{4.6}$$

Here, q^d, $\dot{q}^d$, $\ddot{q}^d$ are the desired position, velocity and acceleration vectors, respectively. e and $\dot{e}$ are the tracking errors.

$$e = q^d - q \tag{4.7}$$

$$\dot{e} = \dot{q}^d - \dot{q} \tag{4.8}$$

Under the assumption that the robot dynamics are known accurately, $M(q^d) = M(q)$ and $V(q^d,\dot{q}^d) = V(q^d,\dot{q}^d)$, nonlinearities are cancelled. If equation (4.4) is substituted into equation (4.2),

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \tag{4.9}$$

K_p and K_v are selected to control the dynamics of the error equation. However, practically the error equation cannot be zero as the robot dynamics are not known accurately. Hence, the equation (4.9) turns into equation (4.10).

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{M}^{-1}(q) \left[\tilde{M}(q) \ddot{q} + \tilde{V}(q, \dot{q}) + v' \right] \tag{4.10}$$

where $\hat{M}(q)$ and $\hat{V}(q, \dot{q})$ are the estimates of $M(q^d)$ and $V(q^d, \dot{q}^d)$, respectively. $\tilde{M}(q) = M(q) - \hat{M}(q)$ and $\tilde{V}(q, \dot{q}) = V(q, \dot{q}) - \hat{V}(q, \dot{q})$ are the errors in estimations and v' is the function of unknown disturbances.

In the next section, estimation of the dynamics will be explained by means of online LSSVR based NARMA-L2 model.

4.2 Online Computed Torque Control By Online LSSVR Based NARMA-L2 Controller

The dynamic model of a robot manipulator given in equation 4.1 is rewritten as follows.

$$\ddot{q} = -M(q)^{-1} V(q, \dot{q}) + M(q)^{-1} \tau \tag{4.11}$$

where

$$V(q, \dot{q}) = C(q, \dot{q}) \dot{q} + G(q) + F(q) \tag{4.12}$$

We recall the NARMA-L2 model discussed in chapter 3,

$$y_{n+1} = \hat{f}_n + \hat{g}_n u_n \tag{4.13}$$

It can be noticed that there is a similarity between equation (4.11) and (4.13). If the following equalizations are done, the NARMA-L2 model of a robot manipulator is obtained easily.

$$y_{n+1} = \ddot{q} \tag{4.14}$$

$$\hat{f}_n = -M(q)^{-1} V(q, \dot{q}) \tag{4.15}$$

$$\hat{g}_n = M(q)^{-1} \tag{4.16}$$

$$u_n = \tau \tag{4.17}$$

By using this model, we can achieve the NARMA-L2 controller which is u^d in this case. It is also necessary to compute M matrix separately. In chapter 3, we have already examined the computation of f_n, g_n and u_n by using online LSSVR, $-M(q)^{-1}V(q,q)$,

$M(q)^{-1}$ and τ can be calculated easily by utilizing the same method. By taking the inverse of g_n, we can manage to calcualate M matrix. For a two-link robot manipulator, the derivations are shown as follows:

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = - \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}^{-1} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}^{-1} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{4.18}$$

$$\begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \end{bmatrix} = \begin{bmatrix} \dfrac{-m_{22}v_1 + m_{12}v_2}{m_{11}m_{22} - m_{12}m_{21}} \\ \dfrac{m_{21}v_1 - m_{22}v_2}{m_{11}m_{22} - m_{12}m_{21}} \end{bmatrix} \tag{4.19}$$

$$\begin{bmatrix} \hat{g}_{11} & \hat{g}_{12} \\ \hat{g}_{21} & \hat{g}_{22} \end{bmatrix} = \begin{bmatrix} \dfrac{m_{22}}{m_{11}m_{22} - m_{12}m_{21}} & \dfrac{-m_{12}}{m_{11}m_{22} - m_{12}m_{21}} \\ \dfrac{-m_{21}}{m_{11}m_{22} - m_{12}m_{21}} & \dfrac{m_{11}}{m_{11}m_{22} - m_{12}m_{21}} \end{bmatrix} \tag{4.20}$$

In chapter 3, $\hat{f}_i$ and $\hat{g}_{ij}$ are given as LSSVR models in equation (3.18) and (3.19). Parameter approximations of these models are done by using equation (3.28), (3.29), (3.30) and (3.31). Since $\hat{g}_{ij}$ can be calculated, the inverse of the inertia matrix M^{-1} can also be calculated. Therefore, by taking the inverse of equation (4.20), we can achieve the estimated inertia matrix $\hat{M}$. The proposed control method is shown in Figure 4.2.

