

SOFT COMPUTING TECHNIQUES FOR ROBUST CONTROLLER DESIGN OF STEWART PLATFORM MANIPULATORS

A THESIS

*Submitted in partial fulfilment of the
requirements for the award of the degree*

of

DOCTOR OF PHILOSOPHY

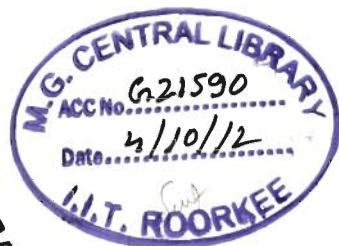
in

ELECTRONICS AND COMPUTER ENGINEERING

(With Specialization in Control and Guidance)

By

DEREJE SHIFERAW NEGASH



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
FEBRUARY, 2011**

©Indian Institute of Technology Roorkee, Roorkee – 2011
ALL RIGHTS RESERVED



INDIAN INSTITUTE OF TECHNOLOGY ROORKEE ROORKEE

CANDIDATE'S DECLARATION

I hereby declare that the work which is being presented in the Ph.D. thesis entitled “**Soft Computing Techniques for Robust Controller Design of Stewart Platform Manipulators**” in partial fulfilment of the requirement for the award of the Degree of Doctor of Philosophy(Ph.D.) in control and guidance and submitted in the department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out by me from August 2007 to February 2011 under the supervision and guidance of Dr. R. Mitra, professor, Electronics and Computer Engineering Department, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India.


I have not submitted the matter embodied in this thesis for the award of any other degree of this or any other institute.


(Dereje Shiferaw Negash)


This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 04 February 2011

Place: Roorkee


(R. Mitra)
Supervisor

The Ph.D. Viva-Voice Examination of **Dereje Shiferaw Negash**, Research Scholar, has been held on ----- 19.12.2011 -----


Signature of Supervisor


Signature of External Examiner

THESIS ERRATA

Page	Line/ Section	Is	Should be
2	1	interconnection	interconnections
2	13	Mass/load	Load/mass
7	15	x	X
9	5/1.1.5	these force/ torque which has to be	these force/torque signals which have to be
52	Fig. 4.1	Not defined	q_i is actual measured leg length for the i^{th} leg and q_{id} is the desired leg length of the i^{th} leg
63	5.1.2i	ΔM not defined	Uncertainty in inertia matrix M
63	5.1.2	ΔC not defined	Uncertainty in coriolis torque C
63	5.1.2	ΔG not defined	Uncertainty in Gravitational torque G
63	5.1.2	Mm not defined	Mm is maximum bound of ΔM
63	5.1.2	Cm not defined	Cm is maximum bound of ΔC
63	5.1.2	Gm not defined	Gm is maximum bound of ΔG
104	13/6.1	Sliding dynamics and gives enables	Sliding dynamics and enables

ABSTRACT

In the last few decades, there has been an increasing interest in model based robust controllers because of two main factors. The first one is technological advances and the other is due to challenges such as demand for higher level of automation to reduce lifecycle costs, issues such as maintenance on demand, zero standstill, and fault tolerance. Due to these challenges linear and decoupled control algorithms have proved to be rather inefficient. On the other hand advances in the fields of microprocessors, microcontrollers, digital signal processors and communication have made implementation of complex and advanced control algorithms very much easier and practical and hence have given an impetus to the field. Added to this, economic and social factors like market competition and increase in demand for high quality products have boosted the need for model based robust and adaptive controllers. This is especially true in control of robotic systems as they are the main workforce in industrial manufacturing and the quest is to achieve improved interaction of robotic systems with humans, to design controllers for efficient task coordination between multiple robots and to design high precision robots having improved rigidity and stiffness. The latter one can be achieved by using parallel robots such as the Stewart platform manipulators and utilizing soft computing techniques for their robust control.

Stewart platform manipulators are six DOF parallel robots having fixed base and movable platform joined by six extensible legs. They have high structural rigidity and stiffness and are preferred over serial robots for applications such as precision machining, robotic surgery and so on. However the absence of robust controllers, which are able to compensate their nonlinear dynamics and uncertainties due to model inaccuracies, parameter variations, and external disturbances, has limited the real application of the manipulators to low speed motion simulators only. Presently, only single input single output PID controllers are in practical use and these controllers have a number of drawbacks including lack of synchronization and low performance. To tackle these problems, some researchers have

proposed various robust and adaptive controllers which are based on hard computing. Nevertheless results have shown that the controllers were unable to tolerate the uncertainties and their performance degrades when the manipulator moves at high speeds. Therefore using soft computing techniques for the design of a robust controller for Stewart platform manipulators is expected to result in a better and effective controller.

Soft computing is a word coined by L.A. Zadeh and refers to a methodology which tends to fuse synergically the different aspects of fuzzy logic, neural network, genetic algorithm and other evolutionary algorithms to achieve a system that is tolerant of imprecision, uncertainty, partial truth and approximation. Researches in the last few decades have witnessed the successful applications of soft computing to aerospace industry, communication systems, consumer appliances, electric power systems and manufacturing automation including serial robots. However, the application of this important tool for the control of parallel robots has not been investigated. This thesis is a step in this direction. To this end, the thesis tries to systematically combine and pull together three important areas: nonlinear robust control, with emphasis on sliding mode control, soft computing and Stewart platform manipulator control.

The thesis begins by presenting a state of the art literature review on sliding mode control, on Stewart platform manipulator modeling and control and on application of soft computing techniques for robust control. Then five important applications of soft computing techniques for robust control are proposed and implemented through simulations. The first application discussed in the thesis is solving forward kinematics problem of the manipulator. Forward kinematics is the computation of end-effector/platform position and orientation from given leg length values and is necessary to close the feedback loop in task space control of the manipulator. Nonetheless, unlike serial manipulators, in Stewart platform manipulator its mathematical formulation is highly nonlinear and coupled making computation highly complex and time taking. To solve this problem, the thesis deals with exhaustive comparison between the performance of two estimation methods, the hard computing Newton Raphson numerical method and the soft computing neural networks method. The performance metrics used for the comparison are: estimation error for position and orientation and average time taken, tolerance to external disturbances and uncertainties present in the manipulator. The methods are compared using various trajectories. Simulation results showed that, the

numerical algorithm, irrespective of initial conditions taken, always has more estimation error than neural network. Moreover, it was found that numerical algorithm takes longer average time while neural network takes less average time with uniform estimation error for all trajectories. Hence we found that using neural network improves control performance.

The second application of soft computing techniques dealt in the thesis is improving robustness of existing controllers. In Stewart platform manipulator, there are two basic approaches used for controller design, namely joint space approach and task space approach. In the joint space approach, the controller is a collection of single input single output (SISO) systems implemented using local information on each actuator length only and the coupling between legs is ignored or is considered as a disturbance. The most important and practically used joint space SISO controller is PID control. The advantage of this approach is that local information required for feedback is obtained easily using simple sensors and the control algorithm is easy for parallel implementation. Due to this, the control algorithm is able to execute reasonably fast. But the performance of such controllers quickly degrades when the manipulator speed is increased. In the thesis, a solution to this problem is proposed using fuzzy logic, where the three gain parameters of the PID controller are tuned using fuzzy logic system. Though using fuzzy logic to tune PID controllers has been proposed by earlier researchers for other systems, it has not been used for Stewart platform. Moreover, in our proposed controller, the fuzzy logic system has additional advantage of achieving synchronization. Hence in our proposal, like the joint space PID control, each leg is controlled by a PID controller but the gains are varied by a fuzzy logic system. The input to each fuzzy logic system is a weighted sum of errors in all legs and the rate of change of the weighted sum of errors. The weight factor is taken based on intuition and it enables to minimize the coupling error between the legs, which is drawback of independent leg PID control. Simulation results showed that the controller has better performance than simple PID controller in terms of tracking accuracy and robustness against parameter uncertainty.

The third application of soft computing techniques discussed in the thesis is in the design of three types of sliding mode controllers. These are: task space fuzzy sliding mode controller (TFSMC), fuzzy sliding mode controller with integral loop (FSMCPI) and hybrid sliding mode controller with synchronization error. The FSMC presented utilizes the full dynamics of the manipulator and fuzzy logic is used as switching controller. Task space

position and velocity errors are used as inputs to the fuzzy logic system and their universe of discourse is selected based on disturbance bounds. Simulation results have shown that it performs better than joint space PID but it has small control signal chattering which is the usual problem of sliding mode controllers. To solve this problem, external PI loop is used to enhance the performance of the controller. The assumption taken is that the PI loop serves as a long time average calculator and minimizes chattering. Simulation results have shown that the FSMCPI performs better than both independent PID controller and fuzzy sliding mode controller. However, a better result is obtained by using hybrid implementation. The hybrid sliding mode controller is a combination of task space and joint space approaches. The switching controller part of the sliding mode controller is implemented in joint space and the model based equivalent control part is implemented in task space. The hybrid structure makes the controller easier to implement and avoids the need for forward kinematics estimation. Moreover the controller uses a newly proposed sliding surface which helps to drive synchronization error to zero and the controller achieves high performance in task space.

The fourth application of soft computing used in this thesis is evolutionary computing technique of genetic algorithm. In this thesis, it is used to solve an important design problem, viz., design of integral sliding mode controller for systems having unmatched uncertainty. There are various methods proposed to solve the basic drawbacks of classical sliding mode controller. These include: higher order sliding modes, boundary layer methods and integral sliding mode control. Integral sliding mode control (ISMC) is an improvement over conventional SMC and uses a nonlinear sliding surface having an integral term. It is able to remove reaching phase problem of conventional sliding mode by using a sliding surface which is designed to constrain the system states to be on sliding mode from initial time. Moreover, the sliding surface of ISMC improves the stability of sliding dynamics and it attempts to enhance robustness against unmatched uncertainties. Nonetheless the design of sliding surface of ISMC is not a simple task and has no formal methods, especially for nonlinear systems with unmatched uncertainty. To solve this problem, the design of integral sliding mode controller is formulated as optimization problem and genetic algorithm is used for its solution. The application of the method to SISO and MIMO systems is discussed using examples. Finally genetic algorithm based multi-objective optimization is proposed as a

design method for design of integral sliding mode controller for Stewart platform manipulators. The simulation results of the controller designed using the proposed method shows that the method can effectively be used for the design of ISMC for MIMO systems having unmatched uncertainty.

In the last part of the thesis, as fifth application of soft computing technique, a new controller having a better robustness and performance is presented. The controller is a neuro-fuzzy sliding mode controller. The controller has two parts: fuzzy logic system and neural network. The fuzzy logic and neural network parts are used concurrently but each part is responsible for one phase of sliding mode controller. The fuzzy logic system is utilized to control reaching phase dynamics and the feedforward neural network is employed to keep the system states on the sliding surface. The neural network is trained online using modified back propagation algorithm. When the controller is used in a closed loop system, initially the fuzzy logic system part of the controller is dominant and has bigger output but as the system moves from reaching phase to sliding phase, the neural network part becomes more active as it learns the dynamics of the system. This hybrid computing paradigm is effective to avoid chattering and to better handle uncertainties. The stability of the system is analyzed using Lyapunov's direct method. The proposed controller is implemented to regulate a second order nonlinear uncertain system and simulation results confirm that the proposed system reduces chattering and improves transient response.

All in all, the thesis deals with the various ways of using soft computing techniques for robust and high performance control of Stewart platform manipulator. The stability of all of the robust controllers designed for Stewart platform manipulator have been analyzed using Lyapunov's method and were implemented using Simmechanics toolbox of MATLAB and simulink. The controllers have been checked by taking wide uncertainty limits to minimize challenges in practical implementation. All of the simulations have shown promising results. The controllers discussed in the thesis are not only useful for the advancement of the field of robotics control but also have wider contribution to the field of nonlinear system control.

To
My mother
Late TEZERASH ABEBE

ACKNOWLEDGMENTS

Thanks be to God for His indescribable gift!

First of all I would like to extend my heartfelt gratitude and thanks to my supervisor **Dr. R. Mitra, professor, department of Electronics and Computer Engineering**. This work is unimaginable without his unreserved and brilliant guidance and encouragement. Throughout the research work he was professionally supportive, helping and also socially positive, understanding and encouraging.

I should also thank all faculty members of the department and all other supportive staff as this work is not only mine but also a fruit of their labor. Especially I thank Research Committee members: Dr. R. C. Joshi and Dr. S. N. Sinha, E & CE department and Dr. H. O. Gupta, electrical department for their willingness to evaluate the research work and for the unreserved comments which they forwarded to me.

I will also take this opportunity to express my special thanks to my beloved wife and sweetheart, Mrs. Yegilenesh Habte for her continuous encouragement and moral support. Your sweet words of love and brilliant advice have led me to this point. During this research work, it was you who shared the most burdens. I have no words to express my thanks to you, my darling, but simply I will say I love you.

In the same manner, I would like to say thank you to my father, Mr. Shiferaw Negash and my sisters and brothers for their love and affection whose sweet fragrance kept me going at hard times. I should also thank my Ethiopian friends in Roorkee. Each and every aspect of life we shared together was remarkable and unforgettable. Without the humor and spiritual nourishment we had together, I cannot think of the work to be like this.

The friendship and brotherly love extended to me from Indian colleagues in the department was remarkable and invaluable. B. B. Gupta, B. M. Patel, Keshavamurthy B.N., Manju and many many others whom I have not listed your name here, I happily say to you thanks a lot for you were with me in all times. There is also one special friend, who deserves a word of thanks, Ries Ahmed, who has been cooking our foods. ‘Dangerous man Beyya‘ thanks a lot, you have been a great help shouldering the burden of cooking.

I have to give special thanks to faculty members in Adama University, department of Electrical/Electronics Technology for their support and cooperation by taking the entire burden for my study leave.

Last but not least, I extend my thanks to the ministry of education of the Ethiopian government for giving me this scholarship.

Glory to God in the highest and on earth peace, God will towards men!!

LIST OF FIGURES

- Fig.1.1 A fully parallel 6 DOF Stewart platform manipulator
- Fig.1.2 Coordinate frame assignment for geometric and kinematic description of Stewart platform manipulator
- Fig. 1.3 Joint space trajectory tracking control approach for a Stewart platform manipulator
- Fig. 1.4 Task space trajectory tracking control approach for a Stewart platform manipulator
- Fig.3.1. Generalized Stewart platform
- Fig.3.2. Three layer feed forward neural network
- Fig3.3. Neural network training
- Fig.3.4. Plot of the random values of x , y , z , α , β and γ values used for training data generation
- Fig.3.5 Training performance of three layer feed-forward network
- Fig.4.1 Fuzzy tuned joint space PID control of a single leg
- Fig.4.2 Simulink block diagram model of single leg PID tuned controller
- Fig.4.3 Graph of membership functions for error E
- Fig.4.4 Graph of membership functions for error rate, ER
- Fig.4.5 Membership functions for output variable
- Fig.4.6a) Three dimensional view of the desired trajectory
- Fig.4.6b) Desired trajectory in x and y directions
- Fig.4.6c) Desired trajectory in Z direction
- Fig.4.7. Joint space tracking error of simple PID
- Fig.4.8. Joint space tracking error of fuzzy tuned PID
- Fig.4.9. Task space tracking error for simple PID
- Fig.4.10. Task space tracking errors of fuzzy tuned PID for x , y and z directions
- Fig.5.1 Block diagram representation of task space fuzzy sliding mode control system.
- Fig.5.2 Graph of membership functions for the input S
- Fig.5.3 Membership functions of the output variable U

- Fig. 5.4a) Estimation error of the task space positions using numerical algorithm
- Fig. 5.4b) Estimation error of the task space orientation angles when using numerical algorithm only
- Fig. 5.4c) Position estimation error when neural network is used with numerical algorithm
- Fig. 5.4d) Orientation angle estimation error when neural network is used with numerical algorithm
- Fig. 5.5) Step response error in x direction used for parameter selection, desired is rise time less than 50msec with no overshoot
- Fig. 5.6a) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller in x direction
- Fig. 5.6b) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller in y direction
- Fig. 5.6c) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller in z direction
- Fig. 5.6d) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller, roll angle
- Fig. 5.6e) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller, pitch angle
- Fig. 5.6f) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller, yaw angle
- Fig. 5.7a) Full load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller, x Direction

Fig. 5.7b) Full load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller, y direction

Fig. 5.8 Control forces in joint space

Fig. 5.9 Block diagram representation of the control system

Fig. 5.10 Comparison of x direction tracking performance of FSMC and the new controller, FSMCIP

Fig. 5.11 Comparison of y direction tracking performance of FSMCIP with fussy SMC

Fig. 5.12 Comparison of Tracking performance in Z direction of new controller(FSMCIP) and fuzzy SMC

Fig. 5.13 Three dimensional view of the helical trajectory used to compare the performance of FSMC and FSMCIP

Fig. 5.14a) Control signal of SMC for leg 1 and 2 when carrying 200Kg payload

Fig. 5.14b) Control signal of SMC for leg 3 and 4 when carrying 200Kg payload

Fig. 5.14c) Control signal of SMC for leg 5 and 6

Fig. 5.15a) Control force on leg 1, 2, 3 of the FSMCIP when carrying payload of 200Kg

Fig. 5.15b) control force on leg 4, 5, 6 of the FSMCIP when carrying a 200Kg payload

Fig. 5.16 Task space feedforward and joint space feedback with SMC control

Fig. 5.17 Task space tracking control in x direction

Fig. 5.18 Task space tracking control in y direction

Fig. 5.19 Task space tracking control in z direction

Fig. 5.20 Task space tracking control for roll angle

Fig. 5.21 Task space tracking control for pitch angle

Fig. 5.22 Task space tracking control for yaw angle

Fig. 5.23 Task space tracking control in x direction with full load and friction

Fig. 5.24 Task space tracking control in y direction with full load and friction

Fig. 5.25 Task space tracking control in z direction with full load and friction

Fig. 5.26 Task space tracking control in roll angle with full load and friction

Fig. 5.27 Task space tracking control in pitch angle with full load and friction

Fig. 5.28 Task space tracking control in roll angle with full load and friction

Fig. 5.29 Torque applied to leg 1 when platform is carrying a load

Fig. 5.30 Torque applied to leg 2 when platform is carrying a load
Fig. 5.31 Torque applied to leg 3 when platform is carrying a load
Fig. 5.32 Torque applied to leg 4 when platform is carrying a load
Fig. 5.33 Torque applied to leg 5 when platform is carrying a load
Fig. 5.34 Torque applied to leg 6 when platform is carrying a load
Fig.6.1 Block diagram representation of integral sliding mode control
Fig.6.2 Steps of Genetic Algorithm tuning for integral sliding mode controller design
Fig.6.3 Closed loop response, X_1
Fig.6.4 Closed response for X_2
Fig.6.5 Closed loop response X_3
Fig.6.6 The sliding parameter of genetic algorithm and LMI
Fig.6.7 Control signal is smooth and chattering is almost zero
Fig.6.8 Task space error in x direction, without load
Fig.6.9 Task space error in y direction, without load
Fig.6.10 Task space error in z direction, without load
Fig.6.11 Task space error in roll angle, without load
Fig. 6.12 Task space error in pitch angle, without load
Fig.6.13 Task space error in yaw angle, without load
Fig.6.14 Task space error in x direction with 200 Kg load
Fig.6.15 Task space error in y direction with 200 Kg load
Fig.6.16 Task space error in z direction with 200 Kg load
Fig.6.17 Task space error in roll angle with 200 Kg load
Fig.6.18 Task space error in pitch angle with 200 Kg load
Fig.6.19 Task space error in yaw angle with 200 Kg load
Fig.6.20 Control torque for leg 1 in the three controllers
Fig.6.21 Control torque for leg 2 in the three controllers
Fig.6.22 Control torque for leg 3 in the three controllers
Fig.6.23 Control torque for leg 4 in the three controllers
Fig.6.24 Control torque for leg 5 in the three controllers
Fig.6.25 Control torque for leg 6 in the three controller

- Fig. 6.26 Step response error in x direction used for parameter selection, desired is rise time less than 50msec with no overshoot
- Fig.6.27 Trajectory tracking performance in x direction of simple task space sliding mode controller and integral sliding mode controller with no load
- Fig.6.28 Trajectory tracking performance in y direction of simple sliding mode controller and integral sliding mode controller with no load
- Fig.6.29 Trajectory tracking performance in z direction of the simple task space sliding mode controller integral sliding mode controller with no load
- Fig.6.30 Trajectory tracking performance for roll angle of simple sliding mode controller and integral sliding mode controller with no load
- Fig.6.31 Trajectory tracking performance for pitch angle of simple sliding mode controller and integral sliding mode controller with no load
- Fig.6.32 Trajectory tracking performance for yaw angle of simple task space sliding mode controller and integral sliding mode controller with no load
- Fig.6.33 X direction trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg
- Fig.6.34 Y direction trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg
- Fig.6.35 Z direction trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg
- Fig.6.36 Roll angle trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg
- Fig.6.37 Pitch angle trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg
- Fig.6.38 Yaw angle trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg
- Fig.6.39 Control forces in joint space
- Fig.6.40 Control forces in joint space
- Fig.7.1 Structure of the neuro-fuzzy controller
- Fig.7.2 Neural network used for equivalent control estimation
- Fig.7.3 Membership functions for input variable

Fig.7. 4 Membership functions for the output variable

Fig.7.5 Nonlinear control surface of the fuzzy logic system

LIST OF TABLES

- Table 3.1 Geometric Specifications of Stewart platform
- Table 3.2 Cartesian position and Orientation Limits of Center of Platform
- Table 3.3 Comparison of different networks based on their training performance
- Table 3.4 Comparison between the performance of numerical and NN methods with respect to trajectory tracking
- Table5.1 Step response and parameters used
- Table5.2 No load tracking performance of the three controllers
- Table5.3 Tracking error performance of the three controllers when platform is carrying payload of 200Kg and actuator friction is considered
- Table6.1 Task space % error comparison (no load and no friction case)
- Table6.2 Task space % error comparison (full load and friction case)

LIST OF SYMBOLS

B_i	centers of i^{th} universal joint at the base of the Stewart platform manipulator
P_i	centers of i^{th} spherical joint at the platform of the Stewart platform manipulator
F_b	reference frame attached to the base
F_p	reference frame attached to the platform
b_i	position vector of the center of universal joints B_i in frame F_b
p_i	position vector of the center of spherical joints P_i in frame F_p
p_{ix}	x coordinate of the position vector of the center of spherical joints P_i in frame F_p
p_{iy}	y coordinate of the position vector of the center of spherical joints P_i in frame F_p
p_{iz}	z coordinate of the position vector of the center of spherical joints P_i in frame F_p
O_p	origin of frame F_p
O_b	origin of frame F_b
r	the position of the origin of the platform frame with respect to the base frame
R	orientation matrix of frame F_p with respect to F_b
R_b	radius of the base
R_p	radius of the platform
φ	angle used to specify location of joints in the platform
δ	angle used to specify location of joints in the base
X	the Cartesian space position and orientation of the moveable platform or end effector
α	Euler rotation angle specifying yaw angle
β	Euler rotation angle specifying pitch angle
γ	Euler rotation angle specifying roll angle
a_i	unit vector in the direction of leg i
J	Jacobian matrix of the Stewart platform manipulator
q	vector consisting of length of legs
M	6x6 manipulator inertia matrix in task space
A	6x6 manipulator inertia matrix in joint space

C	6x6 coriolis and centrifugal torque matrix in task space
B	6x6 coriolis and centrifugal torque matrix in joint space
G	6x1 gravitational torque vector in task space
Q	6x1 gravitation torque vector in joint space
τ	Actuator torque applied at the legs
τ_d	external disturbance torque acting on the legs
τ_0	nominal control torque of integral sliding mode controller
f_f	friction torque acting on the actuators
F_p	total force and moment exerted on the platform
J_{vi}	Jacobian matrix that relates the velocity of the origin of the platform frame to the Cartesian velocity transferred from each leg to the platform.
J_i	kinematic Jacobian matrix of leg i and it gives the velocity transferred to the platform from each leg i.
H_i	the dynamics of leg i
x	an n dimensional real valued state vector
f	vector valued nonlinear function of dimension n
g	matrix valued nonlinear function of dimension nxm
u	control signal of appropriate dimension
d	lumped uncertainty
Λ	constant gain matrix of conventional sliding surface surface
K	gain of discontinuous control signal
Ψ	small constant used to denote the width of boundary layer in sliding mode control
α_f	output of fuzzy logic system used to tune PID controller
K_p	proportional gain
K_D	derivative gain constant
K_I	integral gain constant
Λ	Constant gain of sliding surface of simple sliding mode controller
G_s	gain matrix of integral sliding surface in integral sliding mode control
S	sliding surface or sliding parameter of a sliding mode controller
f_{obj}	objective function for genetic algorithm

Table of Content

Abstract	i
Acknowledgments	vii
List of figures	ix
List of tables	xiv
List of symbols and acronyms	xv
1 Introduction	1
1.1 Stewart platform manipulator	2
1.1.1 Geometry and kinematics	2
1.1.2 Dynamics	5
1.1.3 Actuators used to drive legs	8
1.1.4 Application areas	9
1.1.5 Control of Stewart platform manipulator	9
1.2 Robust control	11
1.3 Soft computing techniques	13
1.4 Objectives of the research	14
1.5 Motivation	15
1.6 Statement of the problem	15
2 Literature review	19
2 Introduction	19
2.1 Sliding mode controllers	19
2.1.1 Basic sliding mode controllers	19
2.1.2 Boundary layer and saturation control	21
2.1.3 Second order Sliding mode controllers	21
2.1.4 Integral sliding mode controllers	22
2.2 Stewart Platform manipulator control	22
2.2.1 Kinematic modeling	23
2.2.2 Dynamic modeling	24
2.2.3 Joint space controllers	27

2.2.4	Task space controllers	29
2.3	Soft computing techniques and sliding mode control	32
2.4	Gaps identified	33
3	Neural Network based forward kinematics estimation and its comparison with other methods	34
3.1	Forward kinematics of Stewart platform	35
3.2	Problem formulation	36
3.3	Numerical estimation method	37
3.4	Neural network estimation method	39
3.4.1	Training data generation	40
3.4.2	Selection of network size	40
3.5	Simulation results and comparison	41
3.5.1	Numerical method	41
3.5.2	Feed forward neural network method	42
3.5.2.1	Training data generation	43
3.5.2.2	Selection of network size	43
3.6	Conclusion	46
4	Fuzzy tuned Joint Space PID controller for Stewart platform manipulators	49
4.1	Introduction	50
4.2	Design of controller	51
4.2.1	Fuzzy logic system	51
4.2.2	Tuning algorithm	53
4.3	Simulation results and discussion	53
4.4	Conclusion	59
5	Sliding mode control and improvements for high performance	60
5.1	Task space fuzzy sliding mode controller	61
5.1.1	Design of controller	61
5.1.2	Stability analysis	63
5.1.3	Fuzzy logic switching surface	64
5.1.4	Simulation study and discussion	65
5.1.4.1	Controller implementation and performance	66

6.6 Comparison between joint space and task space ISMC	143
6.7 Conclusion	144
7 Neuro-fuzzy sliding mode control	145
7.1 Neuro-fuzzy hybridization and sliding mode control	145
7.2 Problem statement	147
7.3 Stability analysis	149
7.4 Neural network equivalent control approximation	150
7.5 Modified back propagation algorithm using sliding parameter	152
7.6 Application to SISO plant	153
7.7 Results and discussion	156
8 Conclusions and future work	159
Bibliography	

5.1.4.2 Discussion of results	68
5.2 Fuzzy sliding mode control with integral loop	76
5.2.1 Source of chattering in fuzzy sliding mode control and proposed approach	76
5.2.1 Design of controller	76
5.2.3 Simulation results and discussion	77
5.3 Hybrid sliding mode with newly proposed sliding surface	83
5.3.1 Design of the hybrid sliding mode controller	85
5.3.2 Robustness analysis	88
5.3.3 Simulation results and discussion	90
5.4 Conclusion	103
6 Genetic Algorithm based integral sliding mode controller design	104
6.1 Introduction	104
6.2 Integral sliding mode control	105
6.3 Integral sliding mode controller design using genetic algorithm	107
6.3.1 Genetic algorithm	107
6.3.2 Fitness function	108
6.3.3 Coding of genes	110
6.3.4 Determining range of values for the parameters	110
6.3.5 SISO system example with constant input and state matrix	111
6.4 Multi-objective genetic algorithm optimization and integral sliding mode controller	113
6.4.1 Multi-objective optimization	113
6.4.2 Selection of objective functions	117
6.5 Application to Stewart platform manipulator	118
6.5.1 Integral sliding mode controller design in joint space	118
6.5.1.1 Design and analysis of controller	118
6.5.1.2 Simulation results and discussion	120
6.5.2 Integral sliding mode controller design in task space	133
6.5.2.1 Design and analysis of controller	133
6.5.2.2 Simulation results and discussion	135

CHAPTER 1

INTRODUCTION

Robots have been in use for more than 60 years to automate industrial manufacturing and today, in addition to industrial manufacturing, there are more and more robots in various applications like medicine, military, housekeeping, elder care, and space exploration and so on[125, 128]. They generally have the following three basic components.