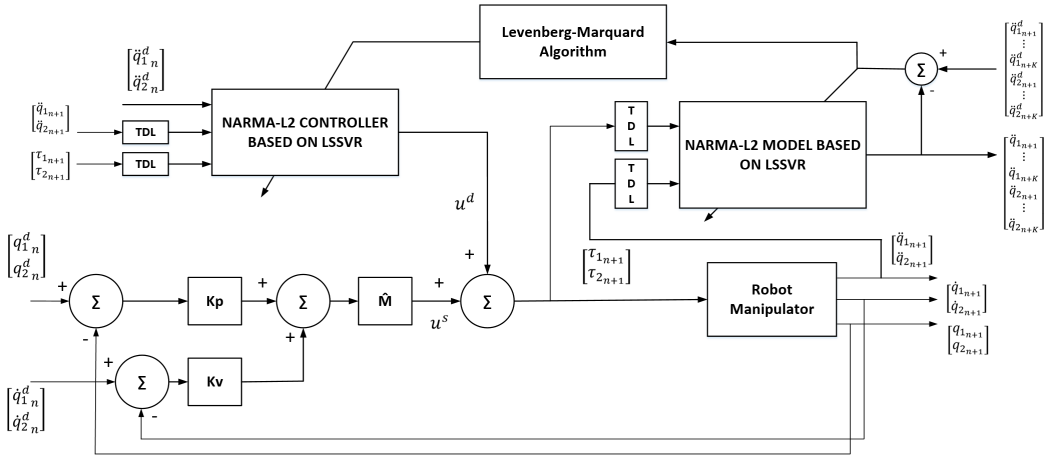

Figure 4.2 : Online computed torque control structure by online LSSVR based NARMA-L2 controller

As it can be seen in Figure 4.2, the primary control signal u^d is calculated by utilizing online LSSVR based NARMA-L2 controller. First of all, the system model is obtained as the NARMA-L2 model via online LSSVR. Then, this model is reconstructed to compute the NARMA-L2 controller. In every iteration, the adaptive parameters inside the NARMA-L2 model and accordingly the NARMA-L2 controller are optimized via Levenberg-Marquard Algorithm. The secondary controller is a PD controller. In order

to realize the dynamics of the error equation given in equation (4.9), the approximated inertia matrix $\hat{M}$ is added and the secondary control signal u^s is obtained. The final control signal is the sum of u^d and u^s.

4.3 Simulations of Online Inverse Dynamics Control On 2-DOF Robot Manipulator

The proposed controller is tested on a 2-DOF robot arm given in Figure 4.3.

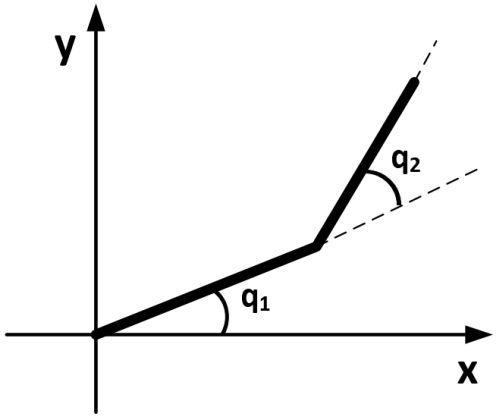

Figure 4.3 : Joint angles of the manipulator

The parameter values of the manipulator are listed in Table 4.1 [34].

Table 4.1 : Parameter values of the manipulator

Parameters	Value	Parameter	Value
Rotor Inertia of motor$_1$	0.267	Length of link$_1$	0.359
Inertia of link$_1$	0.334	Length of link$_2$	0.24
Rotor Inertia of motor$_2$	0.0075	CG distance of link$_1$	0.136
Stator Inertia of motor$_2$	0.040	CG distance of link$_2$	0.102
Inertia of link$_2$	0.063	Friction of link$_1$	4.90
Mass of motor$_1$	73.0	Friction of link$_2$	1.67
Mass of link$_1$	9.78	Torque$_1$ limitation	245.0
Mass of motor$_2$	3.0	Torque$_2$ limitation	39.2
Mass of link$_2$	4.45		

The dynamical equation of the manipulator is given below:

$$M(q)\ddot{q} + V(q,\dot{q}) + F = u \tag{4.21}$$

Here, q represents the joint angles, q_1 and q_2. u indicates the vector of applied torques τ_1 and τ_2. F is the Coulomb friction. The inertia matrix M and the vector of Coriolis

and centripetal forces V are given below.

$$M(q) = \begin{pmatrix} p_1 + 2p_3 cos(q_2) & p_2 + p_3 cos(q_2) \\ p_2 + p_3 cos(q_2) & p_2 \end{pmatrix} \tag{4.22}$$

$$V(q,q) = \begin{pmatrix} -q_2 (2q_1 + q_2) p_3 sin(q_2) \\ q_1^2 p_3 sin(q_2) \end{pmatrix} \tag{4.23}$$

where $p_1 = 3.1877$, $p_2 = 0.1168$, and $p_3 = 0.1630$. Simulations have been carried out by the FORK method. The sampling time is selected as $T_s = 1ms$.

The design parameters are selected as follows: The number of the previous inputs and outputs n_{u_1}, n_{u_2}, n_{y_1} and n_{y_2} are again chosen as 3. Exponential Radial Basis Function is selected as the Kernel function of LSSVR. The Kernel parameters and the regularization parameter are set to $\sigma_1 = \sigma_2 = 0.5$ and $C = 1000$, respectively. The length of the sliding window and the prediction horizon are fixed as $L = 25$ and $K = 5$, respectively. The penalty parameter is chosen as $\lambda = 0.1$.

In addition, the following first order low pass filters are applied to the controllers.

$$H_1(z) = \frac{0.7135 z^{-1}}{1 - 0.2865 z^{-1}} \tag{4.24}$$

$$H_2(z) = \frac{0.0645 z^{-1}}{1 - 0.9355 z^{-1}} \tag{4.25}$$

The coefficients of the PD controllers are found via Genetic Algorithm and tuned for a better result as follows:

$$K_{p_1} = 25; \quad K_{d_1} = 15 \tag{4.26}$$

$$K_{p_2} = 20; \quad K_{d_2} = 5 \tag{4.27}$$

Two different sinusoidal references have been applied to the system. First of all, only PD controllers have been used. The performance of the PD controller is depicted in Figures 4.4 and 4.5. The steady state errors can be observed in the simulation results.

Secondly, the simulations have been executed with the proposed controller. The results are observed in Figures 4.6 and 4.7. The steady state errors have been eliminated and the system displays a good trajectory tracking performance. The alteration of μ_{ij} parameters are also given in Figures 4.8, 4.9, 4.10 and 4.11.