- a) mechanical links and joints
- b) sensors and actuators
- c) controller

The mechanical links and joints of a robot determine the shape, size, workspace and other basic characteristics of the robot while the sensors and actuators act as gateways to the external environment. The controller is the brain of the robot and it is one of the basic driving forces to the changes made in robotics and is the main subject of the thesis. However, design of robust and high performance controller needs the study of the mechanical links and joints and their modeling making it the first step in the design of controller for robotic systems.

There are two basic models that must be derived for any robotic system, the kinematic model and dynamic model. The kinematic model gives the position and velocity of the end effector of the robot as a function of the position and velocity of the joints of the robot, while the dynamic model gives the acceleration of the end effector as a function of the force required at the joints. Both the kinematic and dynamic models depend on the mechanical structure of the robot. Based on their mechanical structure, there are two types of robots, namely serial and parallel robots. The serial robots are the most common ones in industrial applications today,

to the extent of 90% [73, 80]. They are interconnection of rigid bodies linked one after the other by one-degree-of-freedom joints. The main advantage of these robots is their large work space and their kinematic design which reduce the mathematics of the robot geometry. The drawbacks of these robots are low load carrying capacity with respect to their mass and low precision. For example an Adept I800 4 DOF SCARA robot has a maximum load carrying capacity of 5.5kg whereas its mass is 34Kg giving a load/mass ratio of 0.1617 [74]. Their precision is also very low due to accumulation of the errors in the serially connected links and they suffer from extensive vibration. To solve these problems parallel robots have been proposed.

A parallel robot is defined as a closed loop mechanism having a fixed base and moveable platform joined by extendable legs or kinematic chains.[80]

In parallel manipulators the end effector is connected to the base through multiple links and hence the load is distributed to the links resulting in a higher mass/load ratio. The error in the links will also be averaged out rather than being summed and hence they have a higher precision in positioning. One of the most important members of the family of parallel robots is Stewart platform manipulator. It is a fully parallel robot having a base and a platform joined by extensible legs.

1.1 Stewart platform manipulator

The manipulating structure named as Stewart platform first appeared in literature in 1965 when D. Stewart proposed a design for a flight simulator [15] [38]. In practical applications, around 1950, V. E. Gough has successfully used a similar structure for tire testing but the name Stewart platform become more dominant in the literature [38][80]. In some literature the name Gough-Stewart is also used for the same mechanical structure. The basic geometrical, kinematics and dynamics of Stewart platform manipulators have been given in [73], [80], [160-162].

1.1.1 Geometry and kinematics

The Stewart platform manipulator has undergone various generalizing modifications from its initial proposal by D. Stewart and as it is understood now, it contains two rigid bodies connected by six extensible legs [38][15]. The first rigid body is known as the base and is

usually fixed. The other one is known as the platform and it is the end effector and is moveable. The extensible legs are joined at the base by either spherical or universal joints and to the platform by universal joints [80]. Six degree-of-freedom motion is obtained by changing the length of the legs. Fig.1.1. shows the general structure of a 6 DOF Stewart platform manipulator.

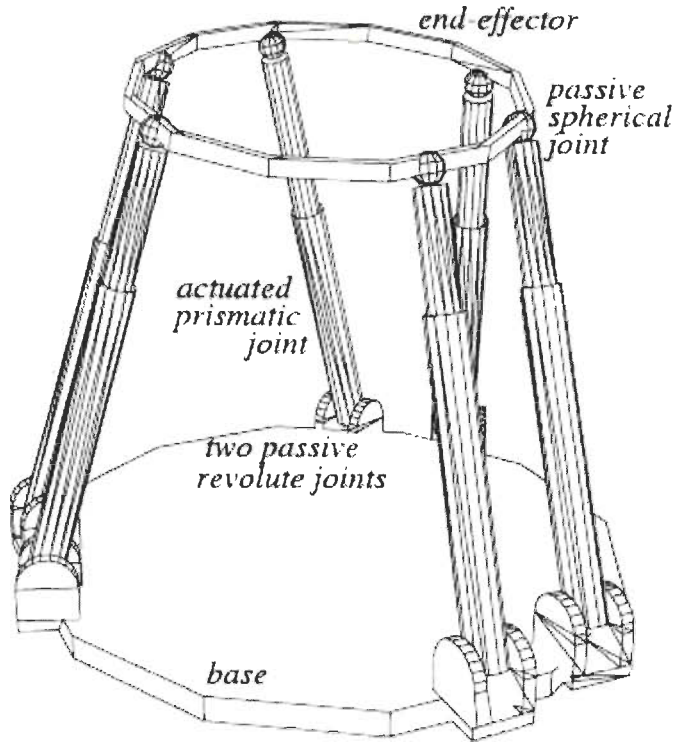


Fig.1.1 A fully parallel 6 DOF Stewart platform manipulator, [65]

The geometry and kinematics of Stewart platform manipulator has been a hot research topic from 1980's till the end of 1990's. An extensive review can be found in [15] and a review on some of the latest researches is given in chapter 2. For easy reference and understanding of the manipulator system, let us consider a Stewart platform with irregular hexagon base and platform as shown in Fig.1.2. For geometric description of the manipulator, coordinate frames are assigned to the base and platform as shown below. Let, the centers of the universal and spherical joints are denoted by B_i ($i = 1, 2 \dots 6$) and P_i ($i = 1, 2, \dots 6$) respectively and let reference frames F_b and F_p be attached to the base and the platform. Let also, the position vector of the center of universal joints B_i in frame F_b be b_i and the position vector of the center of spherical joints P_i in frame F_p be p_i . Then the position

vectors for the spherical joints and universal joints the in the platform and base frames respectively are given as:

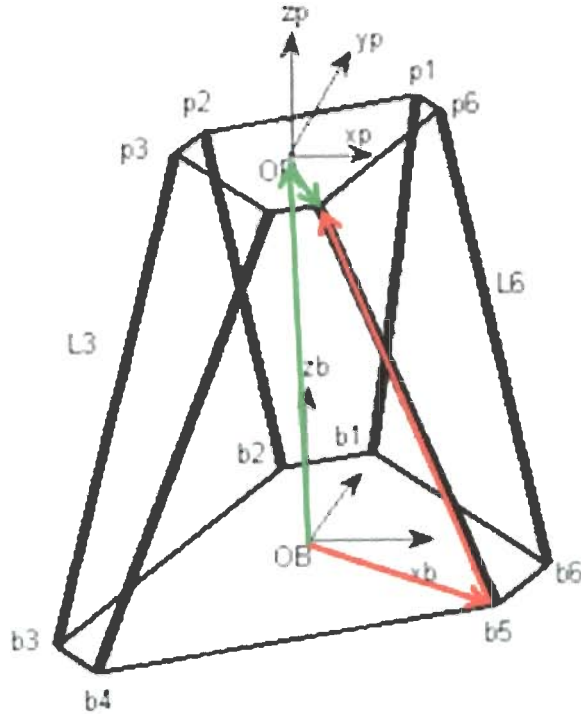


Fig.1.2 Coordinate frame assignment for geometric and kinematic description of Stewart platform manipulator

$$\begin{aligned} p_i^p &= [p_{ix} \quad p_{iy} \quad p_{iz}] \\ &= [R_p \cos(\varphi_i) \quad R_p \sin(\varphi_i) \quad 0] \end{aligned} \quad (1.1)$$

$$\begin{aligned} b_i^b &= [b_{ix} \quad b_{iy} \quad b_{iz}] \\ &= [R_b \cos(\delta_i) \quad R_b \sin(\delta_i) \quad 0] \end{aligned} \quad (1.2)$$

In (1.1) and (1.2), R_p and R_b are radius of the platform and base respectively, the angles φ_i and δ_i are angles used to specify location of joints in the platform and base. Let $r = [r_x, r_y, r_z]$ be the position of the origin O_p of frame F_p with respect to O_b and also let R denote the orientation matrix of frame F_p with respect to F_b . Thus the Cartesian space position and orientation of the moveable platform or end effector is specified by $X = [r_x, r_y, r_z, \alpha, \beta, \gamma]$ where the three angles α, β, γ are three rotation angles that constitute the transformation matrix R , which is given as:

$$R = \begin{bmatrix} C_\gamma C_\beta & -S_\gamma C_\alpha + C_\gamma S_\beta S_\alpha & S_\gamma S_\alpha + C_\alpha C_\gamma S_\beta \\ C_\beta S_\gamma & C_\gamma C_\alpha + S_\alpha S_\beta S_\gamma & -S_\alpha C_\gamma + C_\alpha S_\gamma S_\beta \\ -S_\beta & C_\beta S_\alpha & C_\beta C_\alpha \end{bmatrix} \quad (1.3)$$

Where C_γ is $\cos(\gamma)$ and S_β is $\sin(\beta)$ and so on. The location of the center of the spherical joints P_i with respect to the base frame is given by

$$p_i^b = R \cdot p_i + r \quad (1.4)$$

where R is the transformation matrix from platform to base and r is position of center of the platform with respect to base frame. Then length of i^{th} leg can be calculated from the vector closure relation shown in Fig.1.2 by the green and red colors and given as

$$\overline{O_b B_i} + \overline{B_i P_i} = \overline{O_b O_p} + \overline{O_p P_i} \quad (1.5)$$

Substituting the vectors by their symbols and rewriting the equation, the length of each leg which is length of vector $B_i P_i$ is given as

$$q_i = \|\overline{R p_i + r - b_i}\| \quad (1.6)$$

This means that if the desired position and orientation of the platform is given, then the length of each leg can be uniquely determined. The converse of (1.6) gives the forward kinematics, which is to calculate the position r and orientation angles of the platform for a given combination of leg lengths. This is highly nonlinear and coupled but is very important for feedback control. The velocity kinematics is obtained by differentiating (1.2) and it gives the velocity of the legs for a given vector of platform velocities as

$$\dot{q} = J \dot{X} \quad (1.7)$$

Where the Jacobian matrix J of the manipulator is given as

$$J_p^{-1} = \begin{bmatrix} a_1 & (R \cdot p_1 + r) \times a_1 \\ a_2 & (R \cdot p_2 + r) \times a_2 \\ \vdots & \vdots \\ a_6 & (R \cdot p_6 + r) \times a_6 \end{bmatrix} \quad (1.8)$$

and a_i is a unit vector in the direction of each leg.

1.1.2. Dynamics

The Dynamic modeling of Stewart platform manipulator can be done using the Lagrangian method or recursive Newton Euler method [7][10][15][80][161-162]. The Lagrangian

formulation gives a closed form equation, which is useful for the design of model based controllers, and it has been used by most authors. Another important point in the formulation of dynamic model of the manipulator is the variables used. The dynamic equation and hence control of the manipulator can be given either in joint space, as a function of the length, velocity and acceleration of legs (q, \dot{q}, \ddot{q}) or in task space using the position, velocity and acceleration of generalized vector $X=[x \ y \ z \ \alpha \ \beta \ \gamma]$ containing Cartesian position of platform center and its orientation. Each approach has its own advantages and disadvantages. The joint space approach is easier from closed loop control point of view as it does not need the use of forward kinematics in feedback loop. In this approach only leg lengths and velocities are needed. The first one can be easily measured by standard sensors and leg velocity can be obtained by differentiating the measured leg lengths and filtering it using low pass filter. But dynamic modeling is complex in joint space because of a nonlinear coordinate transformation needed [173]. On the other hand task space approach is easier for dynamic modeling but it needs the highly nonlinear forward kinematics for feedback controller implementation [58]. Using joint space approach, the dynamic equation is given by

$$A(q)\ddot{q} + B(q, \dot{q})\dot{q} + Q(q) = \tau - f_f - \tau_d \quad (1.9)$$

where q is the length of legs, A is 6x6 manipulator inertia matrix, B is also 6x6 coriolis and centrifugal torque/force, Q is 6x1 gravitational torque/force, τ is the actuator torque, f_f is torque due to friction and τ_d is disturbance torque.

In the task space, the dynamic equation is given by

$$M(X)\ddot{X} + C(X, \dot{X})\dot{X} + G(X) = J^T (\tau - f_f - \tau_d) \quad (1.10)$$

where X is the 6DOF generalized Cartesian position and orientation vector of the platform given by $X=[x, y, z, \alpha, \beta, \gamma]$. M is 6x6 manipulator inertia matrix, C is also 6x6 coriolis and centrifugal torque/force, G is 6x1 gravitational torque/force, J is 6x6 manipulator Jacobian, τ is the actuator torque, f_f is torque due to friction and τ_d is external disturbance torque.

Alternatively, writing platform dynamics and leg dynamics separately, the dynamic equation can be rewritten as:

$$(M_1(X) + M_2(X))\ddot{X} + (C_1(X, \dot{X}) + C_2(X, \dot{X}))\dot{X} + G_1(X) + G_2(X) = J^T (\tau - f_f - \tau_d) \quad (1.11)$$

where parameters with subscript 1 are for the platform and parameters with subscript 2 are for leg. This kind of dynamic formulation helps in controller design where the manipulator dynamics is calculated in task space and the leg dynamics is calculated in joint space for improved efficiency. An extended version of this formulation has been suggested by Khalil [161], [162]. In his method, the closed chain is divided into two subsystems, platform and legs. The dynamics of the platform is calculated as a function of the position, velocity and acceleration of the platform (x, \dot{x}, \ddot{x}) whereas the dynamics of the legs is calculated as a function of the joint position, velocity and acceleration $(q_i, \dot{q}_i, \ddot{q}_i)$. Then the actuator torque is calculated as a sum of the two, after projecting them into the active joint space. Hence the active joint torque is given by

$$\tau = J^T \left(F_p + \sum_{i=1}^6 J_{vi}^T J_i^T H_i \right) \quad (1.12)$$

where

- J is the platform Jacobian matrix
- F_p is total force and moment exerted on the platform
- J_{vi} is Jacobian matrix that relates the velocity of the origin of the platform frame to the Cartesian velocity transferred from each leg to the platform.
- J_i is kinematic Jacobian matrix of leg i and it gives the velocity transferred to the platform from each leg i .
- H_i is the dynamics of leg i

Another important dynamic formulation is that given in [113] and which is based on principle of virtual work and includes the actuator dynamics. The actuator torque is given as

$$\tau_i = \sum_{k=1}^6 \left\{ m_{ik} (X(t)) \ddot{q}_k (t) \right\} + C_i (X(t), \dot{X}(t), \dot{q}_i) + G_i (X(t)) \quad (1.13)$$

The parameters x , m and G are as defined above and the actuator friction is included in C . There are few assumptions used in the dynamic equations during controller design. These are:

Assumption 1: Manipulator inertia matrix is non singular

Assumption 2: Manipulator Jacobian matrix is nonsingular throughout the workspace

In addition to the above two assumptions, in the design of robust controllers, there are some

assumptions about the uncertainties.

Assumption 3: The uncertainties in the dynamic parameters are additive and can be expressed as nominal and deviation. In analyzing the uncertainty of the inertia matrix, the payload is assumed to be symmetrical and diagonal terms are mostly ignored. In some applications where the platform is subject to nonsymmetrical loads such as antennas, the uncertainty in off diagonal terms can also be significant [132]. The dynamic model including uncertainties is given as:

$$M_N(X)\ddot{X} + C_N(X, \dot{X})\dot{X} + G_N(X) = J^T(\tau - f_r - \tau_d) + d \quad (1.14)$$

where d is given by

$$d = \Delta M(X)\ddot{X} + \Delta C(X, \dot{X})\dot{X} + \Delta G(X) \quad (1.15)$$

1.1.3 Actuators used to drive legs

Various actuators have been used to drive Gough-Stewart platform depending up on the size, accuracy or precision required and speed of operation [73][80]. The most important ones are: hydraulic system, electric motor and piezoelectric systems.

1.1.3.1 Hydraulic Systems

Hydraulic actuators have high power-to-weight ratio and rapid response and they are used for applications such as flight and other motion simulators and telescopic antenna derives where the payload is very big. Their input output characteristic is not linear; they exhibit high nonlinearity. Moreover, hydraulic actuators resemble velocity source rather than force source and therefore their control is relatively complex than electric motors and their dynamics cannot be simply neglected rather it has to be modeled and used with platform dynamics if good motion tracking is to be obtained [7][29][71][67][101][173].

1.1.3.2 Electric motors

Various types of electric motors, including AC and DC servo motors and linear direct drive motors have been used to drive Gough-Stewart platform manipulators[52][58][89][94][113]. In case of AC and DC servo motors, the rotary motion of the electric motor will be converted to linear motion using gear assembly and lead screw. The interesting feature of electric

motors is their linear characteristics when they are driven using direct force control (current control) method. Mostly their dynamics can be neglected without having significant effect on controller performance.

1.1.3.3 Piezoelectric systems

Piezoelectric actuators are used for small sized micro manipulators used for robotic intervention [19], force sensing and pointing applications such as in optical communication [63]. Generally, piezoelectric actuators offer high bandwidth, good control linearity and high force output with small size. For example, a piezoelectric actuator stack of size 5mmx5mmx18mm can give 840N output force with deflection of 14 μ m and response time of 50 μ s [19].

1.1.4 Application areas

Stewart platform manipulator is generally applied in areas where high precision and/or high load carrying capacity is required. Some of these are

- Flight and other motion simulation[7][6][15][38][73][80][101][162]
- Medical robots/minimal invasive surgery [19][27][73][80][100][104][166]
- Precision machining[113][145][177][174]
- Force sensor[73][80][131]
- As pointing device[56][72]
- Telescopic Antenna Positioning[20][176][132][133]

1.1.5 Control of Stewart platform

To effectively utilize the structural advantages, namely high rigidity, stiffness and high precision, of Stewart platform manipulator to the above mentioned applications, a robust and high performance controller is necessary. A controller in Stewart platform manipulator has to generate torque signals which will be applied at the legs such that the moveable platform moves in a desired direction at a desired speed. The relationship between these force/torque which has to be given at the legs and the acceleration of the center of the platform is what is known as the forward dynamics and it is highly nonlinear and coupled. This makes the controller design very challenging. There are two approaches to the controller design

problem. The first one is to convert the desired task space position, velocity and acceleration of the platform center to desired joint leg lengths and close the loop by using measured leg lengths as feedback. This approach is known as joint space. In this approach, the individual leg measurements and desired values are taken separately and control is single input single output. In the second approach, the desired task space position, velocity and acceleration is not converted to desired leg length rather it is used directly by taking measured or estimated task space position, velocity and acceleration as feedback. Hence in this approach, control signal is calculated in task space and then it will be converted to joint space using the Jacobian matrix. The manipulator control system in this approach is therefore a multiple input multiple output system. Simplified block diagram of these two approaches is given in Fig.1.3 and Fig.1.4.

When a Stewart platform manipulator is used, it is expected to move without any vibration and to follow a given trajectory with minimum error and at high speed for all loading conditions. A controller for such an application should be able to handle the nonlinear dynamics and the various uncertainties such as parameter uncertainty, actuator friction and modeling errors. Furthermore, if numerical estimation is used to get feedback about end effector pose, the estimation error brings unmatched uncertainty. Therefore controller design and stability analysis for this system has to include all these uncertainties. The control method presently used in the industry is independent leg PID control [36][43]. This controller is not effective for high precision and high speed applications. In this method, measured leg lengths are used for feedback and information from corresponding legs only is used to generate control signals. This method results in less performance because of lack of synchronization and the inability of the linear PID controller to compensate the highly nonlinear coupled dynamics [58].

Few researches have been made to solve this problem and researches in the derivation of robust control strategies which make use of the in-parallel structure of the manipulator are highly needed. A state of the art review on the existing controllers will be presented in chapter 2.

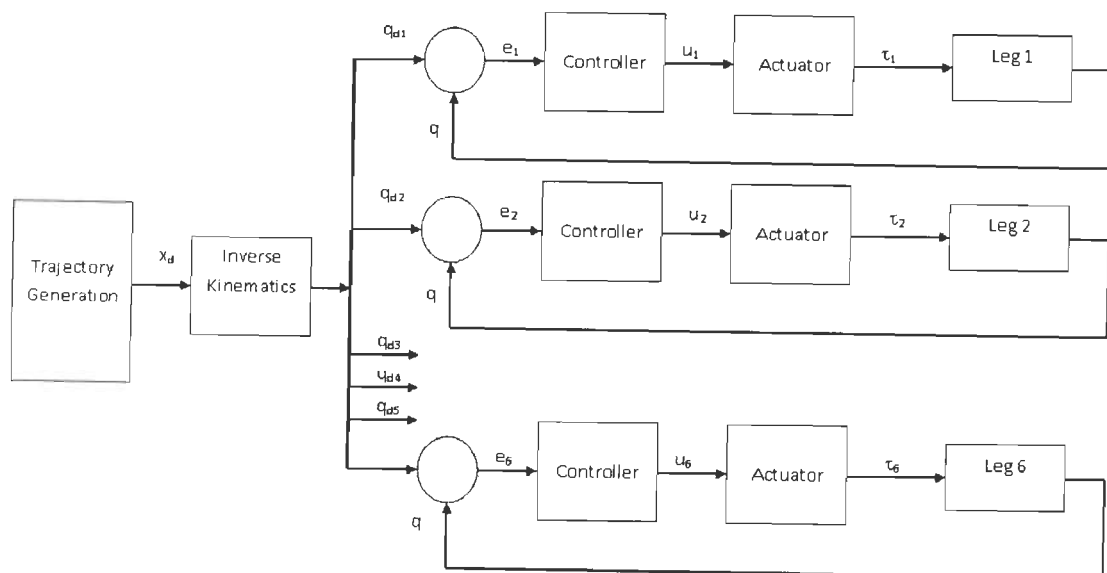


Fig. 1.3 Joint space trajectory tracking control approach for a Stewart platform manipulator

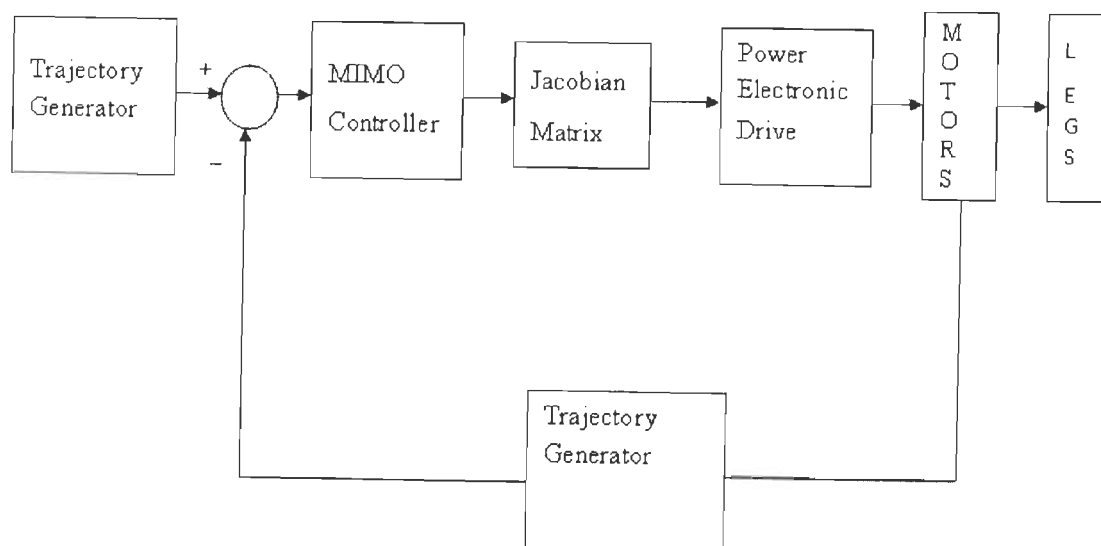


Fig. 1.4 Task space trajectory tracking control approach for a Stewart platform manipulator

1.2 Robust Control

Limitations of a popular methodology have always been among the factors stimulating new research. Such is the case with failure of previously famous control techniques such as LQR, state feedback and adaptive control when they are faced with uncertainties which were not considered in the design or modeling step [124][140]. Therefore it has now become a necessity to analyze uncertainties and design robust controllers. Robustness is a property

which guarantees that essential functions of the designed system are maintained under adverse conditions in which the model no longer accurately reflects reality [124]. Hence a robust controller is expected to maintain performance when the uncertainties are within a determined bounded limit or when they are from a known distribution. Though the idea of such controllers dates back to the time of classical feedback controllers, which use high gain feedback to suppress disturbances, present day challenges and the need for operation in nonlinear regions has resulted in a boost for the need for nonlinear robust controllers [129]. In this regard, a range of nonlinear robust control methods have been suggested in the last few decades, namely: nonlinear robust controllers based on computed torque method, Lyapunov based controllers, passivity based controllers, nonlinear H-infinity controllers and sliding mode controllers [67] [101][128][145][164].

In the case of nonlinear robust controllers based on computed torque control method first the nonlinear system is linearized using computed torque feedback control method and then linear robust control techniques are applied to control the resulting linear system. The method is very good but it highly depends on the linearizing computed torque feedback control part and it fails when there is a mismatch between the actual system and linearizing feedback parts which usually is the case [67][101]. The other two methods, Lyapunov method and passivity based method are very effective for systems having both structured and unstructured uncertainties [145][164]. However, both methods need the matching condition to be fulfilled [124]. Nonlinear H-infinity is an extension of the linear H-infinity case and is a very important method but it has an involved mathematics which makes its design and implementation difficult. Sliding mode control (SMC) is a robust controller design method, which is effective in controlling systems with significant uncertainties including parameter variations, unmodeled dynamics and external disturbances [15] [76] [142] [144] [157]. The design of SMC involves two steps: the design of stable surface or manifold known as sliding surface and the design of the control law, which is used to drive the system states towards the sliding surface and keep them on it. In the ideal case, once the system is driven towards this sliding surface, it is insensitive to all types of uncertainties [15][76] [157]. But, practical SMC has some drawbacks and these are: (i) chattering or high frequency oscillation of control signal, (ii) lack of robustness during the sliding phase, (iii) reduced life time of actuators and (iv) is unable to compensate unmatched uncertainties [144]. Various

modifications such as boundary control, saturation control, higher order sliding mode and integral sliding mode have been proposed to solve the above problems [30][42][55][66]. However, there still exist some problems which need to be solved. In this thesis, soft computing techniques are used to address all of these problems and enhance the performance of basic sliding mode controller.

1.3 Soft computing techniques

Soft computing is the fusion or combination of fuzzy, neural and evolutionary computing. It is meant for construction of new generation of artificial intelligence and for solving nonlinear and mathematically unmodeled systems at a lower cost [144][167]. The potential of soft computing techniques in solving existing and future challenges has been emphasized by various researchers including L. A. Zadeh, the pioneer of the subject. In his mail broadcast in May 19, 2009, L. Zadeh stated, “As we move further into the age of intelligent systems, the problems that we are faced with become more complex and harder to solve. To address the problems, we have an array of methodologies – principally fuzzy logic, neurocomputing, evolutionary computing and probabilistic computing. In large measure, the methodologies are complementary; and yet, there is an element of competition among them. In this setting, what makes sense is formation of a coalition. It is this perception that motivated the genesis of soft computing—a coalition of fuzzy logic, neurocomputing, evolutionary computing, probabilistic computing and other methodologies”[167].

As pointed by L. A. Zadeh, the individual methods of neural, fuzzy and evolutionary computing methods have some advantages and also disadvantages which can be minimized by using one or more of the other method. Neural networks have flexible learning capabilities and it is possible to develop nonlinear model using only input-output data. However it is difficult to fine tune the parameters to improve the modeling accuracy. On the other hand fuzzy logic gives a clear advantage in decision making and representation of expert knowledge. But it has poor adaptation capabilities. Combining the advantages of these two gives a more important intelligent system and adaptive neuro fuzzy inference systems (ANFIS) is one of the possible hybridization methods. Similarly evolutionary computation methods such as genetic algorithm, particle swarm computing and so on are useful for optimization of large scale and complex systems. The complexity of the parameter search or fine tuning of neural network and fuzzy logic systems can be solved by these evolutionary

computation methods. These kinds of hybridization or fusion have made soft computing to be a best candidate for solving many practical problems which cannot be solved by traditional or hard computing methods. The various fields of applications of soft computing are aerospace, communications, consumer appliances, electric power systems, manufacturing automation and robotics, power electronics and motion control, process engineering and transportation and so on. In this thesis, soft computing techniques are used to design robust controller for robotic system which is one of the potential application areas.

The most important property of soft computing techniques which is to be exploited in robust control systems is tolerance to uncertainty. Unlike traditional computing which is based on exact concepts, soft computing techniques are based on non exact concepts. This is similar to computation mechanism of human brain and due to this underlining nature soft computing techniques have the ability to deal with uncertainties. Hence it is very important to study the uncertainty tolerance property of soft computing techniques and utilize it for the design of robust controllers for complex nonlinear systems. This is the main goal of this thesis.

1.4 Objectives of the research

The main objectives of this thesis work can be summarized as

- To study previous works on application of soft computing techniques for robust control of nonlinear systems and extend them for the Stewart platform manipulator control.
- To design cost effective, practically realizable and high performance robust controller for Stewart platform to enhance its potential application in various fields.
- To study classical sliding mode control and its recent developments and solve some of still existing problems by using soft computing techniques.
- To study specific control problems of Stewart platform manipulator and design robust and high performance controller by using soft computing techniques.
- To study robust control problems of nonlinear systems and find new ways of soft computing techniques for better control results.