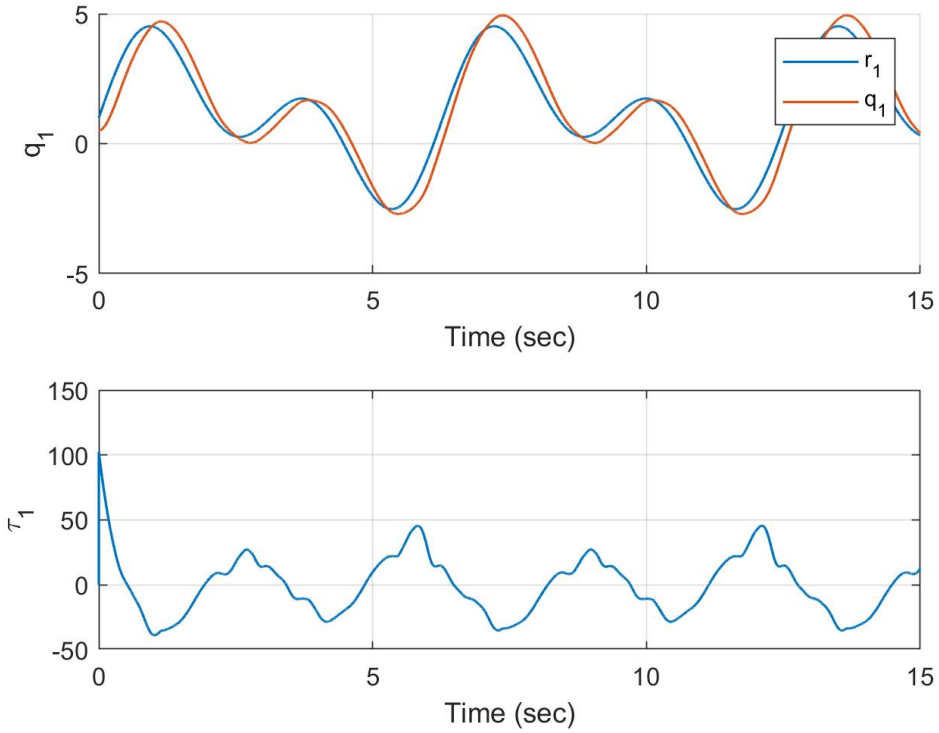

Figure 4.4 : Trajectory tracking and control input of link 1 with PD controller

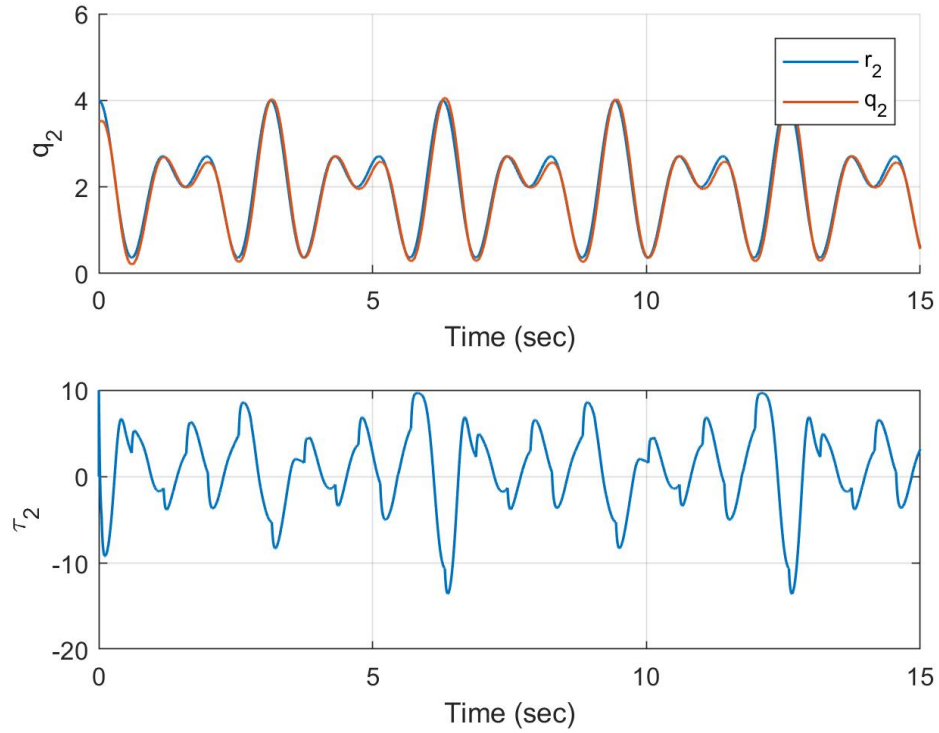

Figure 4.5 : Trajectory tracking and control input of link 2 with PD controller

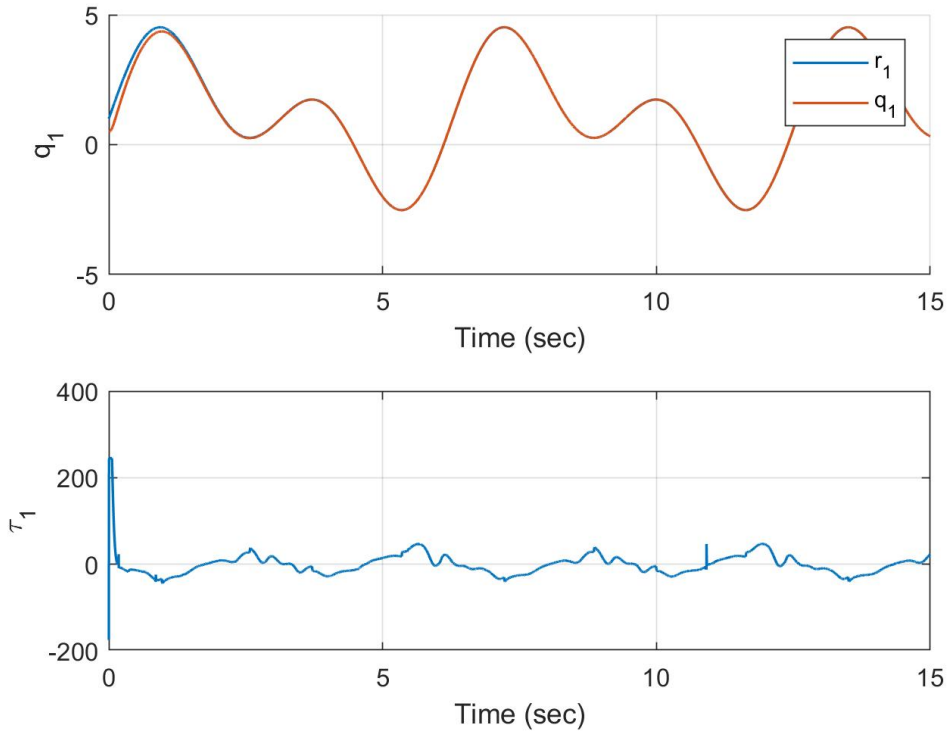

Figure 4.6 : Trajectory tracking and control input of link 1 with the proposed controller

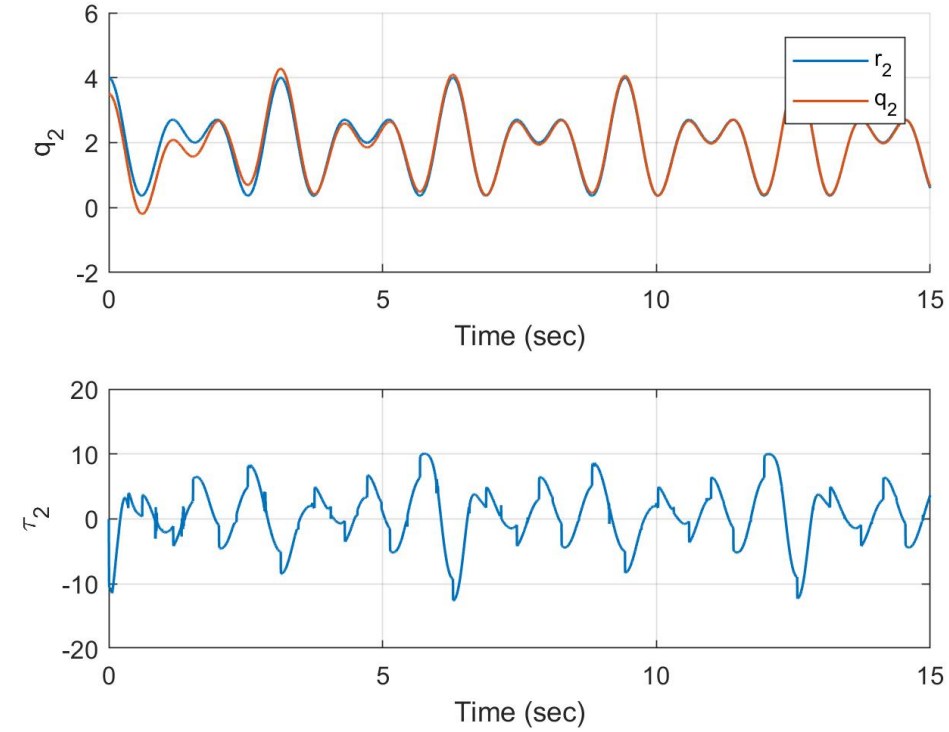

Figure 4.7 : Trajectory tracking and control input of link 2 with the proposed controller

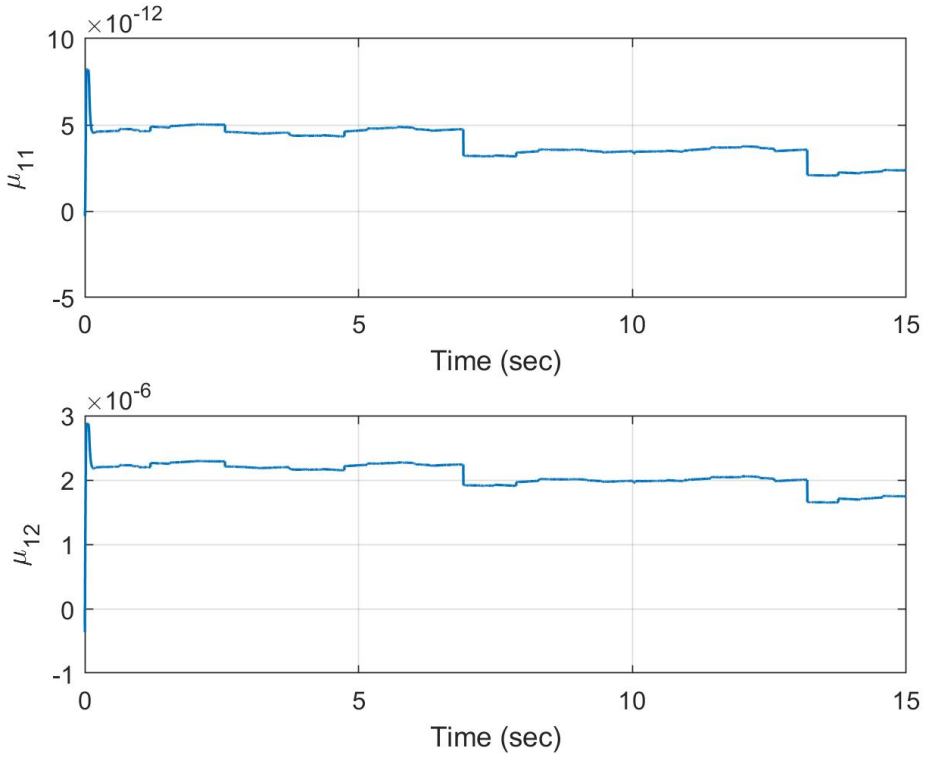

Figure 4.8 : The alteration of adaptive $\mu_{11}(.)$ and $\mu_{12}(.)$ parameters