1.5 Motivation

Stewart platform manipulator was first proposed by D. Stewart for flight simulation and today its application has crossed multiple disciplines like robotic surgery, precision manufacturing, large telescopic antenna positioning and so on. In robotic surgery applications Stewart platform is used to achieve force feedback by giving six dimensional forces sensing capability. This is very important to achieve tactile sensing in remote surgery. It was also shown that force feedback has a promising potential in compensating physiological motions such as heart beat and motion due to breathing and improving surgical accuracy [177]. In precision manufacturing applications, Stewart platform is used for optical communications, laser weapons pointing, remote sensing and micro manufacturing.

In addition to these specific applications, the manipulator is also used as a test bed for nonlinear controllers. The highly coupled and nonlinear nature of the dynamic equation makes the manipulator control problem to be a challenging task. The manipulator control problem can be formulated as either SISO or MIMO depending up on whether the control is in joint space or task space. Moreover, the controller design is a multidisciplinary problem and needs understanding of the mechanical system, electrical sensing and actuation methods of the manipulator and the electronic system components required in addition to the aspects of control system. Therefore, the research work has multiple contributions to the field of controls system as well as to the specific applications listed above. The manipulator considered in the simulation works contained in the thesis is a large one with mass of legs being $1/4^{\text{th}}$ of the mass of the platform and hence non negligible. This makes the control problem nontrivial and important.

1.6 Statement of the problem

Given the mechanical structure of a 6DOF Stewart platform manipulator and the limits of variation of the payload, the statement of problem of the thesis is to

- Specify the limits of uncertainty of the dynamic parameters for a given high speed trajectory

- Study various soft computing techniques and use them to design various types of joint space and task space robust, low cost and easier controllers to drive the manipulator in a given trajectory
- Compare the performance of the various robust controllers in terms of their robustness, ease of implementation and cost

The organization of the thesis is as follows.

- Chapter 1 gives a brief introduction of the research work, motivation, and problem formulation.
- Chapter 2 gives a state of the art literature review on sliding mode control, on Stewart platform manipulator modeling and control and on application of soft computing techniques for robust control.
- Chapter 3 presents a detail analysis of the forward kinematics problem of the manipulator. The chapter deals with exhaustive comparison between the performance of two estimation methods, Newton Raphson numerical method and neural network. The performance metrics used for the comparison are: estimation error and average time taken. The methods are compared using various trajectories and simulation result shows that, the numerical algorithm, irrespective of initial conditions taken, always performs less than neural network for all trajectories. Moreover, the numerical algorithm takes longer average time while neural network takes less average time with uniform estimation error for all trajectories.
- Chapter 4 discusses the design and stability analysis of a fuzzy logic system based model less adaptive controller for trajectory tracking control of Stewart platform manipulator. The basic controller structure is SISO PID type of controller and the three gain parameters are tuned using fuzzy logic system. The input to the fuzzy logic is a weighted sum of errors in the leg lengths and output is a signal used to tune the gains. The control structure helps to achieve nonlinear PID and due to the weighted sum of errors synchronization error is minimized. Simulation results show that the controller has a good adaptive performance and achieves a better tracking accuracy than simple PID controller.

- Chapter 5 of the thesis deals with the design and stability analysis of three types of sliding mode controllers. These are fuzzy sliding mode controller implemented in task space, fuzzy sliding mode controller with integral loop (FSMCPI) and hybrid sliding mode with synchronization error. The task space fuzzy sliding mode controller has better performance than joint space PID, but it has some chattering which is the usual problem of sliding mode controllers. The fuzzy sliding mode controller enhanced with external PI loop shows better performance than both independent PID controller and fuzzy sliding mode controller. The hybrid sliding mode controller uses a new sliding surface which helps to drive synchronization error to zero and the controller achieves high performance in task space without using the complex forward kinematics. This has great advantage since it avoids the forward kinematics problem which is seen as a hindrance to controller implementation in task space.
- Chapter 6 deals with a further improvement to the sliding mode controllers using genetic algorithm. The chapter deals with integral sliding mode control (ISMC) and its design problems. ISMC is an improvement to basic sliding mode which uses sliding surface having an integral term. But its design is not a simple task and has no formal methods, especially for nonlinear systems. Hence, in this chapter the problem is formulated as an optimization problem and genetic algorithm is used. The application of the method to SISO and MIMO systems is discussed using examples. Finally how to apply multi-objective optimization using genetic algorithm for design of integral sliding mode controller for Gough-Stewart platform manipulator is discussed. The results show that the method can be effectively for the design of ISMC.
- Chapter 7 discusses a more robust and better controller, which is made from hybrid of fuzzy and neural networks. The controller is named as neuro fuzzy sliding mode controller and has two parts, viz., fuzzy logic system and neural network. They are used concurrently but each part is responsible for one phase of sliding mode controller. The fuzzy logic system is utilized to control reaching phase dynamics and the feedforward neural network is

employed to keep the system states on the sliding surface. The neural network is trained online using modified back propagation algorithm. Initially, fuzzy logic system is dominant and as the system moves from reaching phase to sliding phase, neural network becomes more active and hence a hybrid computing paradigm is achieved. The stability of the system is analyzed using Lyapunov's direct method. The proposed controller is implemented to regulate a second order nonlinear uncertain system and simulation results confirm that the proposed system reduces chattering and improves transient response. The results for an inverted pendulum system and its application for Gough-Stewart platform manipulator have confirmed the significant improvements obtained from the controller.

- Chapter 8 presents the summary of the contributions made in the thesis and the future scope of the work.

Chapter 2

LITERATURE REVIEW

1 Introduction

In this chapter we will establish a background for the research work by presenting a state of the art review of some of the related researches. In the first section, one of the important nonlinear robust control methods, sliding mode controller, is discussed. In addition to the basic or classical structure, proposed modifications and recent advances in solving drawbacks of the classical sliding mode controller are included. In the second section, an extensive review of research papers on design and development of controller for Stewart platform manipulator control is documented. The dynamic models used, the assumptions made and their advantages and disadvantages are discussed. The third section of the chapter covers review of soft computing techniques and their application for robust control. Neural network, fuzzy logic and genetic algorithms and their hybrid implementations are presented. At the end of the chapter the research gaps identified are listed and the foundation for the research is laid.

2.1. Sliding mode controllers

2.1.1. Basic sliding mode controllers

Sliding mode controllers are robust controllers effective for systems with significant uncertainties including parameter variations, unmodelled dynamics and external disturbances [144][14][142][76][157]. Since 1950, they had been applied for control of various types of systems including linear and nonlinear ones [142][76]. The problem definition of sliding mode controller can be stated as follows [87]. Given a nonlinear uncertain system

$$\dot{x} = f(x, t) + g(x, t)u + d(x, t) \quad (2.1)$$

where $x \in \mathbb{R}^n$, $f(x, t)$ is $n \times 1$ vector valued nonlinear function, $g(x, t)$ is $n \times m$ matrix valued nonlinear function, u is $m \times 1$ control input and $d(x, t)$ is the uncertainty,

- Find m sliding surfaces represented in vector form as

$$s(x) = \Lambda x \quad (2.2)$$

- Find a control law u of the form in (2.3) that drive the system states, from any initial point, towards a surface or manifold (2.2) in a finite time.

$$u = K \operatorname{sgn}(s) \quad (2.3)$$

Where x is the state vector of the system, Λ is $m \times n$ dimensional constant gain matrix, K is the gain of the discontinuous control signal which is responsible to drive the system states towards the sliding surface and $\operatorname{sgn}(\cdot)$ is the switching function given by,

$$\operatorname{sgn}(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \end{cases} \quad (2.4)$$

The design of the controller involves two steps, the design of stable surface or manifold known as sliding surface and the design of the control law which is used to drive the system states towards the sliding surface [14][157]. The stable sliding surface determines the performance of the system during sliding and hence is a very important factor [142][76]. It is determined based on required specifications and system constraints. The survey paper of [76] gives a detail discussion on the design of the sliding surface, characterization of the sliding and reaching modes and the design of the control law for linear systems. For linear systems design methods such as linear quadratic regulator, pole placement and Lyapunov method have been used to obtain the stable sliding surface [78][81][153]. All of these methods give a single sliding surface, which results in a single closed loop damping ratio of a system to be controlled.

The above discussed conventional sliding mode controllers have two phases: (i) reaching phase, where states of a controlled system are driven towards a stable sliding surface and (ii) sliding phase, where the system states are constrained on the sliding surface and the system is insensitive to uncertainties [157]. Such conventional sliding mode controllers have some basic drawbacks, which motivated various researches on sliding mode controllers. The basic drawbacks are: (i) high frequency oscillation of the sliding variable near the sliding

surface, which is known as chattering, (ii) lack of robustness during the reaching phase and (iii) lack of robustness to unmatched uncertainties [144]. In the following sub-sections, we will review some of the recent and important researches proposed to solve these problems.

2.1.2. Boundary layer and saturation control

One of the serious drawbacks of sliding mode controllers is chattering. This is high frequency oscillation of the control signal which excites high frequency dynamics of a controlled system and may result in instability. Because of this, most of the research in sliding mode control has been aimed at solving the chattering problem. The solutions proposed differ on the assumption they made on the source of the chattering. Some of these assumptions are: the discontinuity of the sgn function, presence of fast actuators and sensors, the existence of time delay and hysteresis and so on [47][48][65][66]. The boundary layer control method is proposed based on the assumption that the source of chattering is existence of delay and hysteresis. Hence in boundary layer control, the controller (2.3) is modified to the form

$$u = K \text{sat}(s) \tag{2.5}$$

where

$$u = \begin{cases} +1 & \text{for } s \geq \psi \\ s & \text{for } -\varphi < s < \psi \\ -1 & \text{for } s \leq -\psi \end{cases} \tag{2.6}$$

Ψ is a small constant which determines the width of the boundary layer [47][48][65][66][76]. The objective of this kind of controller is to replace the relay like switching by a high slope smooth function and reduce the chattering. In state space, such controller appears to have a boundary around the sliding surface which is having width of Ψ . The problem of this controller is once the system states are driven to the boundary layer, the dynamic of the states is not known [47][48].

2.1.3. Second order sliding mode controllers

The use of high slope smoothing devices described in section above was unable to solve the chattering problem and further research resulted in higher order sliding mode controllers. The main objective of higher order sliding mode controllers, where second order is the most famous one, is to influence higher order derivatives of the sliding surface. In conventional

sliding mode controllers, the first derivative of the sliding surface is driven or influenced by the control signal. However, in higher order sliding mode controllers, the control signal acts on the sliding variable and some of its higher order derivatives [47] [65][66]. But such controllers have two limitations. The first one is actuator saturation. In higher order sliding modes, the control signal which act on the first order derivative of the sliding surface is obtained by integrating the discontinuous control [47][48]. This integration may result in actuator saturation. The other problem of higher order sliding modes is they need systems having relative degree greater than one. Relative degree of a system is the number of times the output has to be differentiated before dependence on control signal is reached. This property limits the applicability of higher order sliding mode controllers.

2.1.4. Integral sliding mode controllers

As described above classical sliding mode controllers have two phases, reaching phase and sliding phase. In the reaching phase, the controller is not insensitive to disturbances and it is not robust. To solve this problem and get additional benefits integral sliding mode controllers have been proposed. Integral sliding mode control (ISMC) is an improvement to conventional SMC and uses a nonlinear sliding surface having an integral term [144] [55], [81] [156]. It is able to remove reaching phase problem of conventional sliding mode by using a sliding surface which is designed to constrain the system states to be on sliding mode from initial time [55], [81]. Moreover, the sliding surface of ISMC improves the stability of sliding dynamics and it enables to enhance robustness against unmatched uncertainties [153].

2.2. Stewart Platform manipulator control

A brief introduction of mechanical structure, modeling and control of Stewart platform manipulator has been given in the introductory section. In this section, latest researches in control and modeling of the manipulator are presented. For an extensive review of kinematics and dynamics modeling of the manipulator, the reader is referred to a review paper by [15]. In that review paper, control aspects were not discussed stating that control research on Stewart platform is fresh and untouched. In the last decade, after the review paper, many research results have been published and a systematic study of the relationship between the researches and the gaps there is presented as follows.

2.2.1. Kinematic modeling

The research on kinematics of Stewart platform manipulator has been mainly on its forward kinematics. This is so because it is very important in the control of the manipulator. In general there are three basic approaches to the forward kinematics problem: geometric, analytic and numerical methods [16][15][16-18][60][82][85][91][103][114][120][117][147][176][177][178]. Some other approaches include use of rotary and vision sensors [15], [64][79][105] nonlinear observer based approaches [99] and neural network estimations [60][91][178]. A recently proposed method is an estimation method using clustering [103]. The forward kinematics is required in feedback control when controller is implemented in task space and it has to be computed or measured at servo rate to get the 6DOF platform position and orientation. There are two problems in this; one is the accuracy of the measurement or estimation and the other is cost of sensors if measurement is to be used or speed in calculation if numerical methods are to be used. The problem with analytical methods is they give multiple solutions (up to 40) or all possible solutions and the exact solution has to be selected [15][73][80][85][114][117][120][147]. This makes the implementation complex and impractical for real time control. The numerical methods are fast to implement but have problems like: reliability, may lead to wrong result, take long time depending on initial guess. The problem with sensor based methods is high cost [64][73][80][79][82], while that of neural networks is reliability. Cascade implementation of neural networks and numerical methods with the objective of improving reliability and estimation accuracy is another proposed method and it was reported that it can result in better solutions [16][32].

Other important kinematics parameters necessary for control and modeling are the joint space and task space velocities. The velocity kinematics is given as

$$\dot{q}=J\dot{X} \tag{2.7}$$

Where J is the Jacobian matrix of the platform given by [16], [159][160]

$$J_p^{-1} = \begin{bmatrix} a_1 & (R.P_1 + r)X a_1 \\ a_2 & (R.P_2 + r)X a_2 \\ \dots & \dots \\ a_6 & (R.P_6 + r)X a_6 \end{bmatrix} \quad (2.8)$$

and

$$a_i = \frac{B_i P_i}{\|B_i P_i\|} \quad (2.9)$$

is unit vector in the direction of leg i . In feedback control applications, it is very important to use high quality velocity signal in order to decrease the effect of measurement disturbances. In transforming task space desired velocities to leg velocity, it is better to differentiate the leg displacement instead of using the above relation (2.7). But in task space control, it is better to differentiate the measured leg lengths and transform it using inverse the Jacobian matrix [177]. This is so because if the task space positions are obtained using numerical methods, the differentiation will amplify the measurement disturbance in the leg length and affecting the stability and performance of control loop.

2.2.2. Dynamic modeling

Dynamic modeling of Stewart platform manipulator can be done using the Lagrangian method or recursive Newton Euler method [15][62]. The Lagrangian formulation gives a closed form equation, which is useful for the design of model based controllers, and it has been used by most of the authors. The dynamic equation of manipulator can be given either in task space or joint space, each having certain advantages. The joint space approach is easier from closed loop control point of view as it does not need the use of forward kinematics in feedback loop. But dynamic modeling is complex in joint space because of a nonlinear coordinate transformation needed [177]. On the other hand task space approach is easier for dynamic modeling but it needs the highly nonlinear forward kinematics for feedback controller implementation. Using joint space approach, the dynamic equation is given by

$$A(q)\ddot{q} + B(q, \dot{q})\dot{q} + Q(q) = \tau - f_r - \tau_d \quad (2.10)$$

Where q is the length A is 6x6 manipulator inertia matrix, B is also 6x6 coriolis and centrifugal torque/force, Q is 6x1 gravitational torque/force, τ is the actuator torque, f_f is torque due to friction and τ_d is the disturbance

In the task space, it is given by

$$M(X)\ddot{X} + C(X, \dot{X})\dot{X} + G(X) = J^T (\tau - f_f - \tau_d) \quad (2.11)$$

Where X is the 6DOF Cartesian position and orientation of the platform given by $X=[x, y, z, \alpha, \beta, \gamma]$, M is 6x6 manipulator inertia matrix, C is also 6x6 coriolis and centrifugal torque/force, G is 6x1 gravitational torque/force, J is 6x6 manipulator Jacobian, τ is the actuator torque, f_f is torque due to friction and τ_d is the disturbance.

The above equations of motion have certain properties which are very important for model based controller design. These are [36], [43], [58], [67], [177]:

- 1) The number of second order differential equations is equal to the number of DOF of the robot
- 2) All elements are defined explicitly
- 3) The domain of definition of the generalized coordinates X and q is a compact set which is the subset of the workspace.
- 4) The inertia matrix is positive definite, bounded and non-singular. It can be also assumed to be symmetrical in special cases where the loading is symmetrical.
- 5) The Jacobian matrix is usually assumed to be nonsingular

Alternatively, writing platform dynamics and leg dynamics separately, the dynamic equation can be rewritten as:

$$(M_1(X) + M_2(X))\ddot{X} + (V_1(X, \dot{X}) + V_2(X, \dot{X}))\dot{X} + G_1(X) + G_2(X) = J^T (\tau - f_f - \tau_d) \quad (2.12)$$

where parameters with subscript 1 are for platform and parameters with subscript 2 are for leg. This kind of dynamic formulation helps in controller design where the manipulator dynamics is calculated in task space and the leg dynamics is calculated in joint space for improved efficiency. An extended version of this formulation has been suggested by Khalil [161-162]. In the method by [162], the closed chain is divided into two subsystems, platform and legs. The dynamics of the platform is calculated as a function of the position, velocity and acceleration of the platform (x, \dot{x}, \ddot{x}) whereas the dynamics of the legs is calculated as a

function of the joint position, velocity and acceleration $(q_i, \dot{q}_i, \ddot{q}_i)$. Then the actuator torque is calculated as a sum of the two, after projecting them into the active joint space. Hence the active joint torque is given by

$$\tau = J_p^T \left(F_p + \sum_{i=1}^6 J_{v_i}^T J_i^T H_i \right) \quad (2.13)$$

Where

J_p is the platform Jacobian matrix

F_p is total force and moment exerted on the platform

J_{v_i} is Jacobian matrix that relates the velocity of the origin of the platform frame to the Cartesian velocity transferred from each leg to the platform.

J_i is kinematic Jacobian matrix of leg i and it gives the velocity transferred to the platform from each leg i .

H_i is the dynamics of leg i

In the above formulation, the platform dynamics and the leg dynamics can be calculated separately using either Newton Rapson or Lagrangian method. The parameters in this model do not fulfill the properties mentioned for Lagrangian form. In [44], this model is used for task space computed torque control. Another important dynamic formulation is that given in [113] and which is based on principle of virtual work and includes the actuator dynamics. The actuator torque is given as

$$\tau_i = \sum_{k=1}^6 \{ m_{ik} (x(t)) \ddot{q}_k (t) \} + C_i (x(t), \dot{x}(t), \dot{q}_i) + G_i (x(t)) \quad (2.14)$$

The parameters X , m and G are as defined above and the actuator friction is included in C . There are few assumptions used in the dynamic equations during controller design. These are:

Assumption 1: Manipulator inertia matrix is non singular

Assumption 2: Manipulator Jacobian matrix is nonsingular throughout the workspace

In addition to the above two assumptions, in the design of robust controllers, there are some assumptions about the uncertainties.

Assumption 3: The uncertainties in the dynamic parameters are additive and can be expressed as nominal and deviation. In the uncertainty of the inertia matrix, the payload is

assumed to be symmetrical and diagonal terms are mostly ignored. In some applications where the platform is subject to nonsymmetrical loads such as antennas, the uncertainty in off diagonal terms can also be significant [133]. The dynamic model including uncertainties is given as:

$$M_N(\mathbf{X})\ddot{\mathbf{X}} + C_N(\mathbf{X}, \dot{\mathbf{X}})\dot{\mathbf{X}} + G_N(\mathbf{X}) = \mathbf{J}^T(\boldsymbol{\tau} - \mathbf{f}_r - \boldsymbol{\tau}_d) + \mathbf{d} \quad (2.15)$$

Where \mathbf{d} is given by

$$\mathbf{d} = \Delta M(\mathbf{X})\ddot{\mathbf{X}} + \Delta V(\mathbf{X}, \dot{\mathbf{X}})\dot{\mathbf{X}} + \Delta G(\mathbf{X}) \quad (2.16)$$

2.2.3. Joint space controllers

In the joint space approach, the controller is a collection of SISO systems implemented using local information on each actuator length only and the coupling between legs is ignored or is considered as a disturbance. The advantage of this approach is that local information required for feedback is obtained easily using simple sensors. Hence the control algorithm can be executed as fast as possible and the individual SISO controllers can be easily implemented in parallel. In this mode, there are various classical and modern controllers proposed and still some authors argue that this is a better method to follow. An early paper by [119] argue that that model based robust controllers need exact models which is not achievable or is costly and proposes modification to the existing linear controllers. The authors proposed a P-P-PI controller which utilizes acceleration feedback. They have shown that their method allows the setting of desired disturbance rejection factor and recovery time. A state-variable filter is utilized for reconstructing acceleration measurements. The authors applied their controller to a delta parallel robot and have shown the validity of the controller experimentally. The speed attained was 2m/sec and maximum acceleration was 88m/sec². Another early work is that of [113] which is a sliding mode controller having an observer for disturbance estimation. The controller is used to drive a manipulator which is used for motion simulation where high payload and low speed is desired and it is based on a dynamic model whose parameters are calculated in task space but the switching control is performed based on joint space sliding variable. Stability analysis for the disturbance estimator and the sliding mode controller is given and the controller is verified experimentally. Due to the low speed requirement, the controller has shown good results. In [43] is suggested a joint space controller of PD with gravity compensation but their controller is not applied for Gough-Stewart platform.

In [177], a high precision joint space controller is discussed. The authors argued that high precision can be obtained by using high quality differential signal for PID controllers and hence their controller uses a nonlinear tracking differentiator (TD) to yield a high quality differential signal in the presence of disturbances and measurement noise in the feedforward path, and an extended states observer (ESO) to provide the system's state and the real action component of the unknown disturbances including nonlinear friction in the feedback path. The nonlinear PD (N-PD) scheme is used to synthesize the control action. As given in [43] model less controllers lack guarantee of stability and performance and in [177] also no stability analysis is given. Especially the stability of the state observer is very critical. The authors have given experimental results for high speed and low speed applications. In [172] also, it was argued that high precision control of parallel manipulator can be obtained by using linear joint space controllers if synchronization error is considered. In joint space control, individual controllers are using local information only and this has two important implications. The first one is, if a disturbance occurs in one of the legs, only the controller of that leg is sensing it and trying to reject the disturbance while the disturbance may affect the whole system. The other point is since the legs are arranged in parallel lack of synchronization may result in large unwanted interaction forces. This has been reported by many authors [36][44][172]. Hence [172] proposes a solution to the synchronization problem by taking differential position error amongst actuators as feedback and using a saturated PI controller. The saturated PI controller is integrated to a usual PD controller which uses position error. The authors have given stability analysis of the controller and also shown the application of the controller for a 3DOF parallel manipulator.

A slightly different type of controller from all the above is the one proposed in [7]. The controller is a model based inverse dynamics controller and the authors have given the design and stability analysis of the controller and also simulation results are shown. In [36] also an adaptive controller which makes use of synchronization error is designed. The controller is based on a joint space dynamic model and feedback is used to stabilize the system. Stability proof and simulation results are given as evidence for the controller. But the above two controllers, [7][36] seem very unrealistic since dynamic modeling in joint space is very difficult. A more recent joint space control effort is given by [6]. The authors have given a joint space PD controller but the parameters of the controller are tuned by a genetic

algorithm. The genetic algorithm determines the PD parameters using dynamic model of the manipulator. Simulation results are given. In this thesis, we present several proposed nonlinear controllers implemented in joint space to improve the drawbacks of some of the above controllers. One of these is sliding mode controller which considers synchronization error. The sliding mode controller uses a new sliding surface which is designed to drive both tracking and synchronization errors to zero. The simulation results obtained are promising. Another controller proposed is joint space integral sliding mode controller designed using genetic algorithm [30].

Though authors of [36][177][172] and others proposed and tried to show that joint space controllers can achieve high performance at high speeds, in [7][52] it was stated that that simple single-joint controllers cause superproportional errors at velocities over 0.4 m/sec which is simply intolerable, since Stewart platform manipulators are supposed to be advantageous in the range of high dynamics. To solve this contradiction, some authors have proposed using model based task space controllers.

2.2.4. Task space controllers

Another controller implementation paradigm is task space or using workspace variable. This method is supposed to have the ability to solve the drawbacks of joint space control and achieve superior performance than joint space control in the presence of various uncertainties [58]. Most model based robust controllers are designed in task space due to the relative ease in computing the dynamic model parameters. An earlier attempt of task space control for 6DOF parallel robot is that of [71]. The controller is a fuzzy sliding mode controller with two inputs, sliding parameter s and its derivative, and one control output. The controller is designed using dynamics of the actuator and the dynamics of the manipulator is not considered. An experimental result is given for DELTA robot. Then in [89], a Cartesian space tracking controller is designed using the passivity principle. In the controller, desired force is calculated in task space from position control and then using the desired force, necessary torque is then computed at joints using force convergent principle. The controller is given as

$$\mathbf{u} = \alpha_3^{-1} \mathbf{F}_d - \alpha_3^{-1} (\mathbf{I}_d + \alpha_2 \mathbf{J}^T) \mathbf{F} - \alpha_3^{-1} \alpha_1 \mathbf{J}^{-1} \dot{\mathbf{X}} \quad (2.17)$$

where F is measured force at the motor input using motor currents and F_d is calculated from dynamic model as

$$F_d = A(x)\ddot{x}_r + B(x, \dot{x})\dot{x}_r + G(x) - Ks \quad (2.18)$$

Where x_r and s are variables given by

$$s = \Lambda e + \dot{e} \quad (2.19)$$

$$x_r = \ddot{x} - \Lambda \dot{e} \quad (2.20)$$

The controller is tested using simulation and also experimental results are given. But the trajectory followed is very slow. The exponential decay of the task space tracking error and stability of the system has also been given by the authors. At about the same time, [177] presented an adaptive task space controller together with a stability proof of the convergence of tracking error and the parameter estimation. The controller is verified using simulation and experimental results and practical application of the same controller for a CNC machine is given in [174]. Later [67], [101] and [145] developed more elaborate task space controllers fully based on dynamic model. In [67] and [101] back stepping method is used to design nonlinear position tracking controller for a hydraulic actuator driving Stewart platform manipulator. The controller is similar in structure to that of [89] except the type of actuator used. It has adaptation mechanism to cope with uncertainties and uses two sliding type observers to avoid acceleration measurement.

The controller is experimentally verified and global asymptotic stability proof is also given. In the controller, Newton Rapson numerical method is used for forward kinematics. In [145] a robust controller is proposed by combining a computed torque controller based on approximate dynamics and h-infinity controller. The h-infinity controller is used to compensate for the model uncertainty due to the approximate dynamics and is implemented in joint space. The authors have given experimental results as a proof of the superior performance of the controller over PID for slow and fast trajectory but have not given stability proof of the combined controller.

In all the above task space controllers, it was assumed that dynamic parameters of the manipulator are calculated in task space. Especially the position dependent mass matrix is very critical and it takes huge computation time and is one of the main problems of model based controllers. To solve this problem, [52] suggested a method that simplify and speed up

the computation of mass matrix for any parallel robot. The main idea of the approach is that the inertia matrix of the whole manipulator has to be a combination of the mass matrices of the incorporated kinematic substrings. Based on this approach, for the case of Stewart platform, the mass/inertia matrix will be a combination of the platform and the legs. However, none of the practical experiments have used the method.

One of the most important features of model based controllers is availability of stability proof. In [58] a rigorous stability proof is given for a robust nonlinear task space controller. The controller given by the authors is designed using Lyapunov redesign method and is similar in structure to that of [89] and [101] but differs in few important points. The first difference is that in [58] actuator dynamics is not considered which is reasonable as electric motors are used. The other difference is in [89] and [101] a simple sliding variable is used as corrective signal but in [58] a switching kind of controller is used. Moreover, the uncertainty analysis given in [58] is more rigorous and realistic as it includes uncertainty due to measurement or estimation of the task space position and angles. The authors used numerical estimation for the forward kinematics but used a filter to obtain smooth outputs. They also incorporated friction estimation to make the controller more efficient. The authors have given experimental verification for the performance of their controller.

A further development in task space control of the manipulator is given in [17] and [28]. In [131], an inverse dynamics based robust controller using approximate dynamics was presented. Similar to [145], part of the controller, leg dynamics compensation, is obtained in joint space and these two controllers can be called hybrid as they mix the two spaces. Stability analysis of these controllers is given but uncertainty due to friction and measurement errors were not considered. The controllers given in [44] and [105] give a different impetus to task space control. The two controllers have a common ground that they both use vision sensor to close the feedback loop and hence avoid the need for forward kinematics estimation. However, in [44] a computed torque controller is given while in [105] a linear decoupled controller is proposed. In the later case decoupling is obtained using a modified Jacobian matrix and both controllers were implemented in simulations. Another important task space controller for hydraulic robot is proposed by [67]. The controller is a model based controller which includes actuator dynamics and is designed based on the inverse dynamics principle. The proposed controller is verified using simulations.

Sliding mode controllers are the more recent controllers which were reported in [132][133]. In [132][133] simple sliding mode controllers are used to control a Gough-Stewart platform manipulator which is used for antenna positioning. In their latter paper [133], the authors have given detail analysis of the dynamic uncertainties. In this thesis various types of sliding mode controllers are designed and applied for the control of Stewart platform manipulator. Analysis for friction and other uncertainties and stability proofs are also given. The simulation results obtained are promising. Especially the integral sliding mode controller solves the unmatched uncertainty problem of the forward kinematics estimation mentioned in [58]. However, for practical implementation, experimental verification of the controllers is required.

2.3. Soft computing techniques and sliding mode control

It is very well known that fuzzy logic systems are very good in decision making, while neural networks are good at function approximation. Therefore, combining neural and fuzzy logic systems with conventional sliding mode controllers can result in a more robust controller. For this kind of research, the studies of [102][24][98][45][54][50][62][60][54][57][74][77] [102][118][122] are worth mentioning. In most of these controllers a fuzzy logic system has been used to replace the switching function except in [98] where fuzzy logic system was used to generate nonlinear sliding surface. Similarly, in [144][12] neural network was employed to estimate an optimal sliding surface and in [148], a radial basis function network was utilized to vary the gain of the switching function of a sliding mode controller. In all the above studies, either neural network or fuzzy logic system was used to improve the performance of basic sliding mode controller. But none of the implementations could solve the chattering problem completely. This is because using fuzzy logic to replace switching function results in a similar effect to that of saturation control. Therefore, a more important development could be the use of two artificial intelligence techniques together.

In an attempt to use the said two artificial intelligence techniques with sliding mode controller, neuro-fuzzy sliding mode controllers had been proposed [2][3][122]. In [122], a neuro-fuzzy system was used to approximate equivalent control part of conventional sliding mode controller but the switching function part was the same with the conventional one. In

the work of [3] the controller had two parts: a neural network part which was employed to estimate the equivalent control signal and a fuzzy logic system, which was utilized as switching function. The output of the fuzzy logic is used to train the neural network. The work reported by [2] uses an ANFIS network to replace the switching function which resulted into adjustable boundary layer but in the same paper, it was reported that the system was sensitive to parameter variation and external disturbance. These latest approaches have shown better performances and are promising in solving the drawback of classical sliding mode control. Hence one part of the thesis deals with a new proposal on neuro fuzzy sliding mode control.

2.4. Gaps identified

From the above literature review, it is clear that there are certain gaps which have to be filled by research outputs. Some of these gaps are listed as follows

- Though chattering has been a hot research topic in the history of sliding mode control, a final or lasting solution has not been achieved. Hence one important direction is to use the important properties of soft computing techniques to reduce chattering in nonlinear systems.
- Another gap is the restrictive matching condition. The insensitivity of sliding mode controller to external disturbance and modeling uncertainty needs matching condition and for unmatched uncertainty the robustness of SMC is not satisfactory. Integral sliding mode control is a good option but its application to complex systems like Stewart platform has to be investigated. Moreover, design approach which is easy and efficient in getting a reasonable sliding surface has to be investigated.
- Though various controllers have been tried for high performance control of Stewart platform manipulator, a practically implementable and real controller has not been achieved yet. Hence design of such robust and high performance controller which uses task space modeling is required in order to utilize the advantages of the parallel kinematics of the manipulator. Sliding mode controller is a best candidate for this.

While trying to find solutions to some of the drawbacks of sliding mode control, at the same time utilizing its merits for high performance control of Stewart platform manipulator is a promising option. In the thesis, these kinds of solutions are proposed and analyzed.

Chapter 3

COMPARISON OF NUMERICAL AND NEURAL NETWORK FORWARD KINEMATICS ESTIMATIONS

To effectively exploit the structural advantages of Stewart platform manipulator, a high performance controller is required and to do so, the forward kinematics of the manipulator has to be computed. But unlike serial manipulators, forward kinematics is not direct in case of Stewart platform manipulator; rather it is nonlinear and complicated. Methods proposed to solve it can be classified as analytical methods, numerical or other estimation methods and sensor based methods. The first method gives multiple solutions, up to 40, out of which the practical and exact solution has to be selected. The last method, sensor based method is to use extra sensors in addition to 6 leg measurements or to use vision sensor. The method is costly and is not much favored. In most practical cases or control algorithms, numerical or observer based methods are used. In this chapter, we will try to exhaustively compare the performance of two estimation methods, Newton Raphson numerical method and feed forward neural network. The performance metrics used for the comparison are: estimation error for position and orientation and average time taken. The methods are compared using various trajectories. The simulation result shows that, the numerical algorithm, irrespective of initial conditions taken, always performs less than neural network for trajectories where there is pitch motion. Moreover, the numerical algorithm takes longer average time while three-layer feed forward neural network takes less average time with uniform estimation error for all trajectories.

3.1. Forward kinematics of Stewart platform

The forward kinematics problem in robotic system is to find the end effector position and orientation with respect to the base for a given joint angle values of its links. In Stewart platform manipulator this is to find the actual position and orientation of the moveable platform with respect to the base for a given set of leg lengths. In serial robots, the computation of the forward kinematics is direct and linear. For a given set of joint angle values, the end effector position is unique. This is not the case in parallel robots in general. In Stewart platform manipulator, for example, there may be multiple orientations or positions for a given set of leg lengths. This is because of the parallel nature of the structure. The mathematical formulation of the forward kinematics of Stewart platform manipulators is highly nonlinear and its computation is complex. Due to this, some authors have proposed direct measurement of the end effector position and orientation using costly sensors or to use extra sensors to make the computation easier. In general because of its importance in task space control, the forward kinematics problem has been a central research issue in Stewart platform manipulator for more than two decades [6][15][16-18][99][60][64][74][82][85][91][103][114][120][117][147][176][175][178].

The solutions proposed to solve this problem can be classified into three main categories: closed form analytical [15][73][80][85][114][120][117][147], numerical or other estimation [6][15][99][60][73][80][91][103][176][175][178] and sensor based [64][73][80][74][82]. Whereas numerical solutions like Newton Raphson have problems of dependence on initial guess, analytical methods give multiple solutions and they are also computationally complex. The sensor based methods result in high implementation cost. The requirements for closed loop control application are to get the position and orientation of the platform fast and to get it with minimum error.

To solve the problems of numerical and analytical solutions, neural network based estimation method has been proposed [6][60][91][178]. The ground for the use of neural network to solve the forward kinematics problem lies in their nonlinear function estimation capacity. The network will be trained offline using input output data obtained from inverse kinematics and then it will be used for online estimation in a feedback loop. In using neural network, after network is trained, it has to be tested for generalization capacity. In [60] training performances were given but generalization is not tested using various trajectories.

In [99] the output of a feed forward neural network is used as the initial guess for the Newton Raphson method but Newton Raphson algorithm may take longer time to converge to a solution, whatever the initial guess is, when a root is at inflectional tangent; moreover, using neural network and Newton Raphson in cascade will increase the total estimation time.

In this chapter we study in detail the performance of a feed forward neural network with enough number of hidden layers and neurons. The main objective is to show that the performance is better for trajectories where the Newton Raphson method fails to give small error and to show the reliability of neural network estimation.

3.2. PROBLEM FORMULATION

As described in section above, the forward kinematics problem of Stewart platform manipulators is to find the position and orientation of the manipulator with respect to the base for a given set of leg lengths. For the problem formulation we briefly summarize the kinematic modeling. Fig.3.1. gives the frame assignment and geometrical description of a general 6 DOF Stewart platform manipulator with hexagonal base and platform. In the dynamic modeling, such hexagonal platform gives none symmetrical mass matrix. Hence in the dynamic modeling and simulation throughout the rest of the thesis, the platform is assumed to be circular and hence the mass and inertia matrices are taken symmetrical. If irregular hexagon shaped platform is used, the effect of the asymmetry is considered as disturbance. As shown in the figure, the manipulator consists of a base B_i ($i=1, 2, \dots, 6$) and platform P_i ($i=1, 2, 6$) joined by six extendable legs. Each leg is attached to the base by universal joint and to the platform by a spherical joint. The length of legs is controlled by an actuated prismatic joint. A reference frame F_b (O_b - X_b , Y_b , and Z_b) and a coordinate frame F_p (O_p - X_p , Y_p , Z_p) are attached to the base and the platform respectively. The position vector of the center of universal joints B_i in frame F_b is $b_i = [b_{ix} \ b_{iy} \ b_{iz}]$ and the position vector of the center of spherical joints P_i in frame F_p is $p_i = [p_{ix} \ p_{iy} \ p_{iz}]$. Let $r = [r_x \ r_y \ r_z]$ be the position of the origin O_p with respect to O_b and also let R denote the orientation of frame F_p with respect to F_b . Thus the Cartesian space position and orientation of the platform is specified by $X = [r_x, r_y, r_z, \alpha, \beta, \gamma]$ where the three angles α, β, γ are the yaw-pitch-roll rotation angles that constitute the transformation matrix R .

For Fig.3.1, a vector equation of the form

$$\overline{O_b B_i} + \overline{B_i P_i} = \overline{O_b O_p} + \overline{O_p P_i} \quad (3.1)$$

can be written where

$\overline{O_b B_i}$ is the vector b_i ,

$\overline{O_b O_p}$ is vector r and $\overline{O_p P_i}$ is vector p_i .

Substituting these symbols and writing all vectors in base frame,

$$\overline{B_i P_i} = R p_i + r - b_i \quad (3.2)$$

The magnitude of $\overline{B_i P_i}$ is the length of i^{th} leg. Hence the inverse kinematic equation, which gives the length of the legs l_i for a given platform position and orientation, is given by

$$l_i = \|\overline{R p_i + r - b_i}\| \quad (3.3)$$

The solution of (3.3) is unique for a given platform position r and orientation R and can be directly calculated. This constitutes the solution of the inverse kinematics problem. The forward kinematics problem is finding the actual Cartesian space position and orientation $X = [r_x, r_y, r_z, \alpha, \beta, \gamma]$ given a set of leg lengths, $l_1, l_2, l_3, \dots, l_6$. This is a nonlinear equation and it has no direct solution.

3.3. NUMERICAL ESTIMATION METHOD

Taking the three orientation angles as the standard yaw- pitch- roll angles, the transformation matrix R can be written as

$$R = R_z(\gamma) R_y(\beta) R_x(\alpha) \quad (3.4)$$

Let r_{ij} be the i^{th} row and j^{th} column of the transformation matrix, then multiplying and taking magnitude of the vector, (3.3) becomes

$$l_i^2 = (r_{11} p_{ix} + r_{12} p_{iy} + r_{13} p_{iz} + r_x - b_{ix})^2 + (r_{21} p_{ix} + r_{22} p_{iy} + r_{23} p_{iz} + r_y - b_{iy})^2 + (r_{31} p_{ix} + r_{32} p_{iy} + r_{33} p_{iz} + r_z - b_{iz})^2 \quad (3.5)$$

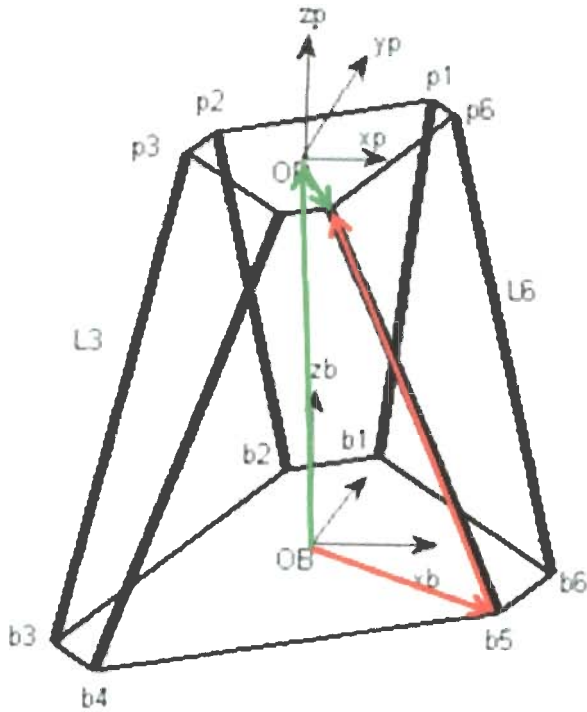


Fig.3.1 Generalized Stewart platform

In the numerical method, first an estimated Cartesian position (r_x, r_y, r_z) and orientation angles (α, β, γ) are taken. Then the corresponding leg lengths are calculated using inverse kinematics equation (3.5) and the error between the calculated leg lengths and the measured values is used to adjust the estimated values. The iteration continues until a tolerable error value is achieved. In the Newton Raphson method [16-18][99][173], the next estimate is calculated using

$$X(k+1) = X(k) + J^{-1}F(X(k)) \quad (3.6)$$

where

$$F(X(k)) = \begin{bmatrix} f_1(X(k)) \\ f_2(X(k)) \\ \vdots \\ f_6(X(k)) \end{bmatrix} \quad (3.7)$$

$$f_i(X(k)) = l_i^2 - ((r_{11}p_{ix} + r_{12}p_{iy} + r_{13}p_{iz} + r_x - b_{ix})^2 + (r_{21}p_{ix} + r_{22}p_{iy} + r_{23}p_{iz} + r_y - b_{iy})^2 + (r_{31}p_{ix} + r_{32}p_{iy} + r_{33}p_{iz} + r_z - b_{iz})^2) \quad (3.8)$$

and J is the Jacobian matrix obtained by differentiating (3.7) with respect to kinematics time and is given by (3.9) .

$$J = [J_{i1} \quad J_{i2} \quad J_{i3} \quad J_{i4} \quad J_{i5} \quad J_{i6}] \quad (3.9)$$

Where

$$\begin{aligned} J_{i1} &= (r_{11}p_{ix} + r_{12}p_{iy} + r_{13}p_{iz} + r_x - b_{ix}) / l_i \\ J_{i2} &= (r_{21}p_{ix} + r_{22}p_{iy} + r_{23}p_{iz} + r_y - b_{iy}) / l_i \\ J_{i3} &= (r_{31}p_{ix} + r_{32}p_{iy} + r_{33}p_{iz} + r_z - b_{iz}) / l_i \\ J_{i4} &= J_{i1} \frac{\partial J_{i1}}{\partial \alpha} + J_{i2} \frac{\partial J_{i2}}{\partial \alpha} + J_{i3} \frac{\partial J_{i3}}{\partial \alpha} \\ J_{i5} &= J_{i1} \frac{\partial J_{i1}}{\partial \beta} + J_{i2} \frac{\partial J_{i2}}{\partial \beta} + J_{i3} \frac{\partial J_{i3}}{\partial \beta} \\ J_{i6} &= J_{i1} \frac{\partial J_{i1}}{\partial \gamma} + J_{i2} \frac{\partial J_{i2}}{\partial \gamma} + J_{i3} \frac{\partial J_{i3}}{\partial \gamma} \end{aligned} \quad (3.10)$$

The Newton Raphson numerical algorithm converges to a solution in four or five iterations in most cases but it depends up on the initial guess taken and the trajectory followed. This can be seen from simulation results given in section 3.5.

3.4. NEURAL NETWORK ESTIMATION METHOD

Neural networks are massively parallel distributed processing systems made up of highly interconnected processing elements that have the ability to learn and acquire knowledge and make it available for use. They are efficient in problems like the forward kinematics problem where input output data is readily available but it is difficult to get easy and working mathematical relations. In the forward kinematics problem, input output data can easily be generated using inverse kinematics, (3.3).

Classical theory of function approximation supports the use of neural networks for function approximation. Generally a feed forward network with sufficient number of neurons in hidden layer can approximate any continuous function to any desired accuracy. In the forward kinematics problem the required functional mapping is from 6 measured joint input values to three Cartesian position and three orientation angle values.

The effectiveness of a neural network in solving such a problem is measured by the complexity of the network, which is indicated by the number of neurons and weights, relative to the complexity of the problem itself. Fig.3.2 shows a fully connected feed forward neural network with three inputs and two outputs. Below two most important points in neural network implementation are described.

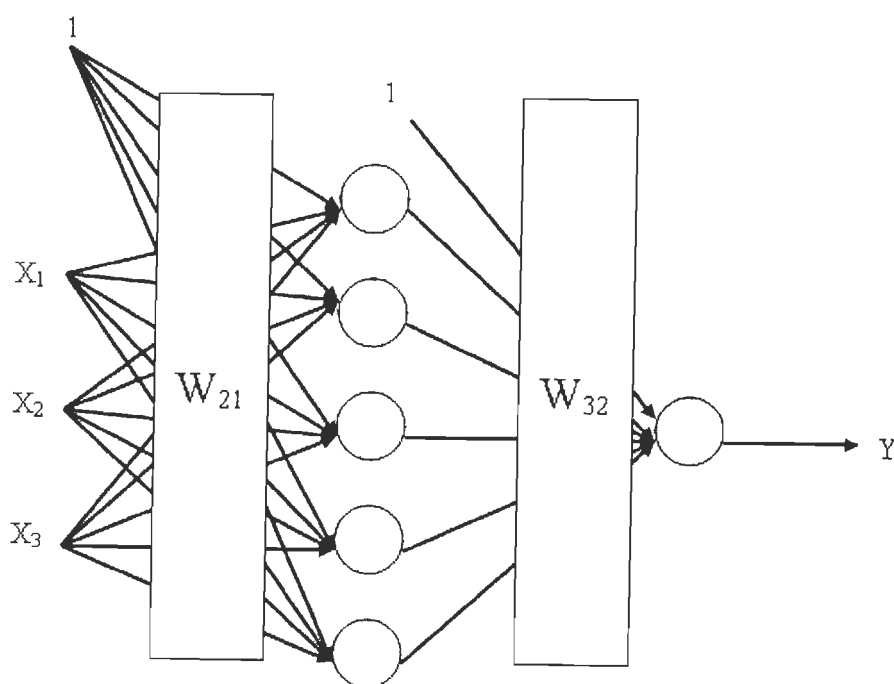


Fig.3.2 Three layer feed forward neural network

3.4.1. Training data generation

Neural network has to be trained by giving it sample inputs and corresponding output values and a training algorithm will adjust the connection weight and bias values until a minimum error or other stopping criteria is reached. The training data has to be taken carefully to consider the complete input range. Normalization and other preprocessing of the data improve the training performance.

3.4.2. Selection of network size

The size of a network refers to the number of layers and the number of neurons in each layer. The problem of selecting appropriate network size is one of the draw backs of neural networks. There is no direct method of deciding the size of a network for a given problem

and one has to use experience or trial error method. In general, when a network is large, the complexity of the function that it can approximate will also increase. But as the network size increase, both training time and its implementation cost increase and hence optimum network size has to be selected for a given problem.

3.5. SIMULATION RESULTS AND COMPARISON

For simulation study, a general 6 DOF Stewart platform manipulator with circular base and platform having geometric specifications given in table I is used [7]. As can be seen from the parameter values of the table, the leg is a piston and cylinder assembly with mass of each leg being 8Kg. In the controller simulations, the leg is assumed to be driven by electrical motors. The mass of the platform is 32Kg and its inertia matrix is taken as diagonal matrix with the assumption that load is symmetrical and platform is circular. The leg inertia is also symmetrical. From the geometric description, the nominal position of the platform center in the base frame is given by

$$R=[0 \ 0 \ 1.563]m$$

and the position of each of the universal joints and spherical joints of the legs with respect to the base is given by

$$b_1 = [0.6928 \ 0.4000 \ 0]$$

$$b_2 = [0.0000 \ 0.8000 \ 0]$$

$$b_3 = [-0.6928 \ 0.4000 \ 0]$$

$$b_4 = [-0.6928 \ -0.4000 \ 0]$$

$$b_5 = [-0.0000 \ -0.8000 \ 0]$$

$$b_6 = [0.6928 \ -0.4000 \ 0]$$

$$p_1 = [0.4830 \ 0.1294 \ 1.5630]$$

$$p_2 = [-0.1294 \ 0.4830 \ 1.5630]$$

$$p_3 = [-0.3536 \ 0.3536 \ 1.5630]$$

$$p_4 = [-0.3536 \ -0.3536 \ 1.5630]$$

$$p_5 = [-0.1294 \ -0.4830 \ 1.5630]$$

$$p_6 = [0.4830 \ -0.1294 \ 1.5630]$$

3.5.1. Numerical Method

Newton Raphson numerical algorithm is implemented using MATLAB. Equations (3.6) to (3.8) are written into an mfile as functions and the nonlinear optimization function fsolve is used to solve the nonlinear equations. The algorithm implemented is similar to [11][114] and the pseudo code is described below.

Table3.1 Geometric Specifications of Stewart platform

Joint	1	2	3	4	5	6
Base	$\pi/6$	$\pi/2$	$5\pi/6$	$7\pi/6$	$3\pi/2$	$11\pi/6$
Platform	$\pi/12$	$7\pi/12$	$3\pi/4$	$5\pi/4$	$17\pi/12$	$23\pi/12$
Base radius 0.8m						
Platform radius 0.5m						
Mass of platform 32kg						
Mass of upper leg 4kg						
Mass of lower leg 4kg						
Initial Height 1.5m						
Platform Inertia $I_{xx}=2, I_{yy}=2$ and $I_{zz}=4$						
Leg Inertia upper $I_{xx}=0.75, I_{yy}=0.75, I_{zz}=0.018$						
Leg Inertia lower $I_{xx}=0.03, I_{yy}=0.03, I_{zz}=0.002$						
CG of upper leg 0.75m from top						
CG of lower leg 0.15m from base						

Step 1: Start with initial guess

Step 2: Calculate length of six legs for the given position and orientation using (3.5)

Step 3: Calculate the error between given leg length and the calculated value in step 2. If error is greater than tolerance go to 4 else go to step 6

Step 4: Calculate Jacobian matrix using (3.7)

Step 5: Evaluate new position using (3.6) and go to step 2

Step 6: Stop

The algorithm is tested for the trajectories given in (3.11)-(3.13) The maximum error and the time to converge for different trajectories is tabulated in table3.2 The result shows that the numerical algorithm gives very small maximum error for trajectories where there is no pitch motion but the time taken to converge is relatively longer.

3.5.2 Feed Forward Neural Network

There are various types of neural network architectures which could be used for nonlinear function estimation. For example, radial basis, feedforward neural networks, adaptive neuro fuzzy inference systems (ANFIS) and cerebella model arithmetic computer networks. In this chapter the simplest and most common fully connected feedforward neural networks are

used. They are selected based on space and time complexity requirements. It is very well known that the size of radial bases network increases with the training data size. Hence with the large training size required for Stewart platform, due to their large size they were not suitable. ANFIS networks have been used for serial robots in [130] but were not selected because of their size problem. Hence two layer and three layer feed forward neural network trained with back propagation algorithm are used to estimate the forward kinematics.

3.5.2.1. Training data generation

Data used for the training of the neural networks is generated using inverse kinematics formula (3.3). Randomly selected platform positions and orientations are given to the inverse kinematics formula and corresponding leg lengths are generated. The leg lengths are taken as input to the neural network and the randomly generated positions and orientations as output and the network is trained. The training process is shown in Fig.3.3.

3.5.2.2. Network size selection

To select the most optimal network size, networks with different sizes and training data are taken and trained offline. The training data is generated by taking random Cartesian space positions and orientations and then calculating corresponding leg lengths using inverse kinematics formula given in (3.3). The range of values for the work space is given in table3.3. The performance of different networks with respect to training time and mean squared error is given in table3.4. Generally, three layer networks have better performance than two layer networks but they have longer training time. Among the three layer networks taken, training performance improves as the number of hidden layer neurons increase which is expected but the trend stops after some time. The MSE of the last network, having 30 and 35 neurons, is bigger compared to the one having 25-35 neurons. One reason for this is, for a good training performance the ration of number of tunable parameters to that of training data size has to be very small and in here network size has increased but training data size is the same. For the last network, the number of tunable parameters is 1511 and ration is 0.252. Increasing the training data size increases the training time and the required memory size drastically.

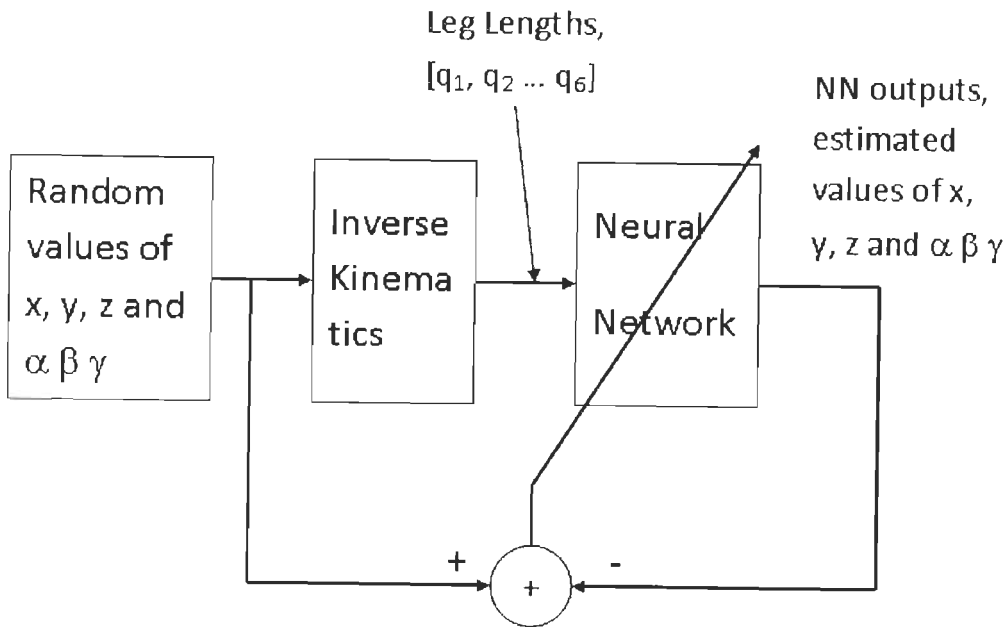


Fig3.3 Neural network training

Therefore for the rest of the simulation, the network with 25-35-hidden layer neurons is taken. Fig.3.5 shows the training performance of this three layer network.

The above network is then tested and compared for its estimation performance when the manipulator is moving in different trajectories and table 3.4 gives result in comparison with the output of the numerical. Trajectories 1-3 are circular trajectories on X-Y plane with various orientation angles, 4-6 are helical trajectories with various orientation angles and 7 is linear. The numerical algorithm gives better result, in terms of estimation error, for all trajectories except 2, 5 and 7. These trajectories contain pitching motion of the platform and they have bigger errors. The algorithm has been checked with various initial conditions and with a different geometry given by [176]. The result is found to be the same. The trained neural network gives more or less a constant estimation error for all trajectories and the time taken is always less than the numerical. Therefore the neural network can be used to estimate forward kinematics better than the numerical for application that need higher accuracy.

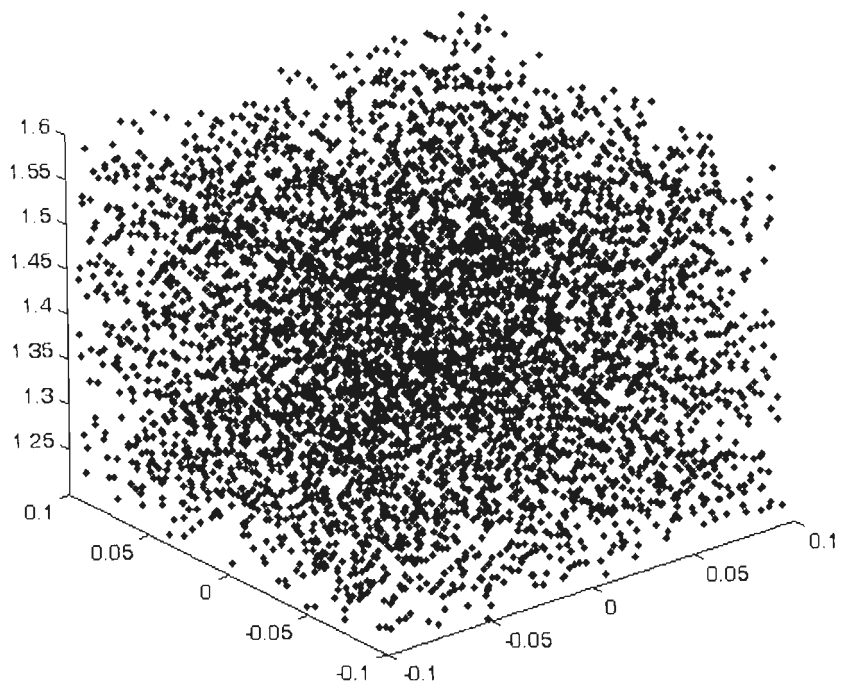


Fig.3.4 Plot of the random values of x , y , z , α , β and γ values used for training data generation

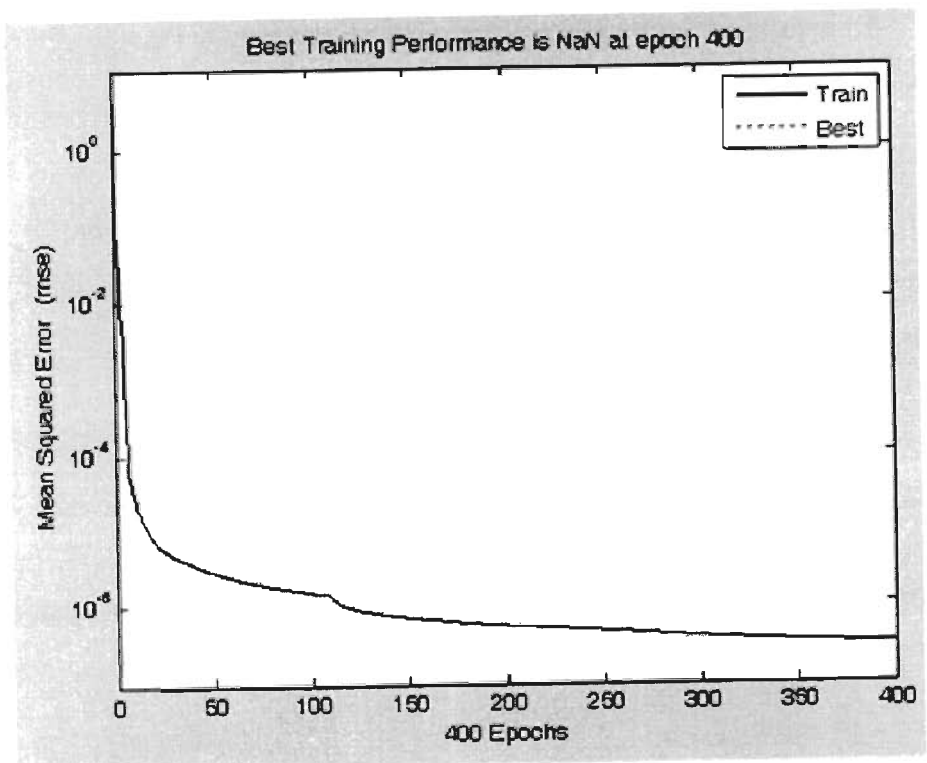


Fig 3.5 Training performance of three layer feed-forward network

3.6. CONCLUSION

The forward kinematics problem of a general Stewart platform manipulator with planar base and platform is formulated and two important solutions, numerical Newton Raphson method and neural network method, were compared using seven different trajectories.