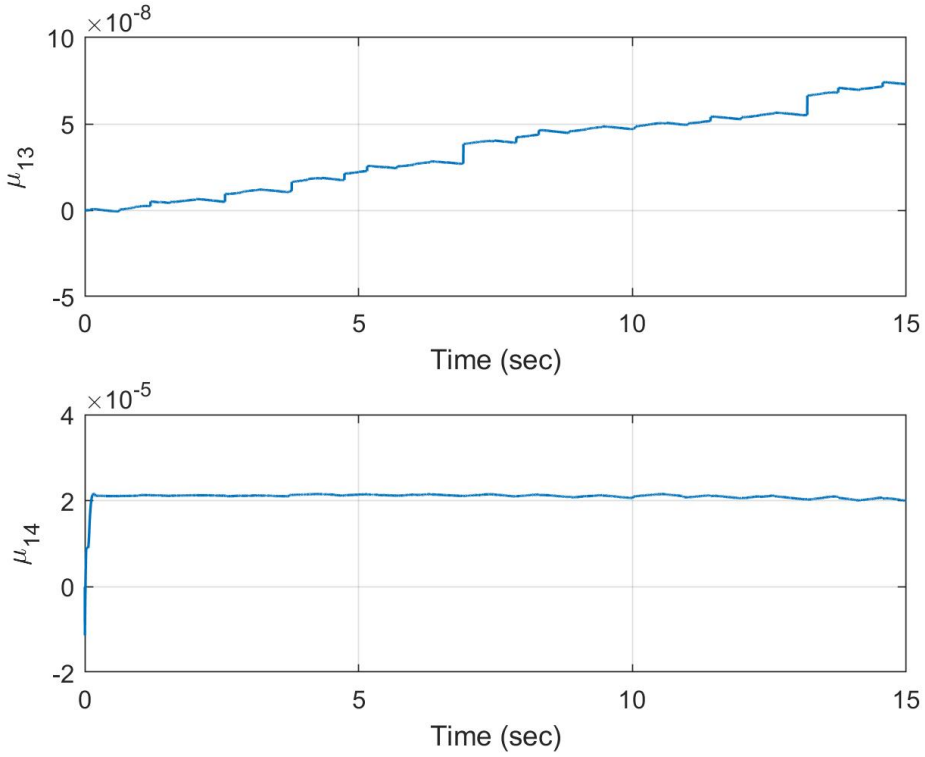

Figure 4.9 : The alteration of adaptive $\mu_{13}(.)$ and $\mu_{14}(.)$ parameters

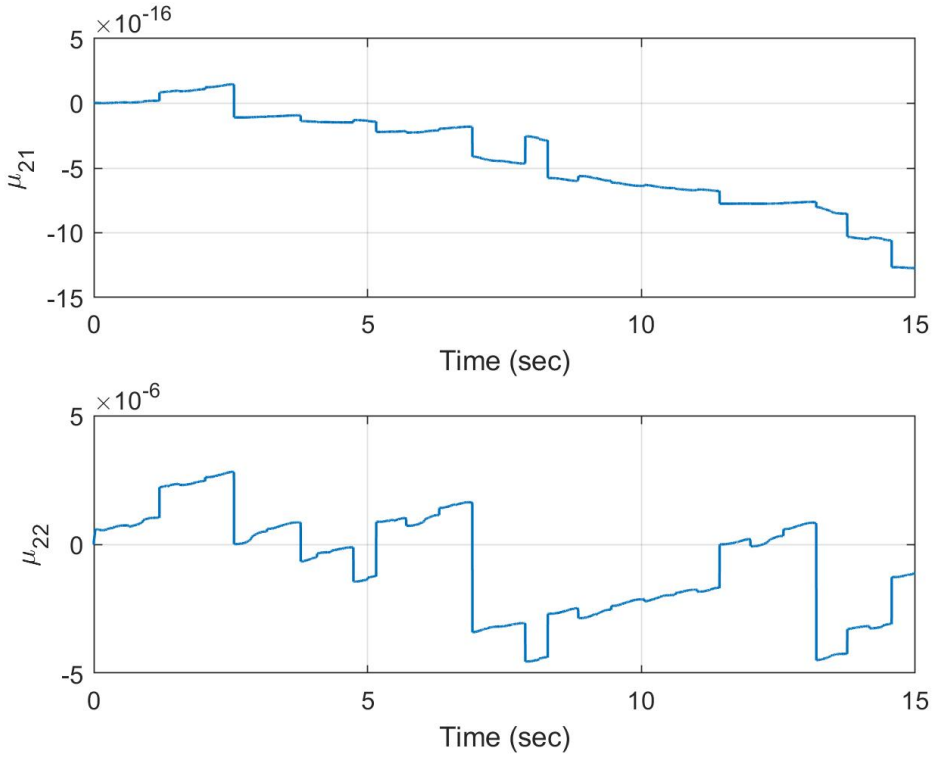

Figure 4.10 : The alteration of adaptive $\mu_{21}(.)$ and $\mu_{22}(.)$ parameters