TABLE 3.2 Cartesian position and Orientation Limits of Center of Platform

	Limiting values
x	[-0.15m 0.15m]
y	[-0.15m 0.15m]
z	[1.3m 1.83m]
α	$[-\pi/18 \ \pi/18]$
β	$[-\pi/18 \ \pi/18]$
γ	$[-\pi/18 \ \pi/18]$

The numerical algorithm performs very well for a trajectory that does not have pitching motion but gives bigger error when the trajectory has pitching motion. On the other hand, the numerical algorithm gives consistent estimation error for all trajectories. When the average time taken is compared, the neural network responds faster than the numerical algorithm which shows that the neural network can be used for faster manipulators. Hence a feed forward neural network can be used to estimate the forward kinematics of Stewart platform manipulator for faster application

Forward kinematics estimation

Table 3.3 Comparison of different networks based on their training performance

<i>Network used</i>	<i>Network size</i>	<i>MSE</i>	<i>Training time(sec)</i>
2 layer network	10-6	1.38×10^{-5}	1.58×10^3
	15-6	1.64×10^{-6}	2.49×10^3
	40-6	9.1×10^{-7}	8.5×10^3
3 layer network	10-10-6	2.90×10^{-6}	3.1×10^3
	15-15-6	5.89×10^{-7}	3.5×10^3
	15-25-6	2.90×10^{-7}	1.19×10^4
	17-35-6	6.25×10^{-8}	2.06×10^4
	25-35-6	3.64×10^{-9}	2.89×10^4
	30-35-6	2.35×10^{-7}	4.38×10^4

Forward kinematics estimation

Table 3.4 Comparison between the performance of numerical and NN methods with respect to trajectory tracking

	Numerical error (mm, 10^{-3} rad)		NN error (mm, 10^{-3} rad)		Average time(m sec)	
	Max error1	Max error 2	Max error 1	Max error2	num erica l	NN
1	0	0	0.12	0.15	15.2	9.4
2	5.8	32.5	0.53	0.38	20.5	10.6
3	0	0	1.4	0.9	18	9.8
4	0	0	1.1	0.9	20	11.6
5	5.6	31.6	1.1	0.9	20	11.8
6	0	0	1.2	1.0	19.1	11.8
7	5.4	31.4	0.086	0.052	21.8	11.0

* error1 and error2 are position error and orientation angle errors

CHAPTER 4

FUZZY TUNED JOINT SPACE PID CONTROLLER FOR STEWART PLATFORM MANIPULATORS

In the last chapter, we proposed a soft computing solution for the forward kinematic estimation problem of Stewart platform manipulator. This forward kinematic estimation is needed when a controller is implemented in task space. Before we discuss task space controllers which utilize the forward kinematic estimation methods, it is necessary to investigate if soft computing techniques can be used to improve existing joint space controllers. The best candidate for this is the industrially famous joint space PID control. Therefore, in the following sections, we will discuss how a fuzzy logic system can be used to solve the synchronization problem of independent leg PID control and compensate uncertainties. It is well documented that one of the promising techniques used to handle uncertainties is fuzzy logic system [26]. The field of fuzzy sets and logic was first introduced by Lotfi Zadeh and fuzzy control was first introduced by E. Mamdani. Since then, fuzzy logic systems have gained great popularity in handling uncertainty. In this chapter, we present the design and stability analysis of a fuzzy logic system based modeless adaptive controller used for trajectory tracking of Stewart platform manipulator. The basic controller structure is SISO PID type of controller and the three gain parameters are tuned using fuzzy logic system. Like the independent PID control, each leg is controlled by a PID controller but the gains are varied by a fuzzy logic system. The input to each fuzzy logic system is a weighted sum of errors in all legs and the rate of change of the weighted sum of errors. The weight factor is taken based on intuition and it enables to minimize the coupling error between the legs, which is drawback of independent leg PID control. Simulation results

revealed that the controller has a good adaptive performance and achieves a better tracking accuracy than simple PID controller.

4.1. INTRODUCTION

The various types of controllers designed for Stewart platform manipulator can be broadly classified as modelless and modelbased controllers [36][43]. The model based controllers such as computed torque controllers [7], sliding mode controllers [113] and passivity based controllers [173] have the potential of giving higher precision in trajectory tracking and global asymptotic stability can be achieved if the dynamic model parameters are identified exactly. But the complexity of the control algorithms has delayed their use in practical application and PID controllers are still in use. Hence it may be better to try to improve these controllers before starting a newer design.

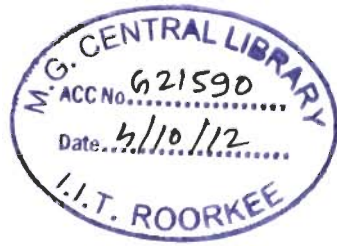
PID controllers, which are mostly implemented in joint space in SISO, do not guarantee high performance because they are unable to compensate the highly nonlinear dynamics of the system and the uncertainty due to variation of the payload [7]. Therefore we propose here to use fuzzy logic to tune the gains of PID controller to improve their performance. Fuzzy logic controllers have been used for various applications and [8][13][6][63][164] discuss the use of fuzzy logic for tuning the gains of a PID controller. In [6] the performance of a conventional PID controller tuned by Ziegler Nichols method, a fuzzy logic PID like controller and a PID controller whose gains are tuned by fuzzy logic have been compared and it was verified that tuning the gains of PID controller using fuzzy logic results in a better controller when the controlled system has nonlinearity. In [164] the use of fuzzy logic for tuning PID controller for non linear systems with H-infinity tracking performance is well reported and the paper has shown that using fuzzy logic to tune PID controller gains will give a better tracking accuracy for nonlinear systems. Therefore in this chapter we will investigate the application of fuzzy logic based PID tuning for Stewart platform manipulator control. In our proposed controller, the fuzzy logic system has additional advantage of achieving synchronization. In most of the above mentioned fuzzy logic based tuning algorithms [8][13][6][63][164], the fuzzy rule base is obtained from step response of the controlled system with the objective of minimizing settling time. In our implementation also similar steps are followed but a weighted sum of errors is taken as input of the fuzzy logic to consider the coupling between

the legs. Hence in our proposal, like the joint space PID control, each leg is controlled by a PID controller but the gains are varied by a fuzzy logic system. The input to each fuzzy logic system is a weighted sum of errors in all legs and the rate of change of the weighted sum of errors. The weight factor is taken based on intuition and it enables to minimize the coupling error between the legs, which is drawback of independent leg PID control. Simulation results have shown that the controller drives the Stewart platform in a desired trajectory with a very good precision.

4.2. DESIGN OF CONTROLLER

A PID controller in its standard form is given as

$$u(t) = K_p \left[e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int_0^t e(t) dt \right] \quad (4.1)$$



Multiplying the gain term outside the bracket by the derivative and integral time constants, the equation can also be written as

$$u(t) = K_p e(t) + K_D \frac{de(t)}{dt} + K_I \int_0^t e(t) dt \quad (4.2)$$

Hence the parameters to be tuned are the three gains K_p , K_D and K_I . There are different tuning algorithms designed for linear systems that aim at achieving control specifications such as set point following, disturbance rejection, robustness to model uncertainty and rejection of measurement noise. The traditional tuning algorithms try to achieve one of these specifications. In the current application the system is required to follow a desired trajectory, which means control objective is not regulation but tracking, and moreover the system is required to operate at different payload conditions. Therefore the control system should be robust to model uncertainty and for that we use fuzzy logic to tune the three gains.

4.2.1 Fuzzy logic system

Fuzzy logic provides a formal methodology for representing, manipulating and implementing a human heuristic's knowledge about how to control a system [26]. The heuristic's knowledge is embedded into the controller in the form of IF... THEN rules. Hence the fuzzy PID controller will have IF... THEN rules and an inference engine. The inference engine uses

the rules to produce an output for a given input. The rules are of the following type.

If E is PS and ER is NB then α is PS

where E and ER are inputs to the fuzzy logic system, α is the output, PS and NB are linguistic values of the inputs and output.

Error, E and rate of error, ER are inputs to the controller and the output α is the change in control gains. The terms PS and NB stand for positive small and negative big respectively. Then a decision is made about the current output value using an inference mechanism and the rule base which contains IF...THEN rules. A final output interface part converts the decision to a numerical value. The fuzzy logic tuned PID control system discussed in this chapter is shown in Fig.4.1. Figure4.2. shows MATLAB Simulink implementation of the controller for a single leg. The input to the block, which is represented as 1, is not a single error but it is weighted sum of the errors from other legs also. This means, the fuzzy logic system tunes the parameters of the PID controller based on absolute error of each leg and the coupling errors from other legs.

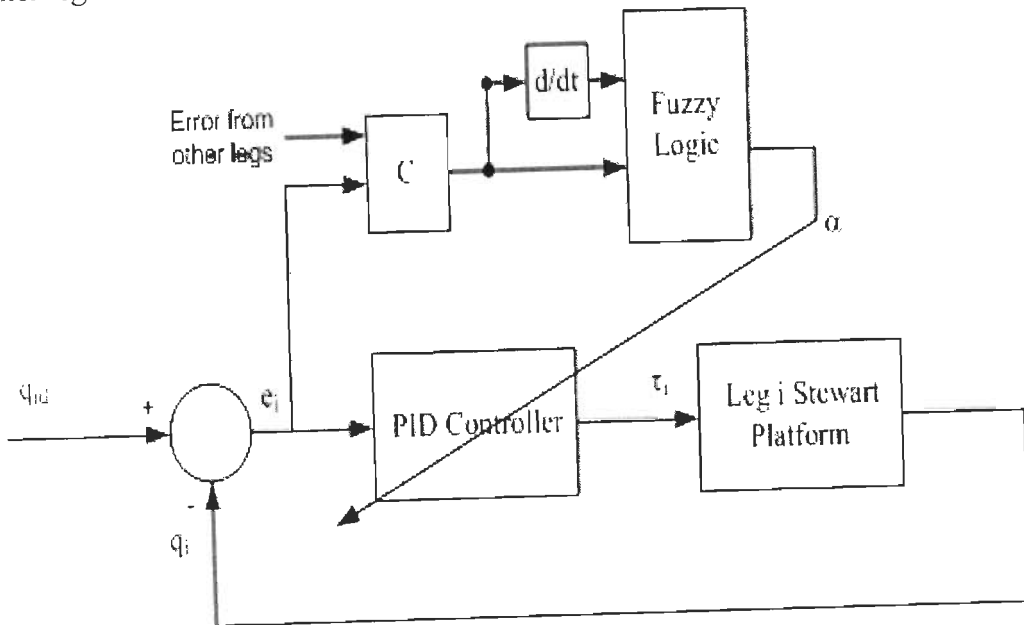


Fig.4.1 Fuzzy tuned joint space PID control of a single leg

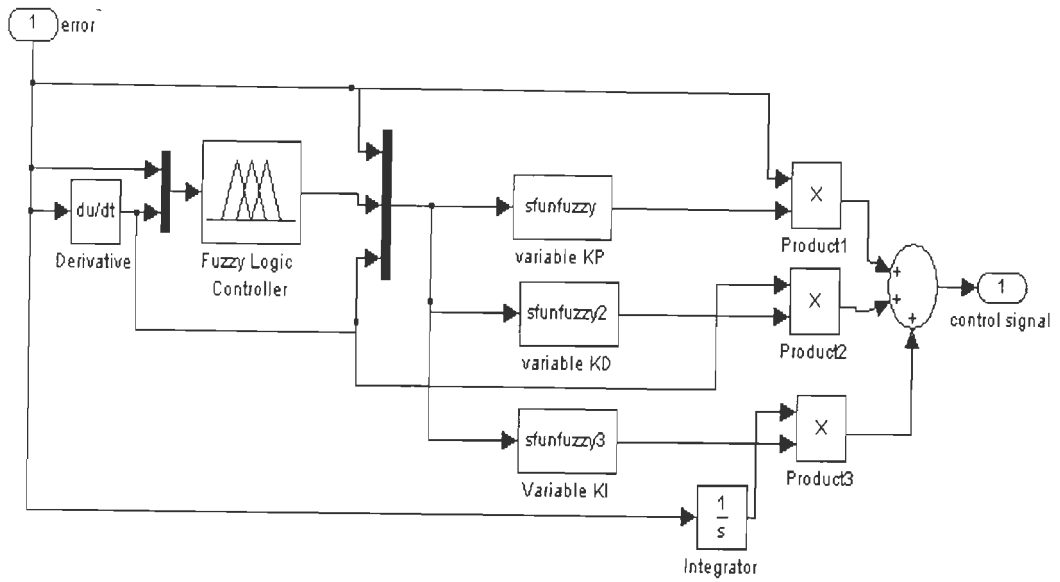


Fig.4.2 Simulink block diagram model of single leg PID tuned controller

4.2.2 Tuning algorithm

For the fuzzy logic gain tuning, error E and rate of error ER are each divided into five linguistic values of negative big (NB), negative small(NS), Zero(Z), positive big(PB) and positive small (PS). The membership functions used are symmetric triangular functions and they are shown in Fig.4.3-Fig.4.5. The output of the fuzzy controller α_f is then used to change the gains of the PID controller as given by the following algorithm.

$$K_p(k+1) = K_p(k) + \alpha_f(k)e(k)K_p(0) \quad (4.3)$$

$$K_D(k+1) = K_D(k) + \alpha_f(k)\dot{e}(k)K_D(0) \quad (4.4)$$

$$K_I(k+1) = K_I(k) + \alpha_f(k)e(k)\dot{e}(k)K_I(0) \quad (4.5)$$

The initial values $K_p(0)$, $K_D(0)$ and $K_I(0)$ and their ranges are determined taken from a separate PID controller tuned manually and which gives acceptable output.

4.3 SIMULATION RESULTS AND DISCUSSION

For the simulation study of the performance of the controller, a typical 6-6 geometry Stewart platform with the geometric parameters given in table3.1 of chapter 3 is implemented using simmechanics tool box of MATLAB. The fuzzy logic tuner is also implemented using fuzzy

Fuzzy tuned joint space PID

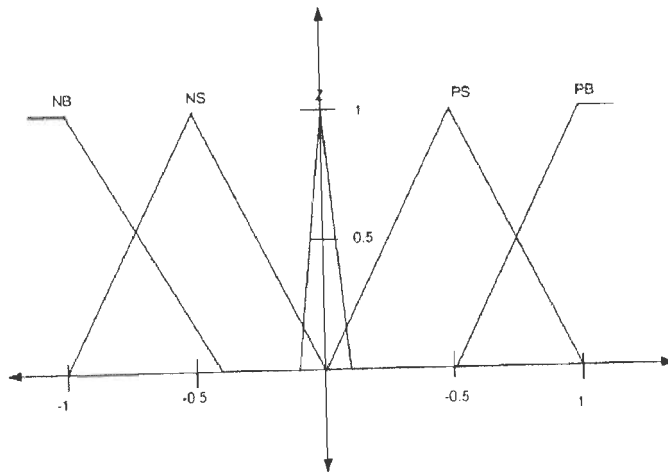


Fig.4.3 Graph of membership functions for error E

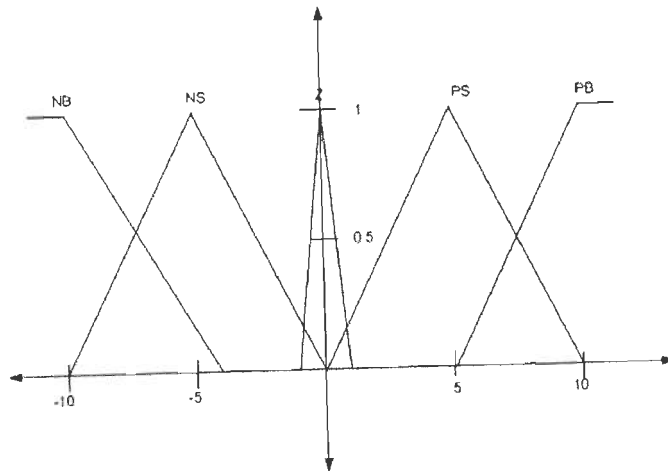


Fig.4.4 Graph of membership functions for error rate, ER

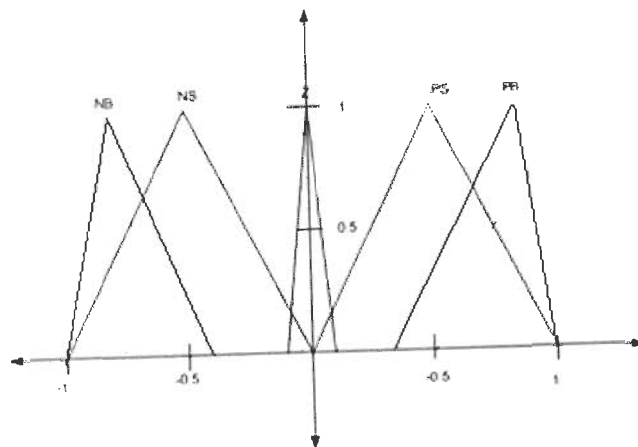


Fig.4.5 Membership functions for output variable

logic toolbox of MATLAB and three mfile sfunctions were written to implement the discrete difference equation given in (4.3)-(4.5). The trajectory taken to test the performance of the controller is as given by (4.6)-(4.8). It has a heave or uplift motion in z direction for 2.5 sec followed by circular motion in the x-y plane. To make the trajectory twice differentiable, the trajectories are joined with parabolic blend. The desired trajectories are plotted in Fig.4.6a-c and the actual trajectories for the simple PID and fuzzy tuned PID controllers are given in Figs.4.7-Fig.4.10. The superior performance of the fuzzy tuned PID is clear when we compare the PID controlled manipulator's joint space tracking errors of three legs shown in Fig.4.7 with the tracking errors of the same legs when the manipulator is controlled by fuzzy tuned PID as given in Fig.4.8. The controller performance is improved by 50% as the error decreased from 10 μ m to 5 μ m. Similarly the task space tracking performance of the fuzzy tuned PID appears to be better when we compare the results shown in Fig.4.9 and Fig.4.10. The joint space PID tracking error is in the order of $\pm 150\mu$ m while the fuzzy tuned PID has reduced the error to $\pm 40\mu$ m. This shows the bigger performance advantage obtained by using fuzzy logic system to tune simple PID controller.

$$x(t) = \begin{cases} 0 & 0 \leq t \leq 0.35\text{sec} \\ c_{11}t^5 + c_{12}t^4 + c_{13}t^3 + c_{14}t^2 + c_{15}t^1 + c_{16} & 0.35 < t \leq 0.5 \\ 0.15\sin(\omega(t-0.35)) & t > 0.5 \end{cases} \quad (4.6)$$

$$y(t) = \begin{cases} 0 & 0 \leq t < 0.25\text{sec} \\ c_{21}t^5 + c_{22}t^4 + c_{23}t^3 + c_{24}t^2 + c_{25}t^1 + c_{26} & 0.25 \leq t < 0.35 \\ 0.15\cos(\omega(t-0.35)) & t > 0.35 \end{cases} \quad (4.7)$$

$$z(t) = \begin{cases} c_{31}t^5 + c_{32}t^4 + c_{33}t^3 + c_{34} & 0 \leq t < 0.35 \\ 0 & t > 0.25 \end{cases} \quad (4.8)$$

The control signals of the two controllers were also compared and the fuzzy logic tuned PID controller has a control signal similar to the simple PID controller in its smoothness and magnitude. The gains of the controller used for the simulation are $K_P = 8.5 \times 10^7$, $K_D = 9 \times 10^5$, and $K_I = 100$.

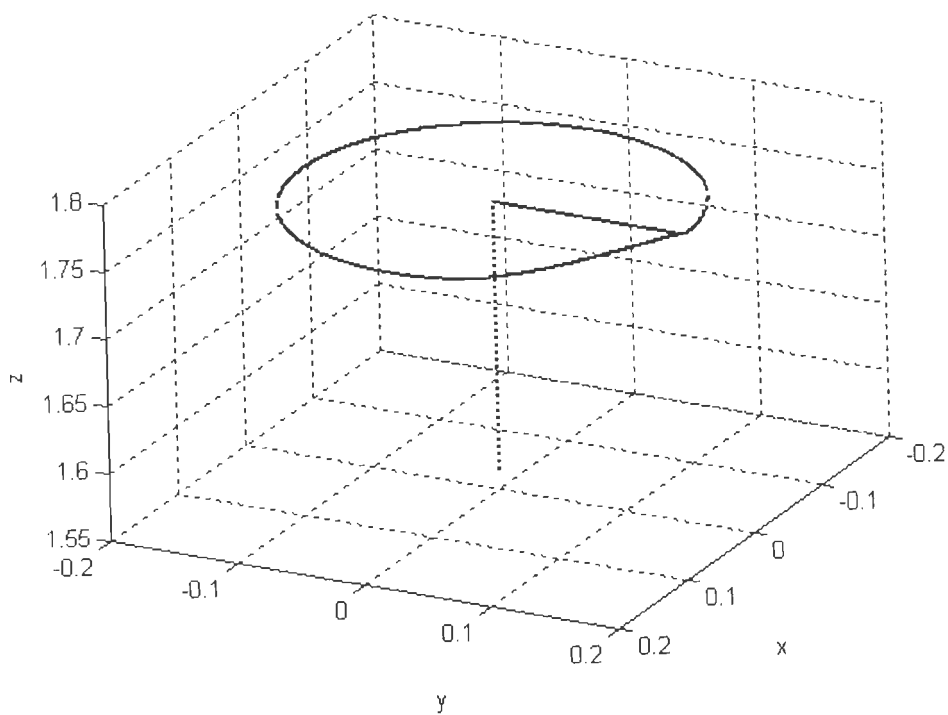


Fig.4.6a) Three dimensional view of the desired trajectory

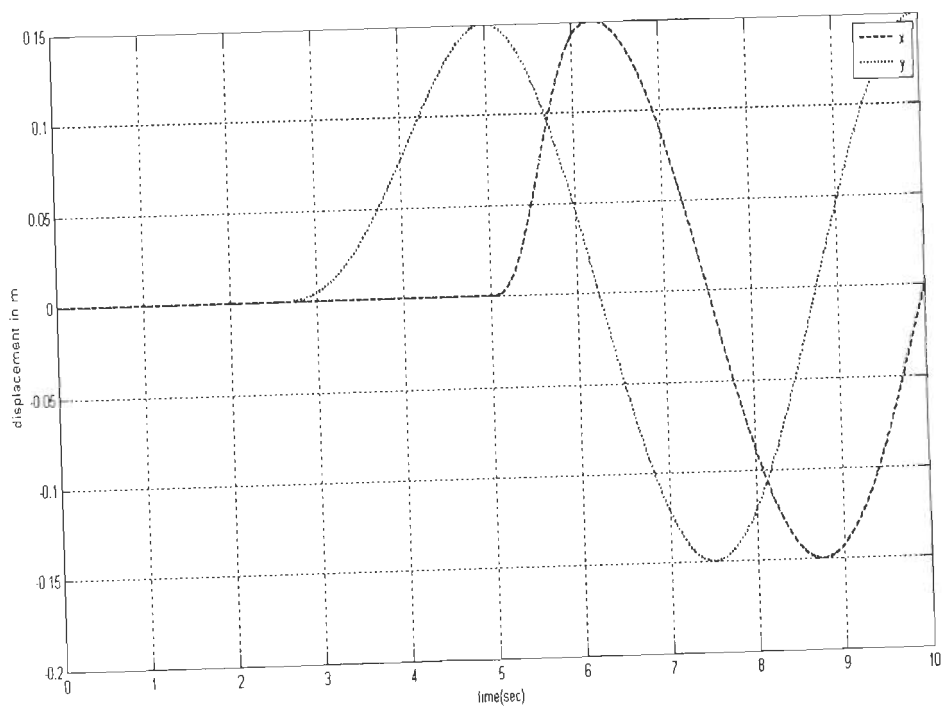


Fig.4.6b) Desired trajectory in x and y directions

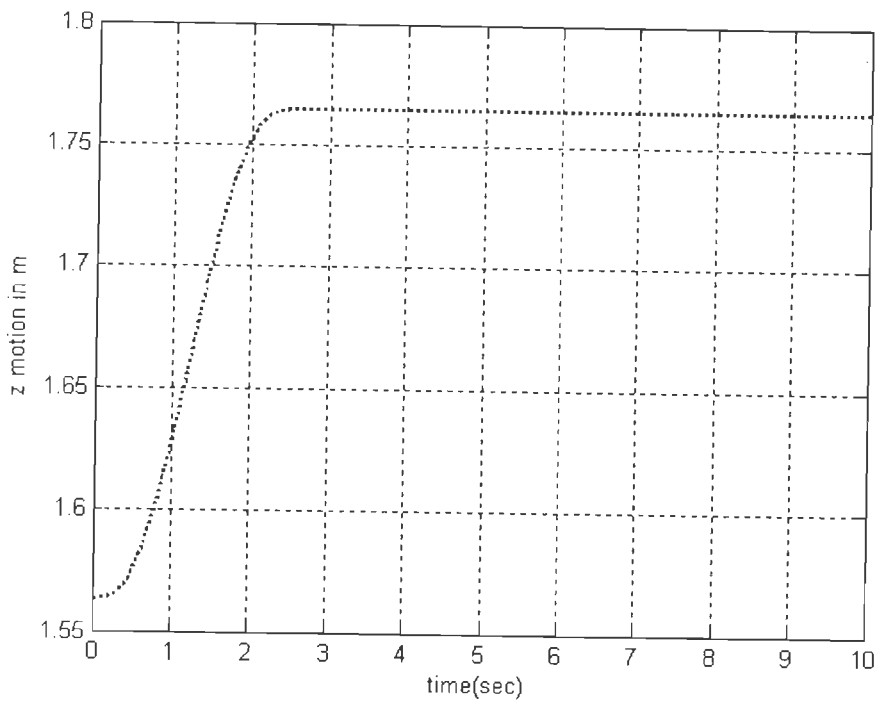


Fig.4.6c) Desired trajectory in Z direction

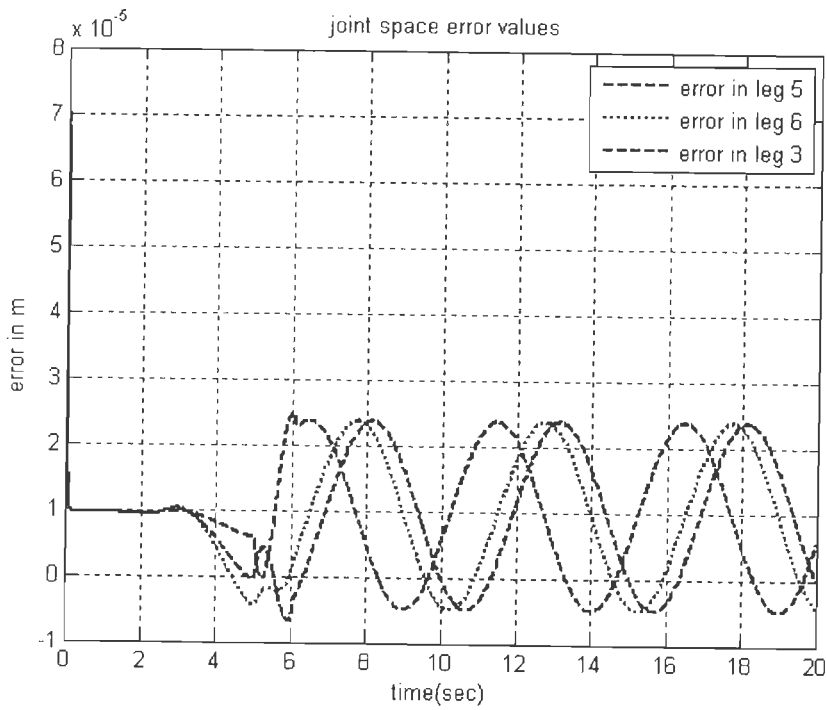


Fig.4.7 Joint space tracking error of simple PID

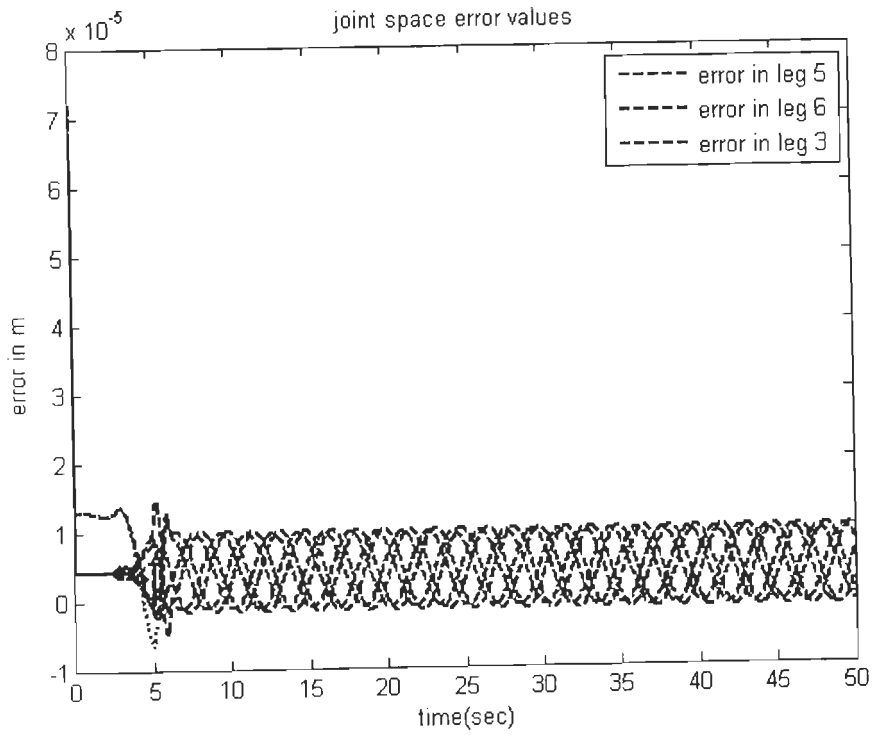


Fig.4.8 Joint space tracking error of fuzzy tuned PID

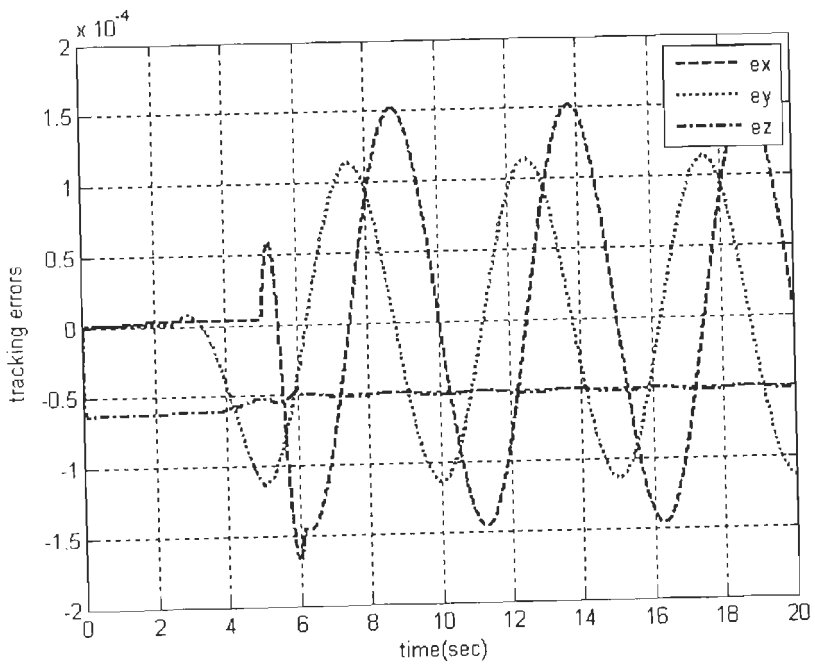


Fig.4.9 Task space tracking error for simple PID

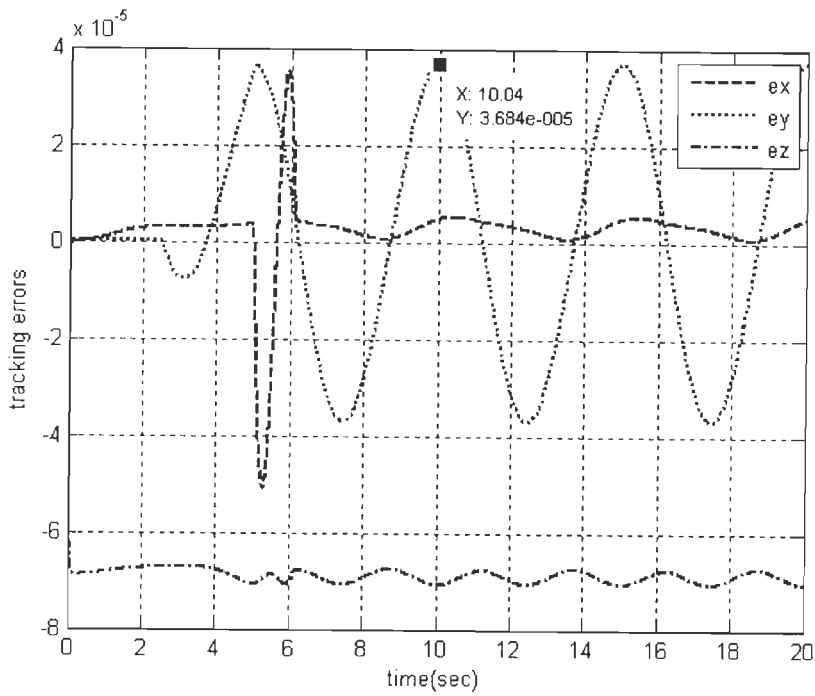


Fig.4.10 Task space tracking errors of fuzzy tuned PID for x, y and z directions

4.4 Conclusion

In this chapter we have seen how a fuzzy logic system can be used to tune an independent leg PID controller. The results have shown that the fuzzy logic system is able to improve the performance of the PID controller and tracking error has reduced by an appreciable amount. However it is clear that the linear PID controller still has limitations in compensating the nonlinear leg dynamics. This shows that the linear PID controller has to be changed by a nonlinear and more robust controller. In the next chapters, we will discuss sliding mode controllers and their implementation.

CHAPTER 5

SLIDING MODE CONTROL AND IMPROVEMENTS FOR HIGH PERFORMANCE

In chapter 4, we have seen the implementation of a modeless controller in joint space. The controller basically is a linear SISO controller with enhancement of fuzzy logic system. The fuzzy logic system is used to minimize the coupling error between legs and to enhance linear PID controllers to compensate for leg dynamics. Though the fuzzy logic tuning has enhanced the performance of the simple PID controller, the performance improvement is not much. Further improvements can be obtained by using nonlinear controllers such as sliding mode controller. This chapter presents design and stability analysis of various types of sliding mode controllers. As discussed in chapter 2, compared with task space approach, joint space control has limitations in achieving high performance. Hence, the controllers discussed in this chapter are mainly task space except the last section which deals with hybrid approach.

In the first section the design and stability analysis of task space fuzzy sliding mode controller is given. Then section two presents an improvement to the fuzzy sliding mode controller. The controller is fuzzy sliding mode controller with integral loop which uses a PI loop in parallel to with the fuzzy sliding mode controller. The last section deals with a new kind of sliding mode controller which is a hybrid of task space and joint space controllers. The sliding mode controller uses a new sliding surface which helps to drive synchronization error to zero and the controller achieves high performance in task space without using the complex forward kinematics.

5.1. TASK SPACE FUZZY SLIDING MODE CONTROLLER

In task space approach, the control of Stewart platform is MIMO and has the potential to drive a 6 DOF parallel robot at high speed with high accuracy. However, it has two major implementation problems. First one is it needs forward kinematics measurement or estimation and second one is parallel implementation is difficult and controller to be used has to be simple to decrease the burden. With these considerations, a fuzzy sliding mode controller is proposed in this section. The main assumption taken is, due to the robustness of SMC against uncertainties [14], [66], estimation errors can be compensated and an easier estimation method can be used for forward kinematics. Moreover the same robust characteristics of sliding mode controller avoid the need for explicit friction estimation unlike the robust nonlinear controller of [58] and implementation becomes easier. Next we will discuss the design of the controller.

5.1.1 Design of the Controller

The dynamic equation of the system in state space without considering actuator friction and external disturbance is given as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= M^{-1} \left(J_p^{-T} \tau - C(x_1, x_2) X_2 - G(x_1) \right) \end{aligned} \quad (5.1)$$

Where x_1 is (6x1) state vector of Cartesian space positions and orientations and x_2 is (6x1) state vector of the Cartesian space velocities. Let x_d be (6x1) vector of desired task space trajectories. Then, the task space tracking error and its rate vector are given as

$$e = x_d - x \quad (5.2)$$

$$\dot{e} = \dot{x}_d - \dot{x} \quad (5.3)$$

Then the control objective is to find a control signal τ that can drive system (5.1) towards a sliding manifold and keep it there so that the error e and its derivative asymptotically move to zero. The sliding manifold is given by

$$S = \Lambda e + \dot{e} \quad (5.4)$$

where Λ is a diagonal matrix, and it determines the rate at which the system moves towards the sliding manifold. By taking Λ to be diagonal matrix, we have decoupled the system into a

second order system and hence one may be tempted to think that the system has lost its coupling and synchronization or cross coupling errors may degrade performance. But since the sliding variable is calculated in the task space, it has to be transformed to joint space using manipulator Jacobian. And this step will bring back the coupling effect back into effect and the controller has better performance.

Taking the derivative of (5.4) with respect to time,

$$\dot{S} = \Lambda \dot{e} + \ddot{e} \quad (5.5)$$

when the system is in sliding mode, $\dot{S} = 0$ and using (5.1), (5.2) and (5.3) and nominal values of the system dynamics, the equivalent control signal, which is the torque required to keep the system on the sliding manifold, is obtained as

$$\tau_{eq} = J^T \left\{ M_N (\Lambda (\dot{X}_d - X_2) + \ddot{X}_d) + C_N(X_1, X_2) X_2 + G_N(X_1) \right\} \quad (5.7)$$

Note: x_2 is the velocity signal obtained by differentiating the output of the forward kinematics estimation methods.

To drive the system states towards the sliding surface and compensate for the disturbances, we use a discontinuous controller given by

$$\tau_s = J^T (K f_f(S)) \quad (5.8)$$

Where K is a diagonal matrix chosen based on the disturbance bounds and f_f is fuzzy logic switching system. Hence the combined control signal we propose is given by

$$\tau = \tau_{eq} + \tau_s \quad (5.9)$$

where τ_{eq} is given by (5.7) and τ_s is given by (5.8).

The block diagram representation of the control system is given in Fig.5.2. In the figure, X is vector of the estimated Cartesian position and orientation and \dot{X} is vector of the corresponding velocities obtained by numerical differentiation of X . Similarly X_d , is vector of the desired position and orientation while \dot{X}_d and \ddot{X}_d are desired velocity and acceleration. Next we analyze the stability of the system under the given controller, i.e. we will show that the system has a finite reaching time and its error decays to zero asymptotically.

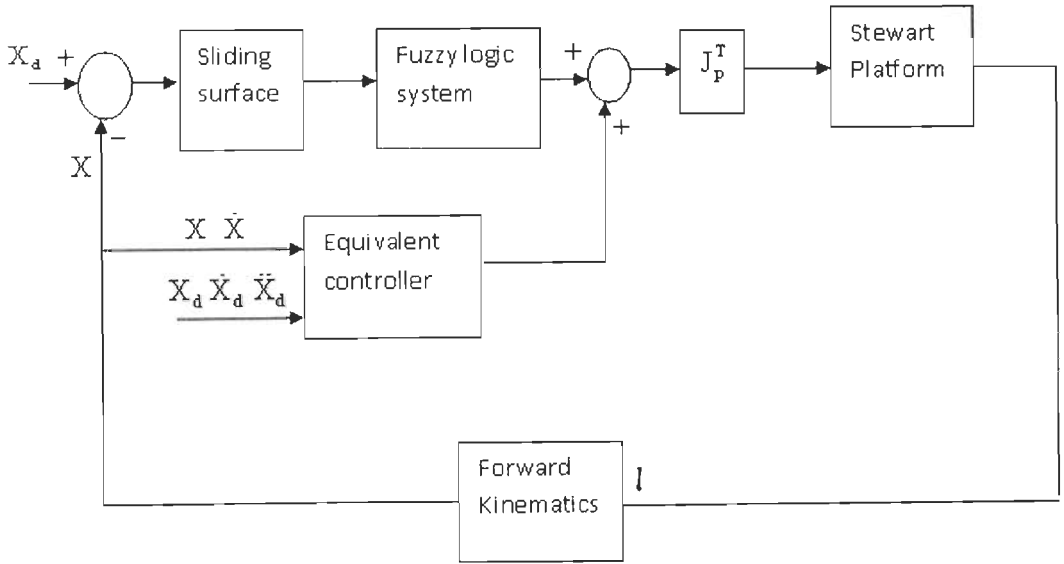


Fig.5.1 Block diagram representation of task space fuzzy sliding mode control system.

5.1.2 Stability analysis

Consider a Lyapunov candidate function given by

$$V = \frac{1}{2} S^2 \quad (5.10)$$

then

$$\dot{V} = S\dot{S} \quad (5.11)$$

$$\dot{V} = S^T (\Lambda \dot{e} + \ddot{e}) \quad (5.12)$$

Then substituting the expressions for the tracking error and its derivative from (5.2) and (5.3) and using the dynamic system model given in (5.1), (5.12) can be written as

$$\dot{V} = S^T \left(\Lambda (\dot{x}_d - \dot{x}_2) + \ddot{x}_d - M^{-1} (J^{-T} \tau - C(x_1, x_2) - G(x_1)) \right) \quad (5.13)$$

Substituting the expression for the control signal from (5.9) in to (5.13) and simplifying

$$\dot{V} = S^T \left(\Delta M (\Lambda (\dot{x}_d - \dot{x}_2) + \ddot{x}_d) - \Delta C(x_1, x_2) \dot{x}_2 - \Delta G(x_1) - K f_r(S) \right) \quad (5.14)$$

For the system to reach to the sliding manifold in a finite time, $\dot{V} \leq -\psi$, where ψ is a positive constant. This can be fulfilled if we impose the following condition on K

$$K \geq G_m + C_m + M_m \left\| \Lambda (\dot{x}_d - \dot{x}_2) + \ddot{x}_d \right\| \quad (5.15)$$

The gain value K in (5.15) depends on the maximum acceleration and velocity and the maximum uncertainties of the dynamics of the system. In section 5, the determination of the

value of K for a typical manipulator will be given. In the derivation of the controller, the actuator dynamics and friction were not considered.

5.1.3. Fuzzy logic switching surface

In the controller given in (5.9), the discontinuous control part used to drive the system towards the sliding mode is implemented using fuzzy logic. As has been described in chapter four fuzzy logic system provides a formal methodology for representing, manipulating and implementing a human heuristic's knowledge about how to control a system. They have been used for control of various systems as standalone controllers and also with sliding mode controllers. Particularly in sliding mode controllers, a fuzzy logic system has the advantage of suppressing chattering. In [71] a fuzzy logic system is used as a nonlinear switching function to control the HEXA robot. In the paper, the sliding variable and its derivative, where used as two inputs and control signal was the output.

For the present case, a single input single output fuzzy logic system is used to control each of the task space dimensions. The input is the sliding variable and the output is the control signal. Hence 6 SISO fuzzy systems are used. The universe of discourse for the input and output is normalized to $[-1 \ 1]$ and tuning parameters are used. Particularly the output tuning term is selected to fulfill the condition given in (5.15). The membership functions used for the input S and the output u are given in Fig.5.2 and Fig.5.3.

Seven linguistic terms are used for both input and output variables and the linguistic terms are NB-Negative Big, Negative-Medium, NS-Negative Small, ZE-Zero, PS-Positive Small, Positive-medium and PB-positive Big. The rule base contains only seven rules as listed below and min/max fuzzification and center of area defuzzification methods are used for the system.

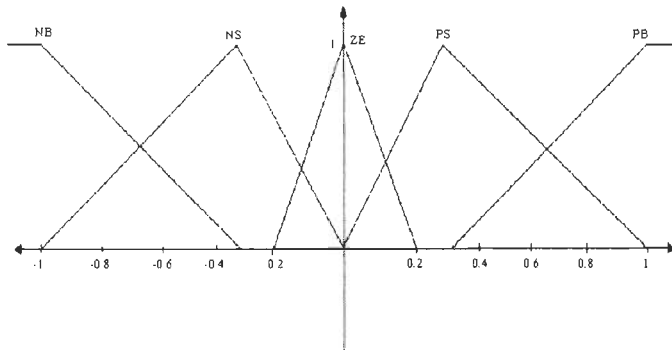


Fig.5.2 Graph of membership functions for the input S

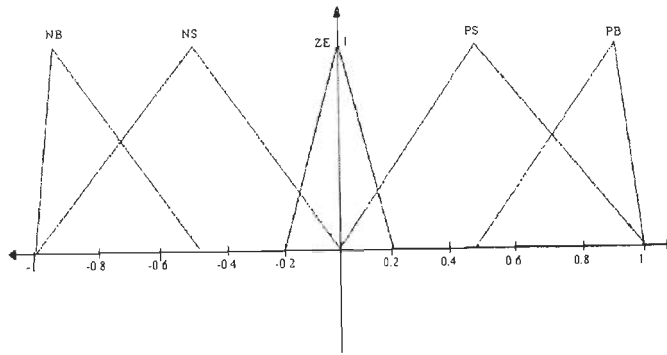


Fig.5.3 Membership functions of the output variable U

IF S is NB THEN U is NB

IF S is NM THEN U is NM

IF S is NS THEN U is NS

IF S is ZE THEN U is ZE

IF S is PS THEN U is PS

IF S is PM THEN U is PM

IF S is PB THEN U is PB

5.1.4 Simulation study

For the simulation study of the performance of the controller, a typical 6-6 geometry Stewart platform with the geometric parameters given in table 3.1 of chapter 3 is again used and implemented using simmechanics tool box of MATLAB. The dynamic parameters are calculated using Lagrangian method and the desired trajectory, given by (5.16), is the same trajectory used in chapter 4 and is a fast trajectory where the platform moves vertically up and down and rotates in an XY plane. It is designed to achieve zero velocity and acceleration at the start of the trajectory.

$$x(t) = 0.15 \{1 - \exp(-\pi t)\} \cos(1.88\pi t), \text{ m}$$

$$y(t) = 0.15 \{1 - \exp(-\pi t)\} \sin(1.88\pi t), \text{ m}$$

$$z(t) = 1.563 + \frac{0.02}{1+0.9t} \sin \left\{ 0.2\pi t \left(\frac{0.1+5.9t}{10.5} \right) + \frac{\pi}{24} \right\}, \text{ m}$$

$$\alpha(t) = 0, \text{ deg}$$

$$\beta(t) = \beta_0 \{1 - \exp(-\pi t)\} \sin(0.86\pi t), \text{ deg}$$

$$\gamma(t) = \gamma_0 \{1 - \exp(-\pi t)\} \sin(0.74\pi t), \text{deg} \quad (5.16)$$

5.1.4.1 Controller Implementation and Performance

The controller given in (5.9) is implemented using MATLAB and Simulink. The fuzzy logic system used for the sliding mode controller is implemented using MATLAB fuzzy logic tool box in Simulink. The tracking performance of the controller is tested for two different cases, no payload and 200kg payload and the results are shown in Fig.5.5-Fig.5.8 below. The parameters of the fuzzy sliding mode controller are determined using a tuning method similar to the Taguchi method. First, the controller is used for single direction regulation control and the step response is observed.

The controller gains are tuned until a desired step response in terms of settling time, overshoot and steady state error is obtained. Then the controller parameters used to obtain the best and worst step responses are used as starting point in the tuning. The initial value for the sliding gain K is obtained from (5.15), i.e. the gain K has to overcome the gravitational, coriolis /centrifugal and inertia uncertainties and these are estimated from the mass variation for the given desired speed as follows. The vertical Z direction motion from nominal value is 0.2m and maximum expected load is 200Kg and hence

$$g_{mz} = 200 * 0.2 * 9.81 = 392.4 \text{NM} \quad (5.17)$$

The maximum speed and acceleration for the desired trajectory are 0.6m/sec and 5.23m/sec² in each direction and using this into the third expression of (32) with the sliding gain being 200, the diagonal elements of the gain K are

$$k_m = 200 \times (0.6 \times 200 + 5.23) = 2.504 \times 10^4 \quad (5.18)$$

Where k_m is a diagonal element for

$$M_m \left\| \Lambda (\dot{X}_d - X_2) + \ddot{X}_d \right\| \quad (5.19)$$

Then the total gain for the sliding surface for each task space direction is the sum of (5.17) and (5.18). Taking this as initial values, the step response analysis was done and the result is given in table5.1. The values of Λ and K in the table are as follows. The step response in x direction is shown in Fig.5.5. It has a rise time less than 0.05sec and zero overshoot with smallest settling time.

The sliding surface slopes

$$\Lambda_1 = \text{diag}_{6 \times 6} (200) \quad (5.19)$$

$$\Lambda_2 = \text{diag}_{6 \times 6} (250) \quad (5.20)$$

$$\Lambda_3 = \text{diag}_{6 \times 6} (300) \quad (5.21)$$

And the gain K is as given below.

$$K = 30 \times 10^4 \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.03 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.03 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.03 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.03 \end{bmatrix} \quad (5.22)$$

Table 5.1 Step response and parameters used

test	K	Λ	% overshoot	ess	Ts(sec)
No load	K	Λ_1	0.166	1.61e-3	0.124
	K	Λ_2	0.114	1.14e-3	0.104
	K	Λ_3	0.1026	1.3e-3	0.118
200Kg load	K	Λ_1	0.1683	1.4e-3	0.122
	K	Λ_2	0.158	1.37e-3	0.11
	K	Λ_3	0.1532	1.32e-3	0.124

From the table it can be inferred that, as the value of Λ increases, overshoot and settling time decrease as expected. This is because, the sliding surface gain determines the error decay rate and in the equivalent controller, it introduces a velocity damping. But on the other hand, as the value of the sliding surface increases, an oscillation occurs at the point where the system reaches the sliding surface. For the above cases, for Λ_3 of the no load case, the chattering in the control signal is sustained but for loaded case the chattering vanishes after some time. Hence for the trajectory tracking case, the first sliding surface gain is used. The result of the trajectory tracking using the above gain and sliding surface slope is shown in Fig. 5.5, 5.6 and 5.7.

To compare its performance with respect to other controllers, a joint space PID controller and SMC in sliding mode controller are implemented. The PID controller parameters are GA

tuned optimal values as used in [6] and are given below. The joint space sliding mode controller gains and sliding surface slope are same as the task space one.

$$k_p = [7.25 \quad 2.54 \quad 3.45 \quad 7.46 \quad 2.54 \quad 3.15] \times 10^5 \quad (5.23)$$

$$k_d = [9.12 \quad 9.35 \quad 8.45 \quad 7.88 \quad 6.87 \quad 5.15] \times 10^4 \quad (5.24)$$

5.1.4.2 Discussion of results

From the figures one can observe that, the tracking performance of the new controller is better than joint space SMC not only in terms of the tracking accuracy but also in terms of the smoothness of the control signals. The results are better than the one given in [113], [132] and [133]. The comparison of the control performance of SMC in joint space, a PID controller and the new controller is given in Fig.5.6 and Fig.5.7.

Figures 5.6a-f shows the tracking performance of the three controllers for the given trajectory. In all cases the error is not asymptotically decaying but is bounded. In the case of the PID, the gains are big and its control force is very large but the error is bounded to only $\pm 15\text{mm}$. For the joint space SMC, the error is bounded within $\pm 5\text{mm}$ but for the task space sliding mode controller (TSMC) the error is very much smaller in all direction, $\pm 0.1\text{mm}$. Considering the speed of operation taken, which is 600mm/sec , this tracking error is very good. The performance of the controller for loaded conditions is also checked using a 200Kg load. The result is shown in Fig.5.7a and Fig.5.7b. From the figures it can be seen that the controller performance has not been degraded much as tracking error is bounded within $\pm 1\text{mm}$. The control force of the task space controller, after it is transformed in to joint space, is shown in Fig.5.8 and it is clear that the signal is not excessive and is also smooth without any chattering. Actually at the start of the trajectory, a small oscillation occurs and then the control force is very smooth.