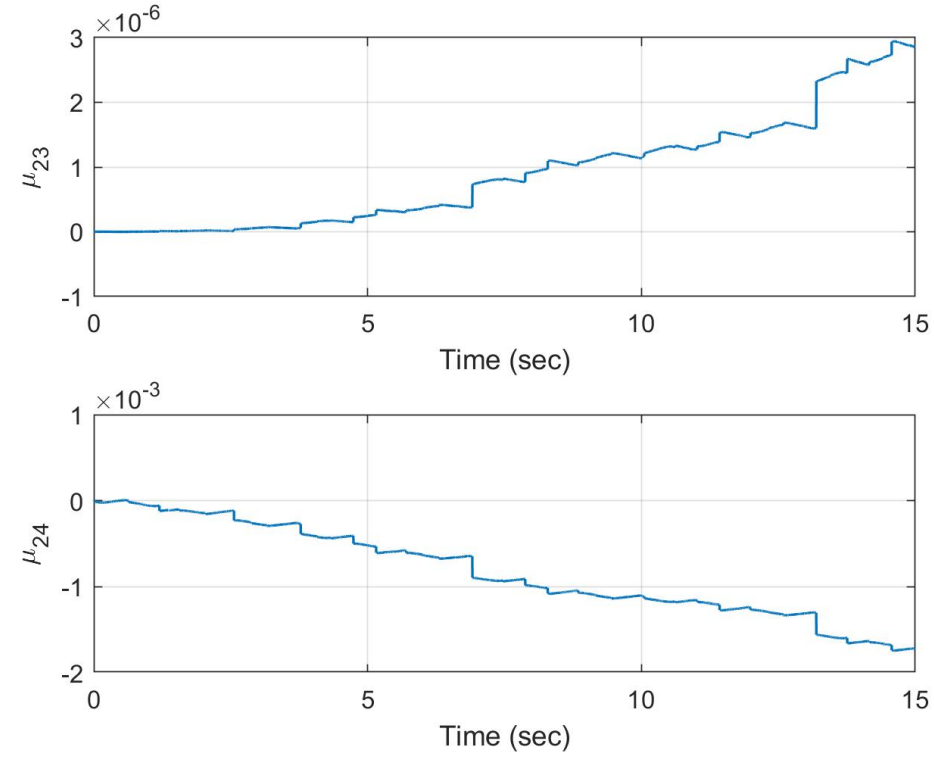

Figure 4.11 : The alteration of adaptive $\mu_{23}(.)$ and $\mu_{24}(.)$ parameters

5. CONCLUSIONS

In this thesis, firstly a NARMA-L2 controller has been designed for TITO nonlinear systems by using online LSSVR. The NARMA-L2 controller is approximated by reconstructing the NARMA-L2 model. Since the NARMA-L2 model consists of two submodels, first of all, the NARX model has been obtained from the current input-output data. Then, the system dynamics obtained by means of NARX model decomposed into the NARMA-L2 submodels. The decomposition procedure of the NARX model has been accomplished by utilizing the principle approach proposed in [5]. In [5], this approach has been utilized for SISO nonlinear systems. This thesis contributes to the method by extending the approach to be applied to TITO nonlinear systems. Moreover, oscillatory behaviour of the controller has been compensated by adding a linear feedback and a first order filter. As another contribution to the method, online LSSVR has been used for the estimation of the NARX model instead of online ε-SVR. In the literature, neural network based approaches have been widely used to estimate system models in model based adaptive control methods. However, backpropagation based techniques have a drawback of getting stuck at local extremum, thus they can only estimate the system locally. SVR is based on the structural risk minimization principle. Accordingly, its generalization competency is very high due to having a convex cost function. The reason behind the preference of LSSVR is that besides having a good generalization property, LSSVR also reduces the computational burden of ε-SVR by transforming the quadratic programming problem into a set of linear equations. The success of the designed control technique is determined by the simulations executed on a three tank system. The asynchronous staircase references and sinusoidal references have been enforced to the system and the results show that the outputs track the references properly. The robustness of the system has been also confirmed by the results after adding the Gaussian measurement noise and making a system paramter uncertain.

Secondly, by using the proposed NARMA-L2 controller based on online LSSVR, an online computed torque control has been designed. In [26], an offline SVR based computed torque controller has been proposed. In order to implement an online computed torque controller, an estimate of the inertia matrix has to be known. In this thesis, the inverse dynamics of the robot manipulator has been expressed by using the NARMA-L2 model explained previously and then the inertia matrix has been obtained separately. Finally, an online computed torque controller comprised of primary and secondary controllers has been achieved. The simulations have been carried out on a 2-DOF robot arm. The sinusoidal references has been applied to test the success of the controller. The robustness and the good trajectory tracking performance of the system is justified by the results.

REFERENCES

[1] **Amari, S.** *et al.* (2003). *The handbook of brain theory and neural networks*, MIT press.

[2] **Narendra, K.S. and Mukhopadhyay, S.** (1997). Adaptive control using neural networks and approximate models, *IEEE Transactions on neural networks*, *8*(3), 475–485.

[3] **Majstorovic, M.**, **Nikolic, I.**, **Radovic, J. and Kvascev, G.** (2008). Neural network control approach for a two-tank system, *2008 9th Symposium on Neural Network Applications in Electrical Engineering*, IEEE, pp.215–218.

[4] **De Jesus, O.**, **Pukrittayakamee, A. and Hagan, M.T.** (2001). A comparison of neural network control algorithms, *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, volume 1, IEEE, pp.521–526.

[5] **Uçak, K. and Günel, G.Ö.** (2016). A novel adaptive NARMA-L2 controller based on online support vector regression for nonlinear systems, *Neural Processing Letters*, *44*(3), 857–886.

[6] **Pukrittayakamee, A.**, **De Jesús, O. and Hagan, M.T.** (2002). Smoothing the control action for NARMA-L2 controllers, *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002.*, volume 3, IEEE, pp.III–III.

[7] **Mokri, S.S.**, **Shafie, A.A.** *et al.* (2008). Real time implementation of NARMA L2 feedback linearization and smoothed NARMA L2 controls of a single link manipulator, *2008 International Conference on Computer and Communication Engineering*, IEEE, pp.691–697.

[8] **Efe, M.O. and Kaynak, O.** (1999). A comparative study of neural network structures in identification of nonlinear systems, *Mechatronics*, *9*(3), 287–300.