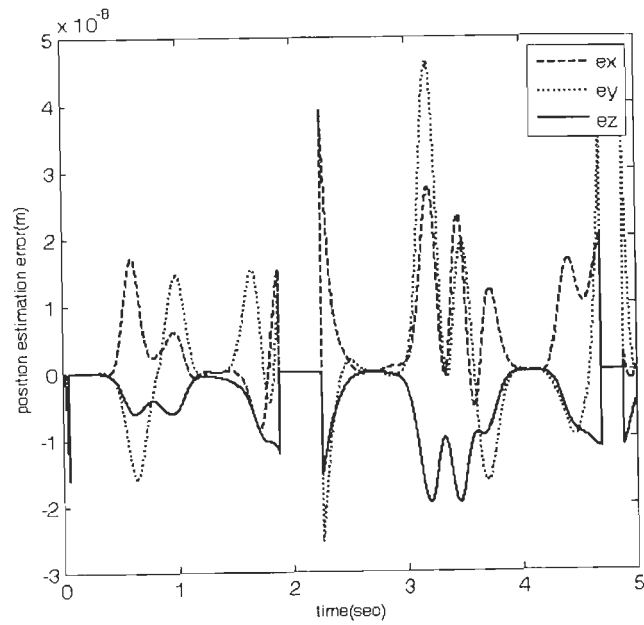


Fig. 5.4a) Estimation error of the task space positions using numerical algorithm

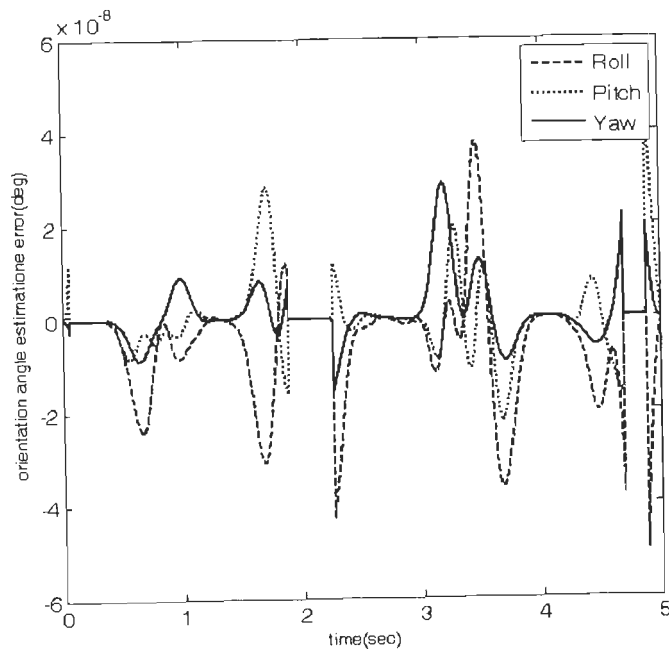


Fig. 5.4b) Estimation error of the task space orientation angles when using numerical algorithm only

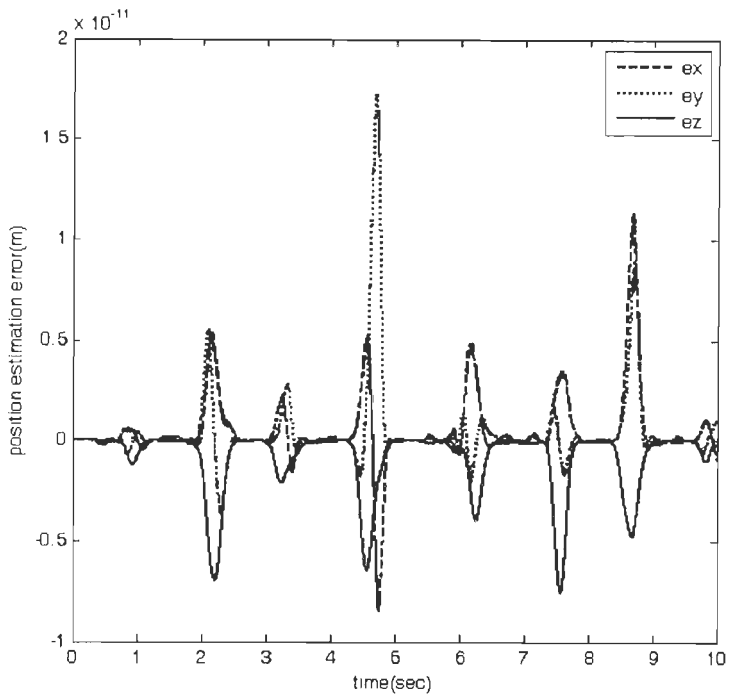


Fig. 5.4c) Position estimation error when neural network is used with numerical algorithm

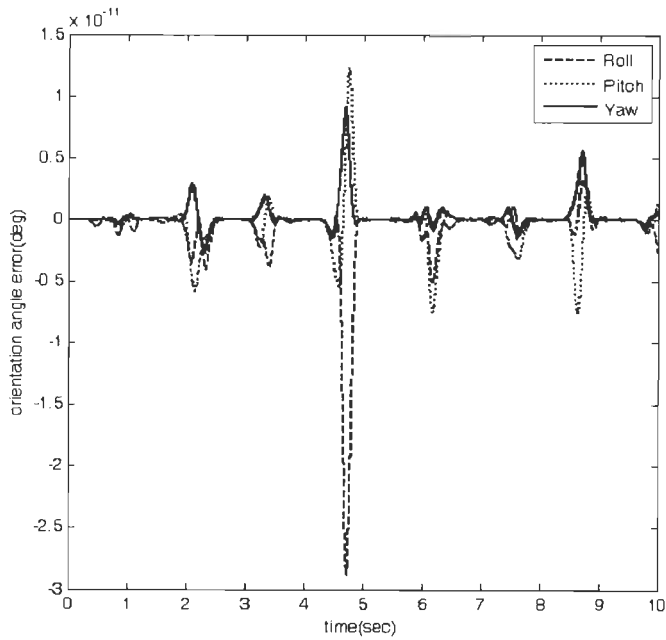


Fig. 5.4d) Orientation angle estimation error when neural network is used with numerical algorithm

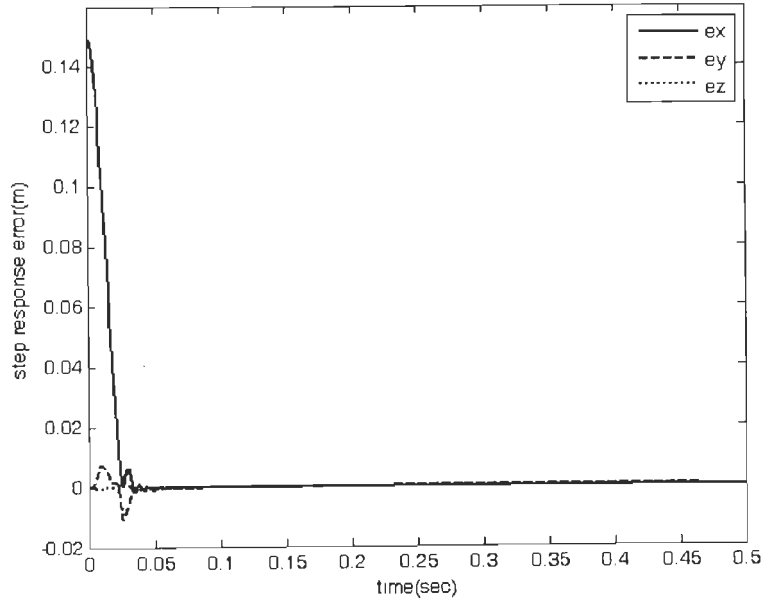


Fig. 5.5 Step response error in x direction used for parameter selection, desired is rise time less than 50msec with no overshoot

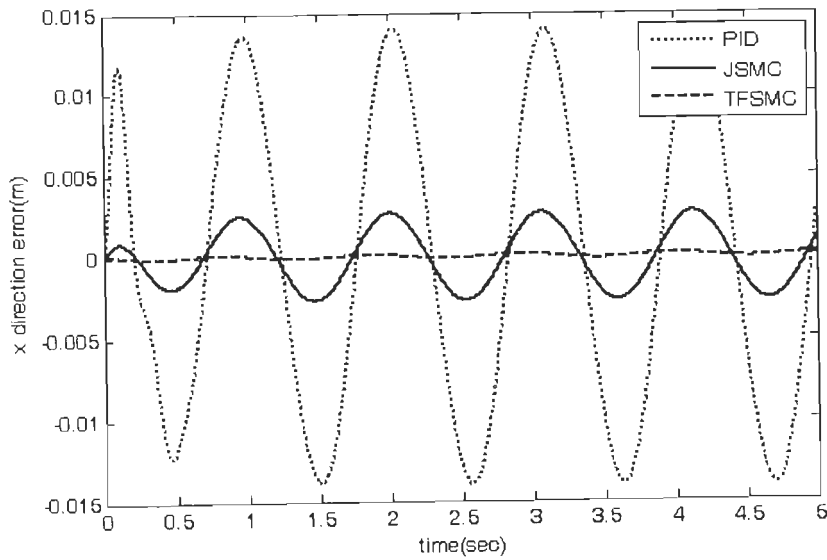


Fig.5.6a) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller in x direction

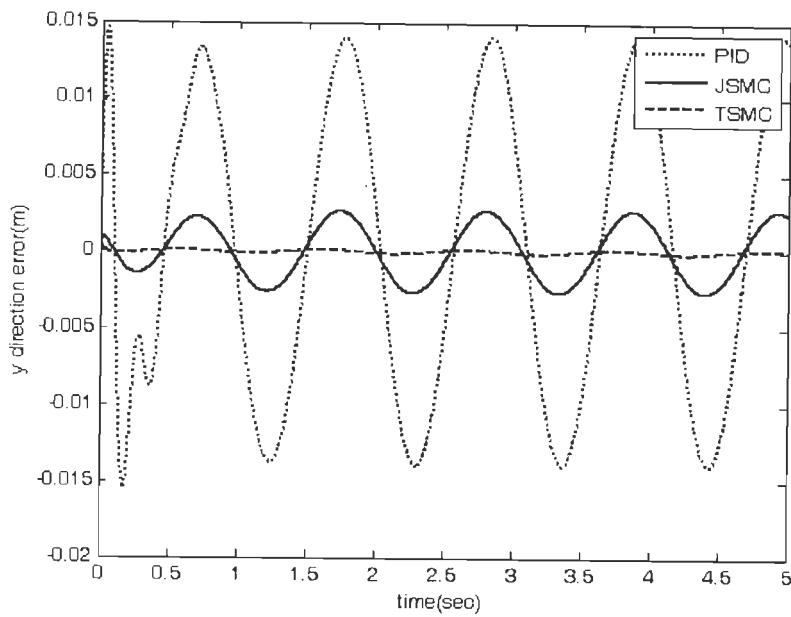


Fig.5.6b) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller in y direction

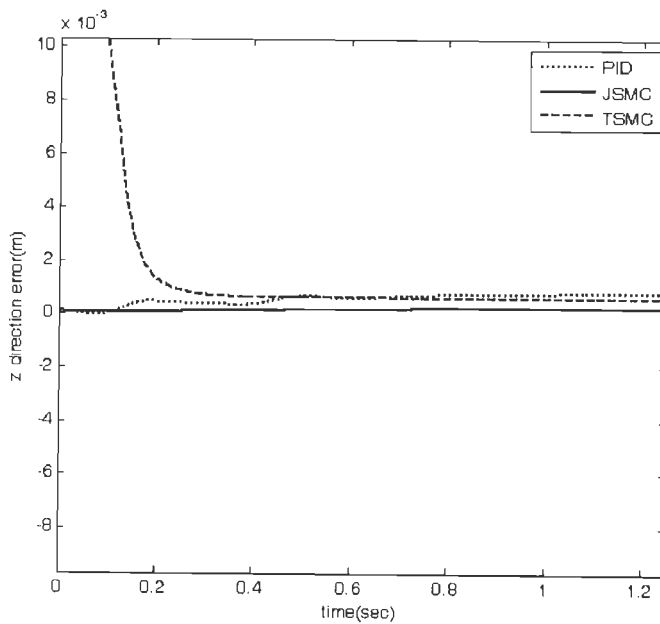


Fig.5.6c) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller in z direction

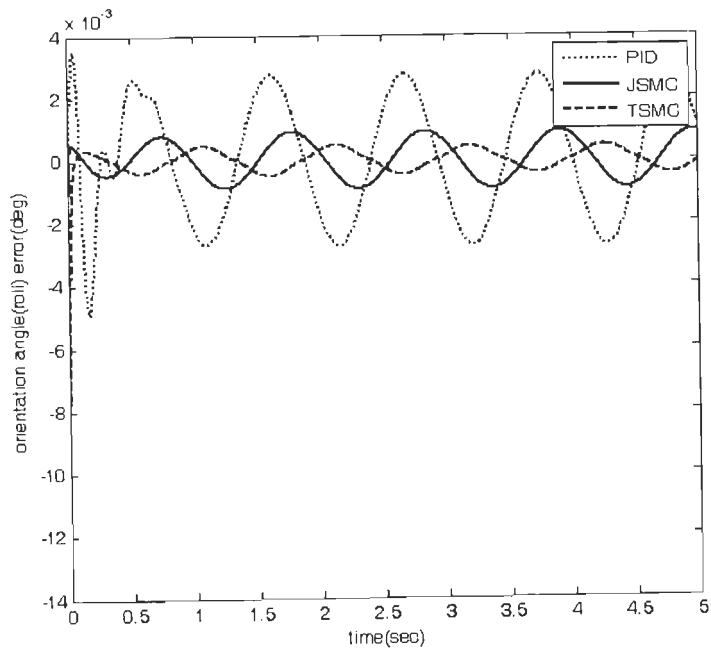


Fig.5.6d) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller, roll angle

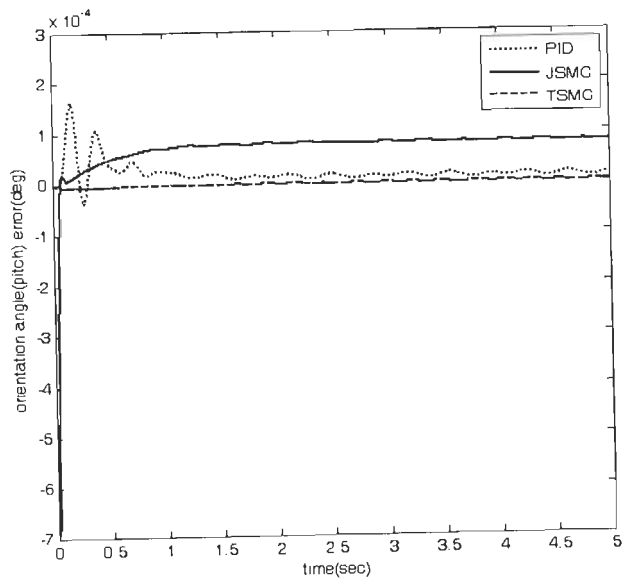


Fig.5.6e) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller, pitch angle

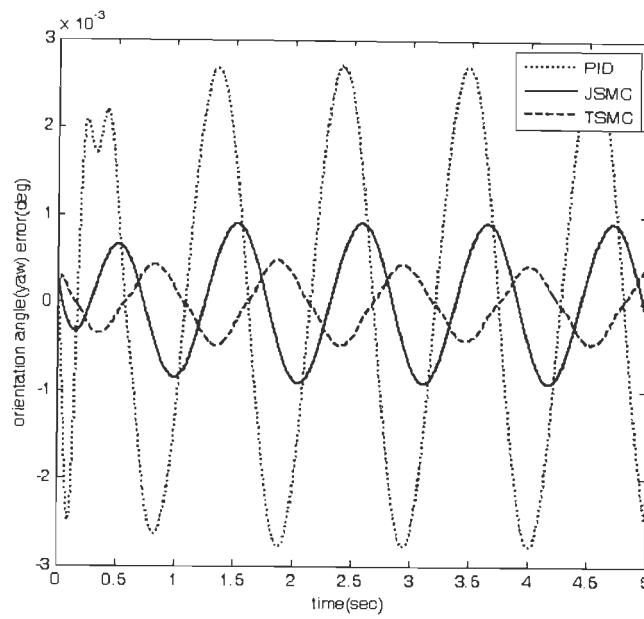


Fig.5.6f) No load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller, yaw angle

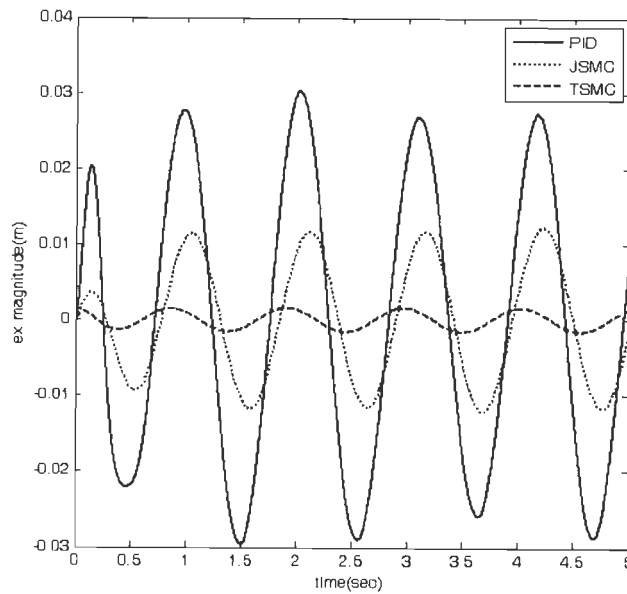


Fig.5.7a) Full load trajectory tracking performance of the task space sliding mode controller (TSMC), joint space sliding mode controller (JSMC) and simple PID controller, x direction

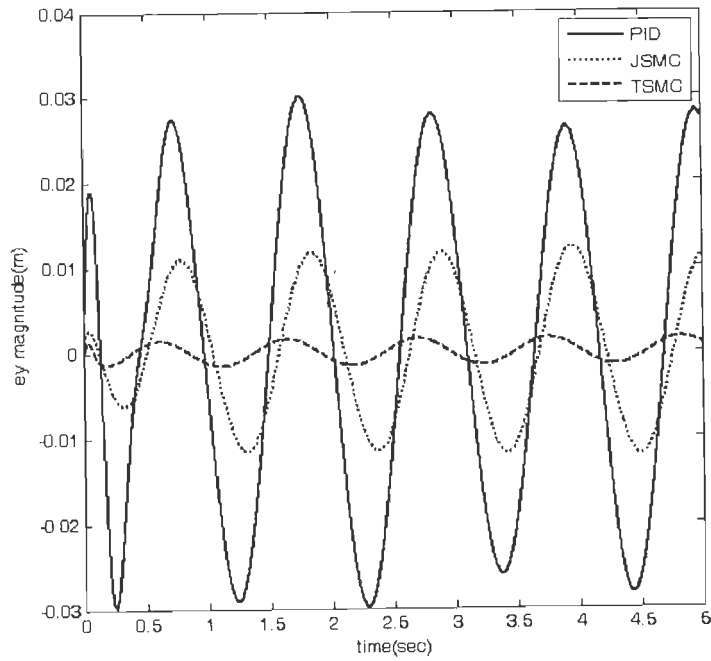


Fig.5.7b) Full load trajectory tracking performance of the three controllers in y direction

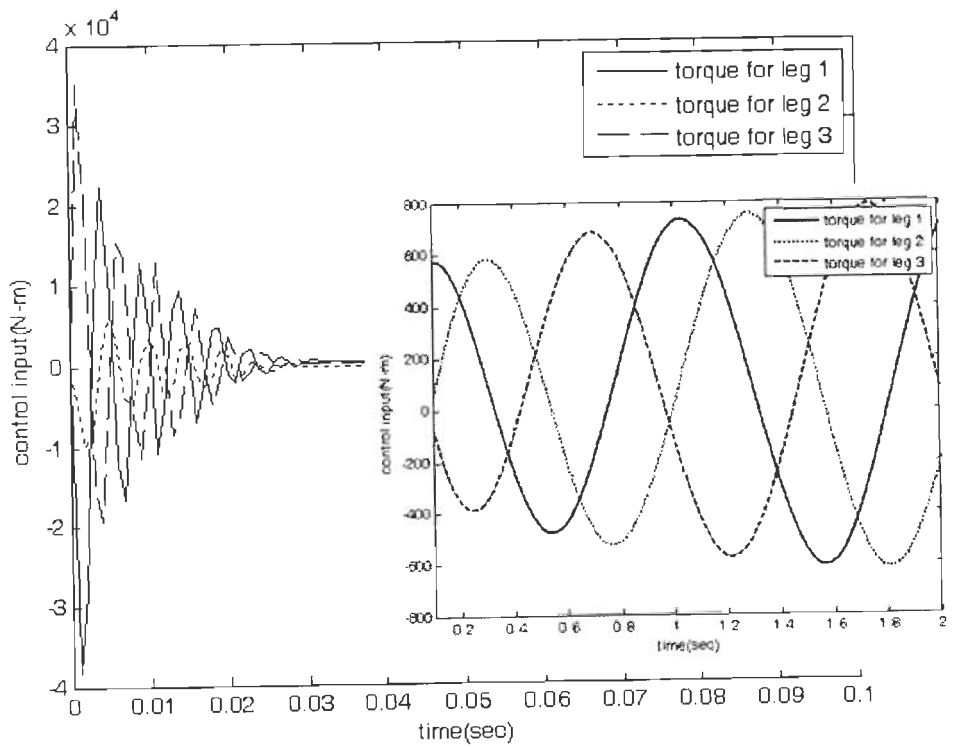


Fig.5.8 Control forces in joint space

5.2. FUZZY SLIDING MODE CONTROLLER WITH INTEGRAL LOOP (FSMCPI)

In the previous section, we have seen fuzzy sliding mode controller applied for Stewart platform control. Though the FSMC has shown good performance, it has some chattering in its control signal. In this section, we present our proposal which enhance performance of fuzzy sliding mode controller. The controller is designed to improve the chattering problem of fuzzy sliding mode controller. As in section one, the fuzzy logic is used as the switching device that generates the desired control signal to drive the system towards the sliding surface. An equivalent control signal composed of feed forward and feedback parts calculated using nominal system dynamics is used.

5.2.1 Source of chattering in fuzzy sliding mode control and proposed approach

Fuzzy sliding mode controllers in effect are similar to saturation control and are unable to restrict the system in the sliding surface [65]. In analyzing the source of chattering in sliding mode controllers, [65] showed that fuzzy sliding mode controllers can only result in continuous control signal. This is because source of chattering is not only because of the discontinuity of control signal but also due to lack of exact control effort which is able to keep the system to remain on sliding surface once it reached there. When the system is in the sliding surface, the output of the switching function is zero and if the control signal necessary to keep the system in the sliding surface is not correct, then the system will leave the surface and it may continue to oscillate within a small boundary. Hence to solve this problem and achieve a stable chatter free operation, we propose a new type of fuzzy sliding mode controller, viz fuzzy sliding mode controller with integral loop. The external PI loop acts as long term average of the distance variable and helps in keeping the system on the sliding surface.

The new controller retains the basic sliding mode control scheme and hence is robust. Moreover, the external integral loop together with the control signal calculated using the nominal dynamics of the platform helps to keep the system in the sliding surface. Therefore the controller achieves the desired high performance. Simulation results showed that the controller has a smooth and reasonable control signal with a very good tracking performance.

5.2.2 Design of the controller

The new controller proposed in this chapter is to use fuzzy sliding mode controller with PI

loop. The control signal is given by (5.25). The fuzzy switching logic given in section two is used with an integral loop to achieve a robust and chatter free controller for the Stewart platform. The block diagram of the FSMCIP is as shown below.

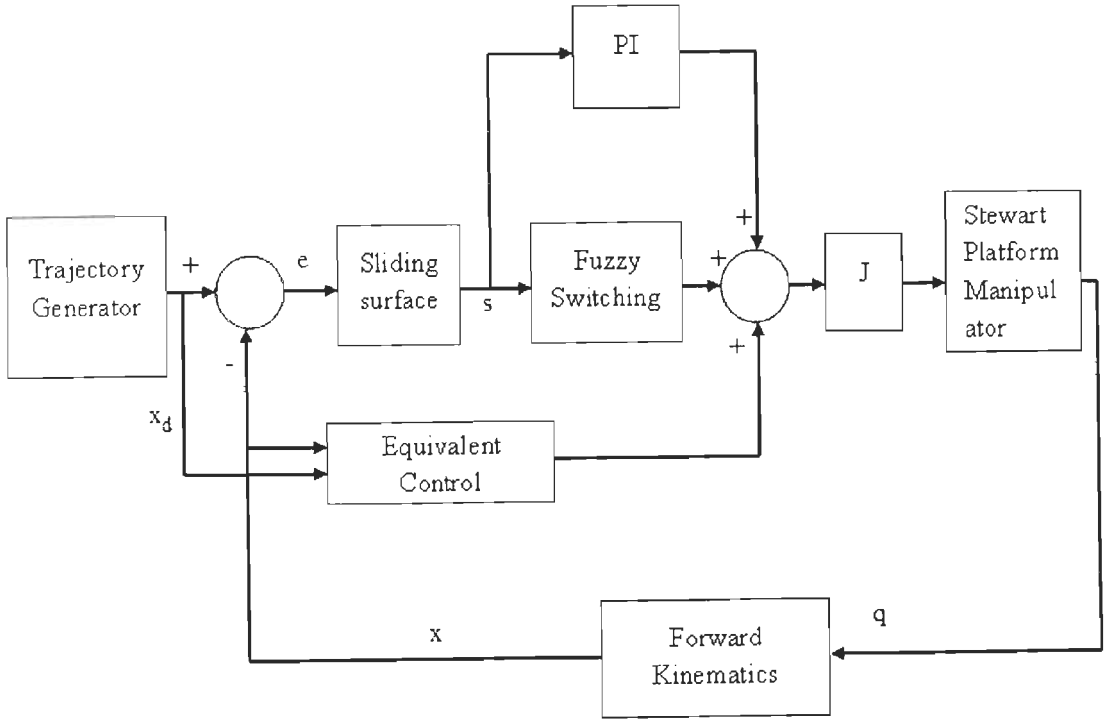


Fig.5.9 Block diagram representation of the control system

$$\tau = \tau_{eq} + \tau_s \quad (5.25)$$

where the equivalent controller τ_{eq} is given by (5.9) and

$$\tau_s = J^T \left(K_f(S) + K_1 \int S dt \right) \quad (5.26)$$

5.2.3. Simulation results and discussion

For the simulation study, the same type of desired trajectory used in previous section, (5.16) is employed. The same fuzzy switching logic is also implemented using MATLAB fuzzy logic tool box in Simulink and the other control algorithms are written as mfile. To check for the robustness of the controller against friction, static friction and viscous friction terms have been added to the SimMechanics model and the controller has shown robust performance. The x, y and z direction tracking performance of the fuzzy sliding mode controller (FSMC) and the proposed fuzzy sliding mode controller with integral loop (FSMCIP) is given in Fig.5.10-Fig.5.12. The slope of the sliding surface Λ , the gain of the switching function in

the SMC K and the integral gain constant of FSMCIP K_I used for the simulation are as follows.

$$\Lambda = \begin{bmatrix} 6 & 0 & \dots & 0 \\ 0 & 6 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 6 \end{bmatrix} \quad (5.27)$$

$$K = 1000 \cdot \begin{bmatrix} 6 & 0 & \dots & 0 \\ 0 & 6 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 6 \end{bmatrix} \quad (5.28)$$

$$K_I = \text{diag}[5000 \quad 5000 \quad 5000 \quad 5000 \quad 5000 \quad 5000] \quad (5.29)$$

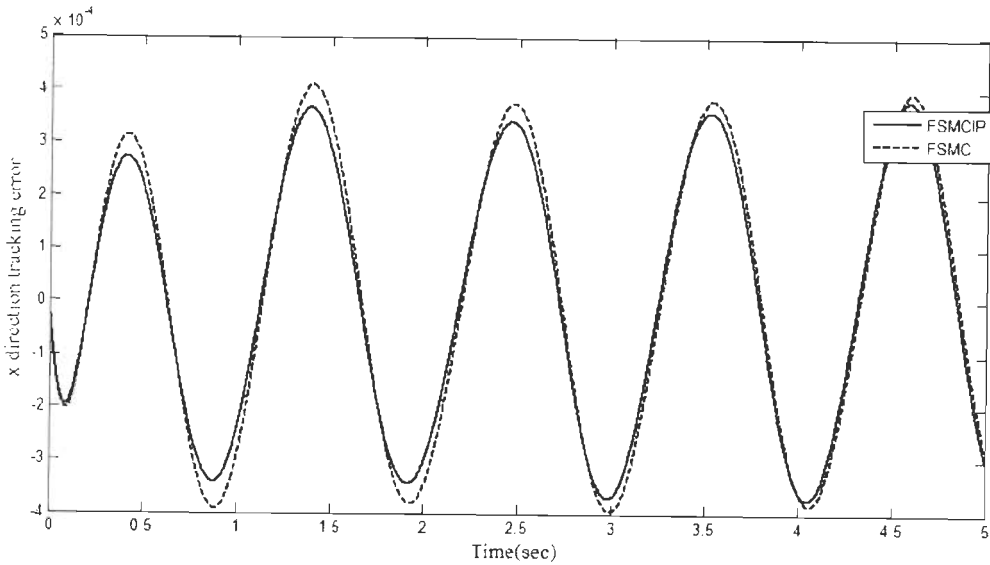


Fig.5.10 Comparison of x direction tracking performance of FSMC and the new controller, FSMCIP

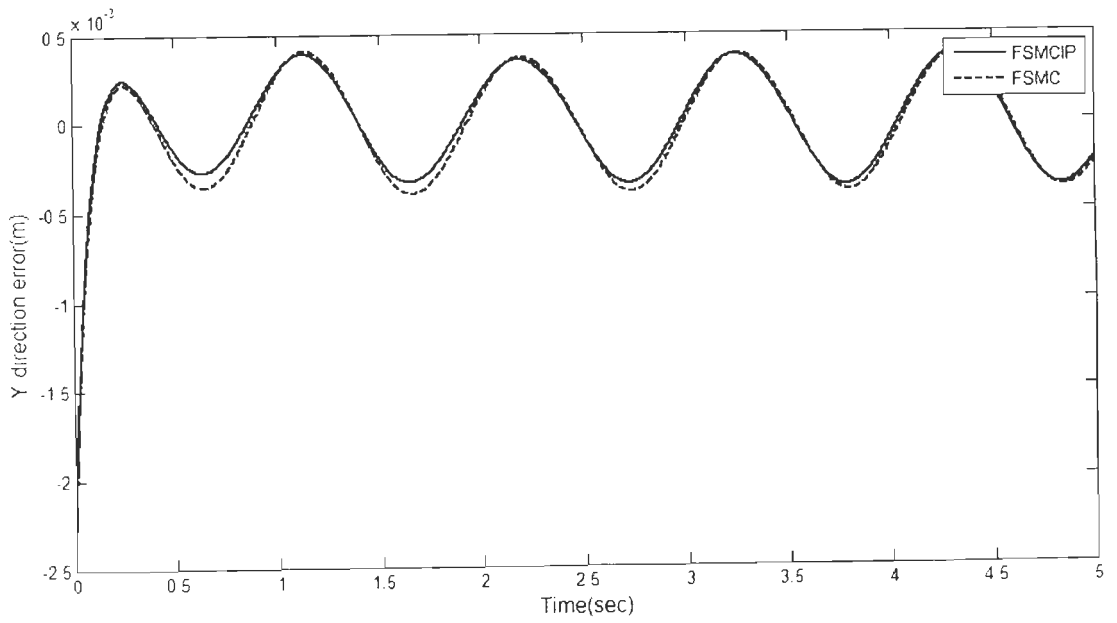


Fig.5.11 Comparison of y direction tracking performance of FSMCIP with fussy SMC

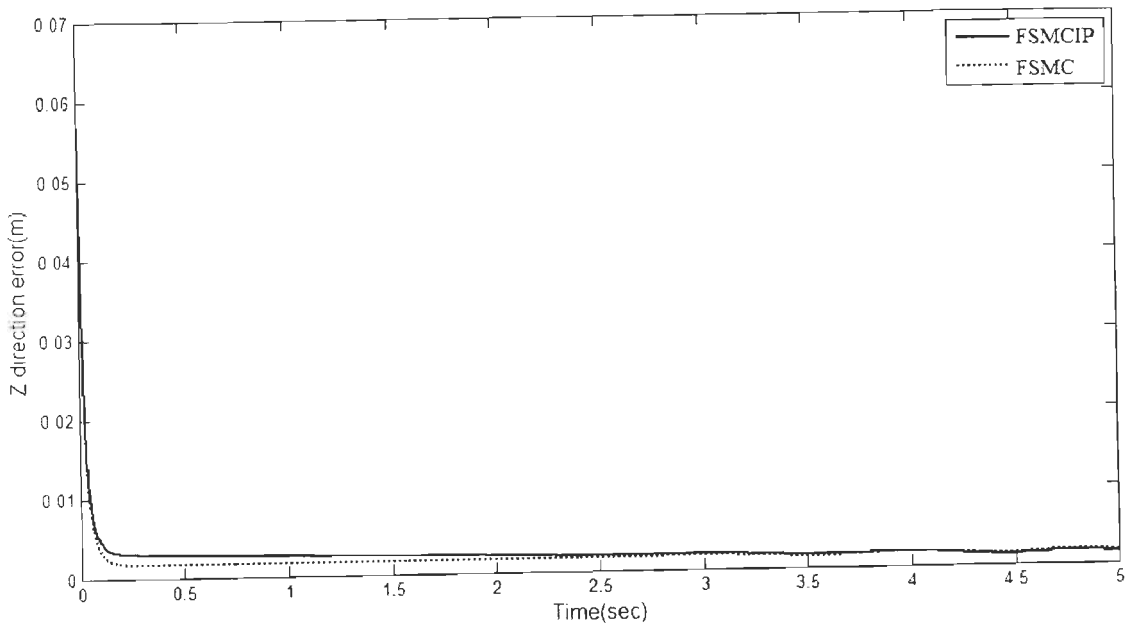


Fig.5.12 Comparison of Tracking performance in Z direction of new controller (FSMCIP) and fuzzy SMC

From the figures, one can observe that the tracking performance of the new controller, FSMCIP, is slightly better than FSMC in terms of the tracking accuracy. However, the most important advantage of the new controller is observed when the manipulator is driven in a helical trajectory given in chapter 4 by (4.6)-(4.7). The trajectory is redrawn in Fig.5.13 for ease of reference. The control signals of the two controllers when the manipulator is driven in this trajectory carrying a load of 200Kg are plotted separately in Fig.5.14 and Fig.5.15. The FSMC has bigger control signals with more chattering which cannot be generated by a real actuator.