[9] **Denai, M.A.**, **Palis, F. and Zeghbib, A.** (2004). ANFIS based modelling and control of non-linear systems: a tutorial, *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 4, IEEE, pp.3433–3438.

[10] **Gretton, A.**, **Doucet, A.**, **Herbrich, R.**, **Rayner, P.J. and Scholkopf, B.** (2001). Support vector regression for black-box system identification, *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing (Cat. No. 01TH8563)*, IEEE, pp.341–344.

[11] **Rong, H.**, **Zhang, G. and Zhang, C.** (2005). Application of support vector machines to nonlinear system identification, *Proceedings Autonomous Decentralized Systems, 2005. ISADS 2005.*, IEEE, pp.501–507.

[12] **Drezet, P. and Harrison, R.** (1998). Support vector machines for system identification.

[13] **Vapnik, V.** (2013). *The nature of statistical learning theory*, Springer science & business media.

[14] **Iplikci, S.** (2006). Support vector machines-based generalized predictive control, *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, *16*(17), 843–862.

[15] **Yao-Nan, W. and Xiao-Fang, Y.** (2008). SVM approximate-based internal model control strategy, *Acta automatica sinica*, *34*(2), 172–179.

[16] **Uçak, K. and Günel, G.Ö.** (2016). An adaptive support vector regressor controller for nonlinear systems, *Soft Computing*, *20*(7), 2531–2556.

[17] **Uçak, K. and Günel, G.Ö.** (2017). Generalized self-tuning regulator based on online support vector regression, *Neural Computing and Applications*, *28*(1), 775–801.

[18] **Suykens, J.A.** (2001). Nonlinear modelling and support vector machines, *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No. 01CH 37188)*, volume 1, IEEE, pp.287–294.

[19] **Wang, H. and Hu, D.** (2005). Comparison of SVM and LS-SVM for regression, *2005 International Conference on Neural Networks and Brain*, volume 1, IEEE, pp.279–283.

[20] **Zhu, Y.f. and Mao, Z.y.** (2004). Online optimal modeling of LS-SVM based on time window, *2004 IEEE International Conference on Industrial Technology, 2004. IEEE ICIT'04.*, volume 3, IEEE, pp.1325–1330.

[21] **Wanfeng, S.**, **Shengdun, Z. and Yajing, S.** (2008). Adaptive PID controller based on online LSSVM identification, *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, IEEE, pp.694–698.

[22] **Uçak, K. and Öke, G.** (2011). Adaptive PID controller based on online LSSVR with kernel tuning, *2011 International Symposium on Innovations in Intelligent Systems and Applications*, IEEE, pp.241–247.

[23] **Efe, M.O. and Kaynak, O.** (2000). A comparative study of soft-computing methodologies in identification of robotic manipulators, *Robotics and autonomous systems*, *30*(3), 221–230.

[24] **Shuzhi, S.G.**, **Hang, C.C. and Woon, L.** (1997). Adaptive neural network control of robot manipulators in task space, *IEEE transactions on industrial electronics*, *44*(6), 746–752.

[25] **Jin, Y.** (1998). Decentralized adaptive fuzzy control of robot manipulators, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *28*(1), 47–57.

[26] **Abdessemed, F.** (2012). Svm-based control system for a robot manipulator, *International Journal of Advanced Robotic Systems*, *9*(6), 247.

[27] **Gunn, S.R.** *et al.* (1998). Support vector machines for classification and regression, *ISIS technical report*, *14*(1), 5–16.

[28] **Smola, A.J. and Schölkopf, B.** (2004). A tutorial on support vector regression, *Statistics and computing*, *14*(3), 199–222.

[29] **Nelles, O.** (2013). *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*, Springer Science & Business Media.

[30] **Iplikci, S.** (2010). A comparative study on a novel model-based PID tuning and control mechanism for nonlinear systems, *International Journal of Robust and Nonlinear Control*, *20*(13), 1483–1501.

[31] **DTS200, A.** (1996). Laboratory setup three tank system, *Amira Gmbh, Duisburgh, Germany*.

[32] **Iplikci, S.** (2010). A support vector machine based control application to the experimental three-tank system, *ISA transactions*, *49*(3), 376–386.

[33] **Basheer, I.A. and Hajmeer, M.** (2000). Artificial neural networks: fundamentals, computing, design, and application, *Journal of microbiological methods*, *43*(1), 3–31.

[34] **Erbatur, K.**, **Kaynak, M.O. and Sabanovic, A.** (1999). A study on robustness property of sliding-mode controllers: A novel design and experimental investigations, *IEEE Transactions on Industrial Electronics*, *46*(5), 1012–1018.

CURRICULUM VITAE

Name Surname: Gökçen Devlet Şen

Place and Date of Birth: Adalar, 07.06.1993

E-Mail: sengo@itu.edu.tr

EDUCATION:

- **B.Sc.:** 2016, Istanbul Technical University, Faculty of Electrical and Electronics, Department of Control and Automation Engineering

PROFESSIONAL EXPERIENCE:

- Research Assistant, Istanbul Technical University (2017, to date)

—