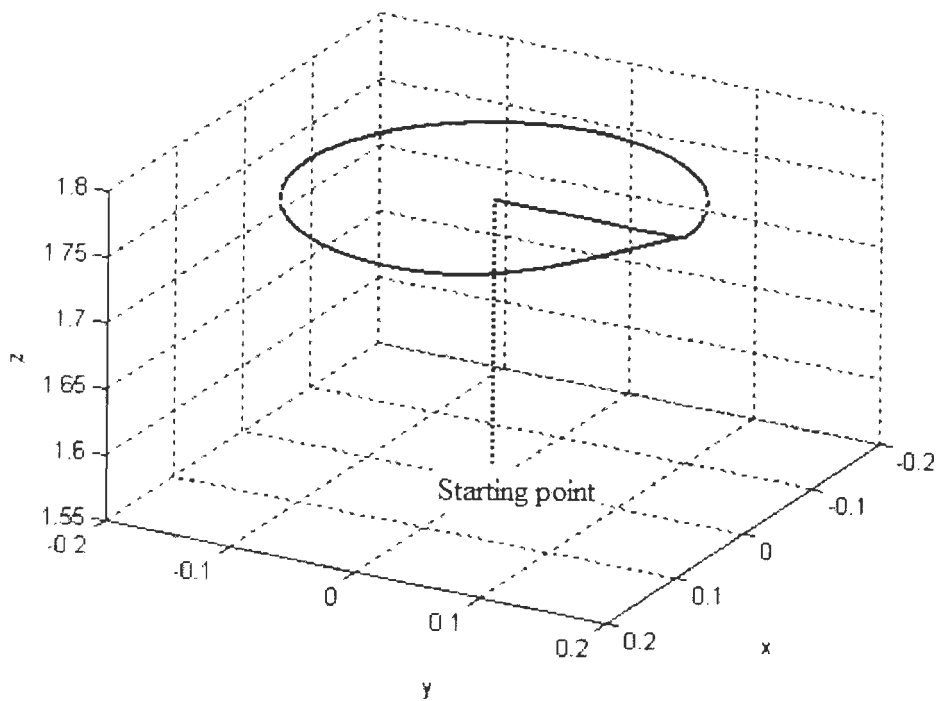


Fig.5.13 Three dimensional view of the helical trajectory used to compare the performance of FSMC and FSMCIP

From the simulation results it can be seen that the FSMCIP has shown a better performance in tracking desired trajectory than the simple fuzzy SMC. In the simulation, it was further noticed that the control system needs a small simulation time, fixed simulation time of 0.1msec. With this simulation time, the calculated velocity and acceleration values for the equivalent control part were initially very big. This may have some constraint on practical implementation.

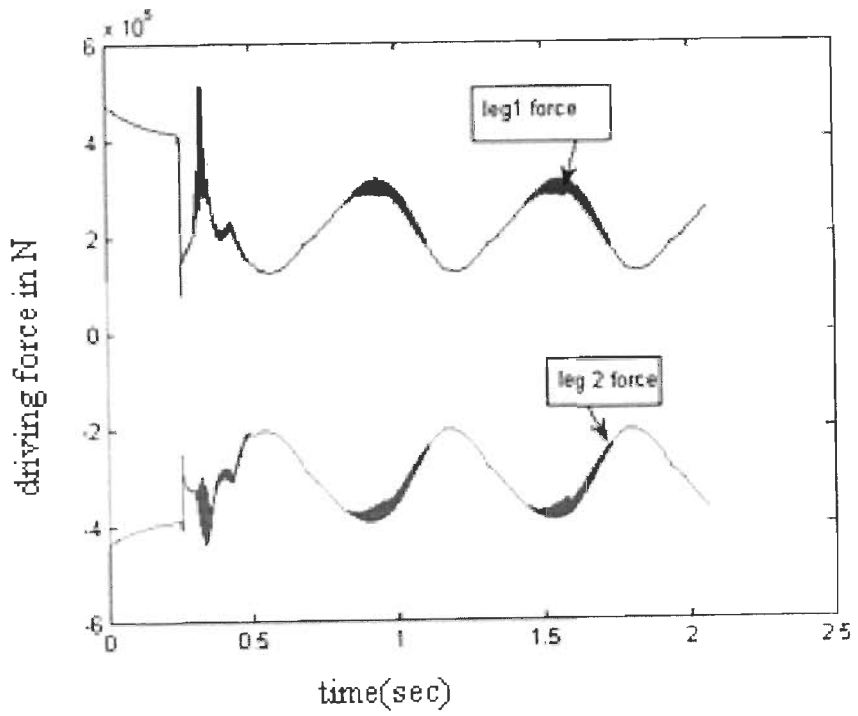


Fig. 5.14a) Control signal of SMC for leg 1 and 2 when carrying 200Kg payload

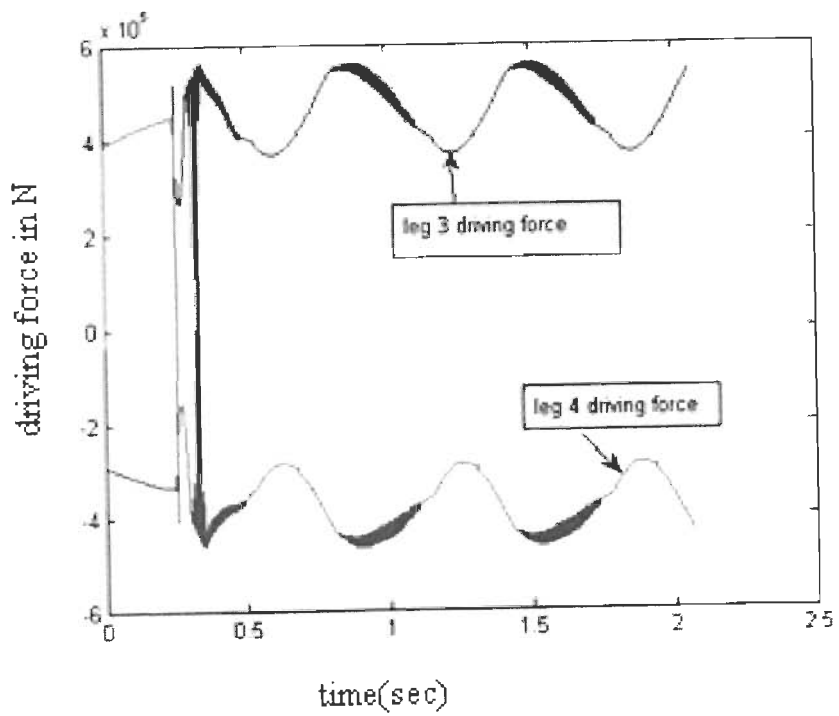


Fig. 5.14b) Control signal of FSMC for leg 3 and 4 when carrying 200Kg payload

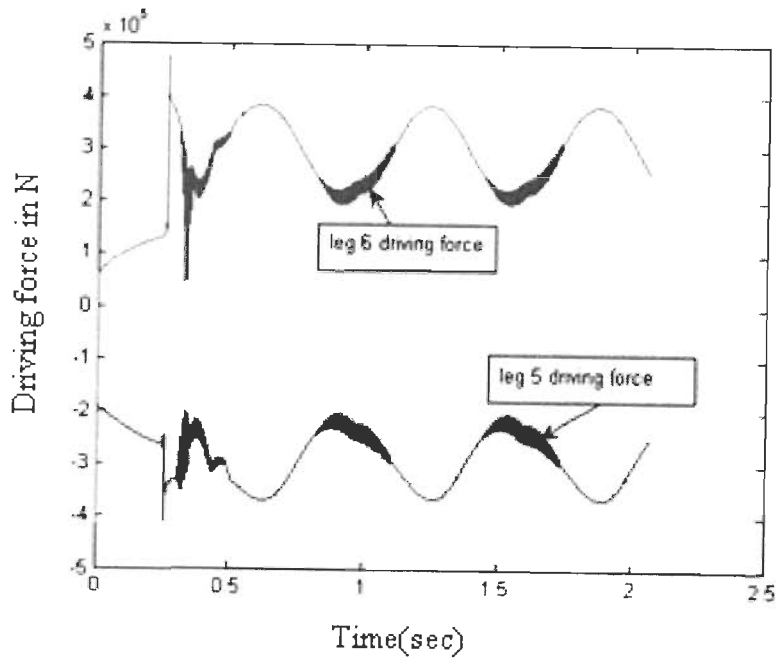


Fig. 5.14c) Control signal of FSMC for leg 5 and 6 when carrying 200Kg payload

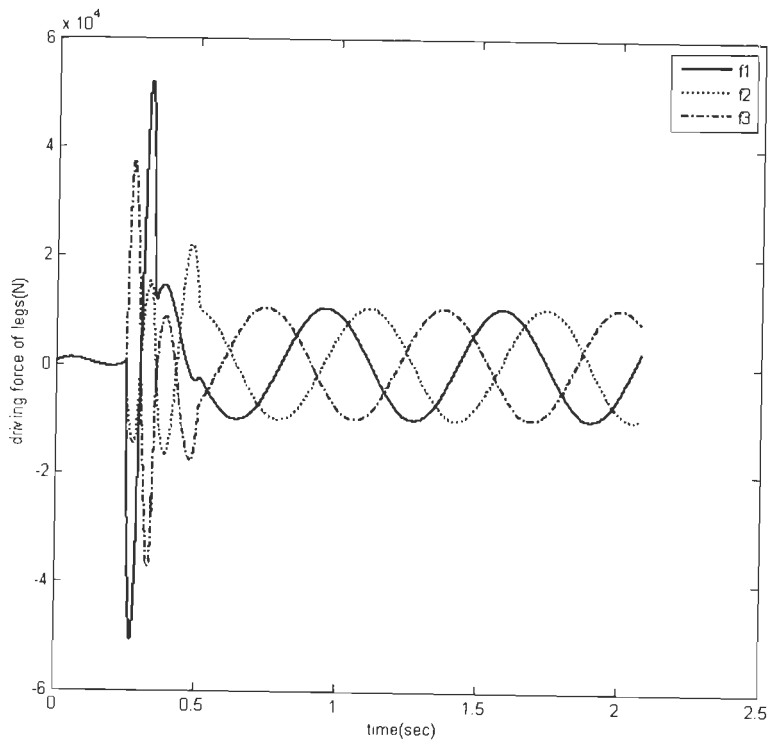


Fig.5.15a) Control force on leg 1, 2, 3 of the FSMCIP when carrying payload of 200Kg

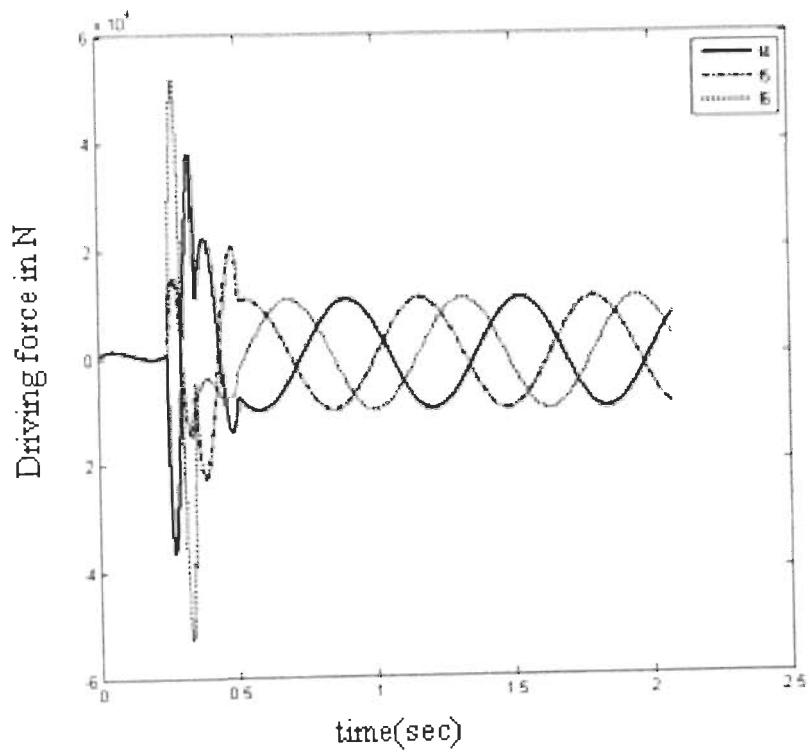


Fig. 5.15b) control force on leg 4, 5, 6 of the FSMCIP when carrying a 200Kg payload

5.3. Hybrid sliding mode control with a newly proposed sliding surface to reduce synchronization error

In the above section, we have seen two different task space sliding mode controllers and discussed that their performance is good. However, task space approach needs 6DOF end effector pose, which is either very costly or computationally intensive [58]. To solve the above mentioned problem, many researchers have proposed joint space feedback control methods improved by using feedforward loops. In [177], a tracking differentiator is used as a feedforward controller to improve joint space PID controller performance. In [36], an adaptive controller having feedback and feedforward parts is designed to compensate for synchronization error of joint space control. But the above schemes cannot achieve high performance, especially at high speeds because the linear PD or PID controllers cannot compensate leg dynamics. To improve drawbacks of joint space, other authors have proposed model based control methods which utilize advantages of task space approach. In [89], the author proposed a control method for an electrically driven manipulator where the desired force is calculated in task space using desired position and velocity signals and the leg

dynamics compensation is achieved by a force convergent principle. The authors of [131] proposed a hybrid control algorithm where the platform dynamics is calculated in task space and the leg dynamics is compensated in joint space by calculating leg dynamics using Newton Euler method. But the method needs task space positions and orientations and the authors did not specify the method used to get the pose of the end effector. Similarly, the authors of [145] used inverse dynamics control with approximate dynamics as a feedforward controller and they employed a linear H-infinity controller in the feedback path to compensate for the approximation error. The last two proposals are promising since they are robust and can achieve high performance due to the model based feedforward part.

With this background, a new kind of hybrid controller based on sliding mode control is proposed. The control scheme presented in this section has similar structure to that of [89][131][145] in the sense that it combines task space feedforward with joint space feedback. But the controller presented here has three basic differences. The first difference is the feedforward controller part, which is implemented in task space, is an equivalent control signal of a sliding mode controller. SMC law is formulated in joint space but the model based equivalent control part is transformed to task space because of the ease in computation of model parameters in task space. The second main difference is the nonlinear controller used in the joint space. In the current controller, the uncertainty in the manipulator leg dynamics is compensated by using sliding mode control, which is a robust and easier control technique [34]. Because of its robustness against uncertainties, SMC has been proposed for the control of various systems including Stewart platform manipulator [30][113][133].

The third and other most important advantage of the controller proposed in this section is, it solves the problem of synchronization using a new sliding surface. The joint space sliding mode controller used here has a newly proposed sliding surface that considers tracking error and synchronization error. It is a linear sliding surface, which is designed to drive both errors to zero. This new sliding surface improves the task space trajectory tracking performance of the controller and reduces the problems of joint space control. The structure of the controller is as shown in Fig 5.16.

Unlike conventional sliding mode controllers, the joint space sliding mode controller uses information from other legs through the synchronization error and hence solves the synchronization problem. Moreover, the sliding mode controller improves the reliability of

the controller to track task space trajectory.

5.3.1 Design of the hybrid sliding mode controller

The task space dynamic model given in (5.1) can be rewritten by including actuator friction and writing the platform and leg dynamic parameters separately as in (5.30) below.

$$\left(M_1(X) + M_2(X)\right)\ddot{X} + \left(C_1(X, \dot{X}) + C_2(X, \dot{X})\right)\dot{X} + G_1(X) + G_2(X) = J_p^{-T} \left(\tau - f_f\right) \quad (5.30)$$

In the above equation, $M_1(X)$, $C_1(X, \dot{X})$ and $G_1(X)$ are inertia matrix, coriolis/centrifugal coefficient matrix and gravitational torque of the platform, $M_2(X)$, $C_2(X, \dot{X})$ and $G_2(X)$ are corresponding parameters of the legs. The dynamic equation can be converted to joint space using the following transformations.

$$\dot{X} = J^{-1}\dot{q} \quad (5.31)$$

$$\ddot{X} = J^{-1}\ddot{q} + \dot{J}^{-1}\dot{q} \quad (5.32)$$

And then the joint space dynamic model becomes

$$A(q)\ddot{q} + B(q, \dot{q})\dot{q} + Q = \tau - f_f \quad (5.33)$$

Where the inertia, corioles centrifugal and gravitational torque matrix and vectors are related by

$$A(q) = J^T (M_1 + M_2) J^{-T} \quad (5.34)$$

$$B(q, \dot{q}) = J^T \left((C_1 + C_2) J^{-T} + (M_1 + M_2) \dot{J}^{-T} \right) \quad (5.35)$$

$$Q = J^T G \quad (5.36)$$

The joint space leg length error vector is given by

$$e = q_d - q. \quad (5.37)$$

where q_d is the desired leg length and q is measured leg length

If a non zero coupling factors c_i such that

$$c_1 e_1 = c_2 e_2 = c_3 e_3 = c_4 e_4 = c_5 e_5 = c_6 e_6 \quad (5.38)$$

is present, then the synchronization error is defined by[36]

$$\varepsilon_i = c_i e_i - c_{i+1} e_{i+1} \quad (5.39)$$

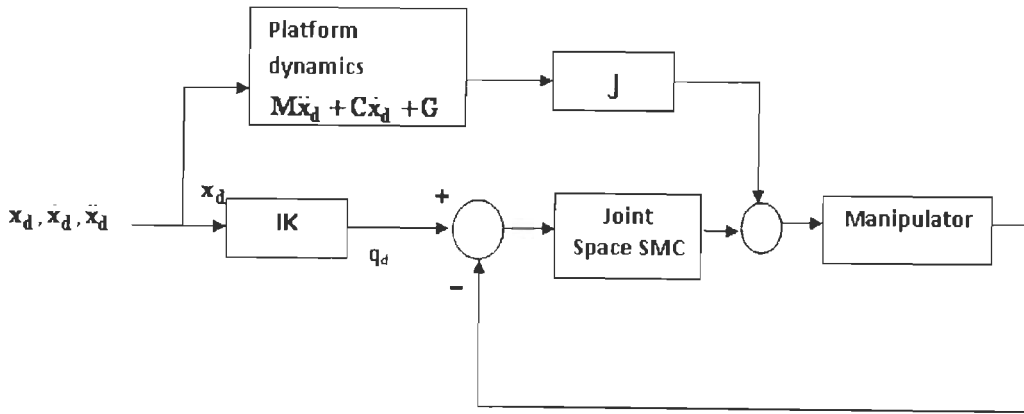


Fig.5.16 Task space feedforward and joint space feedback with SMC control

Combining the tracking error and synchronization errors, a single variable referred as cross coupling error is obtained and it is defined as [36]

$$\chi_{ci} = c_i e_i + \mu \int_0^t (e_i - e_{i-1}) dt \quad (5.40)$$

for $i=1,2,\dots, 6$ and $i-1=6$ when $i=1$ and μ is coupling parameter

In vector form, (5.40) can be represented as

$$\chi_e = Ce + \mu \int f(e) \quad (5.41)$$

Using this cross coupling and the tracking error, the error dynamics of the Gough-Stewart platform manipulator is rewritten as a system with three state variables at each leg and is given as

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= \ddot{q}_d - A(q)^{-1} (\tau - f_f - B(q, \dot{q}) - Q(q)) \\ \dot{X}_3 &= CX_2 + \mu f(X_1) \end{aligned} \quad (5.42)$$

where the states X_1 and X_2 are the tracking error e and its derivative \dot{e} and X_3 is the cross coupling error χ_e .

In sliding mode control, there are two basic steps and the first step is the design of a stable sliding surface which is formed from the states of a system to be controlled and the second step is the design of control law which will drive the system states towards the stable sliding surface. The sliding surface has to be designed such that when the states are on the sliding surface, the system has to be insensitive to external disturbances and the states should move

towards an equilibrium point. In the present case, the system dynamics (5.42) is formulated in terms of tracking error and synchronization error. If a new sliding surface containing the three states is proposed so that when the system is on that sliding surface, both synchronization error and tracking error asymptotically move to zero, then we can drive both synchronization error and tracking error to zero by using an appropriate control law. Hence, the following new sliding surface, which is used to constrain tracking error and synchronization error, is defined for joint space sliding mode control

$$S = \Lambda_1 X_1 + \Lambda_2 X_2 + \Lambda_3 X_3 \quad (5.43)$$

The matrices Λ_1 , Λ_2 and Λ_3 are 6x6 and without loss of generality, Λ_2 is assumed to be identity matrix. From (5.43) it is clear that, when $s=0$, the sliding surface is a plane in 3D which passes through the origin and if a control signal is designed properly, both the tracking error and synchronization error can be driven to zero.

Taking the derivative of S and equating it to zero, the following equivalent controller is obtained,

$$\tau = A(q)(\ddot{q}_d + \Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1))) + B(q, \dot{q})\dot{q} + Q(q) \quad (5.44)$$

Substituting (5.44) into (5.42) and assuming that the dynamic parameters are computed exactly, the sliding dynamics of the system becomes

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= -\Lambda_1 X_2 - \Lambda_3 (CX_1 + \mu f(X_1)) - A(q)^{-1} f_f \\ \dot{X}_3 &= CX_2 + \mu f(X_1) \end{aligned} \quad (5.45)$$

If friction is neglected, the sliding dynamics is a second order ordinary differential equation and both the tracking error and the synchronization error can be made to zero and asymptotic stability can be achieved. But in practical systems, friction is a critical factor and it cannot be neglected. Hence the sliding dynamics cannot achieve asymptotic stability but the tracking error and synchronization errors can be made to be bounded within a tolerable limit. Another point to be noted is, the parameters of the equivalent controller (5.44) are in joint space and it has been reported by various researchers that computation of these model parameters in joint space is difficult and needs complex coordinate transformations [173]. Using the transformations given in 5.31 and 5.32, the task space equivalent of these model parameters can be obtained and the equivalent control signal can be calculated in task space as shown below.

$$\tau = J^T (M_1 + M_2) J^{-T} (\ddot{q}_d + \Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1))) + J^T ((C_1 + C_2) J^{-T} + (M_1 + M_2) \dot{J}^{-T}) \dot{q} + J^T G \quad (5.46)$$

This can be rewritten again as

$$\tau = J^T \tau_f + J^T (M_1 + M_2) J^{-T} (\Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1))) \quad (5.47)$$

Where

$$\tau_f = (M_1 + M_2) (J^{-T} \ddot{q}_d + \dot{J}^{-T} \dot{q}) + (C_1 + C_2) J^{-T} \dot{q} + G_1 + G_2 \quad (5.48)$$

If \dot{q} is replaced by \dot{q}_d , τ_f can be rewritten as

$$\tau_f = (M_1 + M_2) \ddot{X}_d + (C_1 + C_2) \dot{X}_d + G_1 + G_2 \quad (5.49)$$

and it can be easily calculated in task space without the need for estimating of the pose of the end effector. Replacing \dot{q} by \dot{q}_d has certain effects on the sliding dynamics and the effect of the change and presence of other uncertainties is analyzed in the next section.

5.3.2 Robustness analysis

The uncertainties in the Gough-Stewart manipulator include parameter variation, actuator friction and backlash. The uncertainties are assumed to be bounded and parameter uncertainties are assumed to be additive so that the parameters are given as nominal and deviation. Therefore, the following assumption is used.

Assumption: The uncertainties in the inertia, Coriolis and centrifugal and gravitational matrices are bounded and can be given as nominal and deviation as:

$$M_1 = M_{1N} + \Delta M_1$$

$$M_2 = M_{2N} + \Delta M_2$$

$$C_1 = C_{1N} + \Delta C_1$$

$$C_2 = C_{2N} + \Delta C_2$$

$$G_1 = G_{1N} + \Delta G$$

$$G_2 = G_{2N} + \Delta G$$

Using this assumption, the uncertain dynamic model of (5.30) can be rewritten as

$$(M_{1N}(X) + M_{2N}(X)) \ddot{X} + (C_{1N}(X, \dot{X}) + C_{2N}(X, \dot{X})) \dot{X} + G_{1N}(X) + G_{2N}(X) = J_p^{-T} (\tau - f_f) - d(X) \quad (5.50)$$

Where d is parametric uncertainty given by

$$d(X) = (\Delta M_1 + \Delta M_2)\ddot{X} + (\Delta C_1 + \Delta C_2)\dot{X} + \Delta G_1 + \Delta G_2 \quad (5.51)$$

Given the nonsingular and positive definiteness assumption on the mass matrix and assuming that the manipulator Jacobian J_p to be also nonsingular over the work space, the control law (5.52) is proposed to stabilize the uncertain dynamic system (5.50), i.e. the controller is able to drive the system towards the sliding surface in a finite time and will keep the system states on the sliding surface with the tracking and synchronization errors decaying to zero asymptotically.

$$\tau = J^T \tau_{IN} + J^T (M_{1N} + M_{2N}) J^{-T} (\Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1))) + J^T (K f_s(s)) \quad (5.52)$$

where $f_s(s)$ is switching control signal and τ_{IN} is the task space equivalent controller calculated using nominal parameters. To analyze the stability of the system under the given controller, the control signal has to be rewritten in joint space as the errors are in joint space. Using the conversion matrices given in (5.31) and (5.32), the control signal can be written as

$$\tau = D_N(q)\ddot{q}_d + B_N(q, \dot{q})\dot{q}_d + Q_N(q) + D_N(q)(\Lambda_1 X_2 + \Lambda_3 (CX_1 + \mu f(X_1))) + K_f(s) \quad (5.53)$$

Taking the Lyapunov function

$$V = \frac{1}{2} s^T s \quad (5.54)$$

Taking the derivative of V

$$\dot{V} = S \dot{S} \quad (5.55)$$

and \dot{S} is calculated from (5.43) as

$$\dot{S} = \Lambda_1 \dot{X}_1 + \Lambda_2 \dot{X}_2 + \Lambda_3 \dot{X}_3 \quad (5.55)$$

and substituting for the state derivatives from (5.45)

$$\dot{S} = \Lambda_1 X_2 - \Lambda_2 (\ddot{q}_d - \ddot{q}) + \Lambda_3 (CX_2 + \mu f(X_1)) \quad (5.56)$$

Substituting for \ddot{q} from (5.33) into (5.56) and using the result into (5.55)

$$\dot{V} = S \left(\Lambda_2 (\ddot{q}_d - A^{-1}(\tau - f_f - B - Q)) + \Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1)) \right) \quad (5.57)$$

Then substituting the control signal from (5.52)

$$\dot{V} = S \left(\begin{array}{l} \ddot{q}_d - A^{-1} (A_N(q)\ddot{q}_d + V_N(q, \dot{q})\dot{q}_d + Q_N(q)) + A^{-1} (A_N(q)(\Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1)))) \\ K_f(S) - f_f - V - Q + \Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1)) \end{array} \right) \quad (5.58)$$

$$\dot{V} = S \left(\left(\ddot{q}_d + \Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1)) \right) \left(I - A^{-1} A_N(q) \right) + A^{-1} (\Delta V + \Delta Q) + A^{-1} (K_f(S) - f_f) \right) \quad (5.59)$$

In (5.59) the matrix A is the uncertain inertia matrix which can be written as nominal and deviation, as

$A = A_N + \Delta A$, and using the Sherman-Morrison formula, the inverse of A can be written as

$$A^{-1} = (A_N + \Delta A)^{-1} = A_N^{-1} - \frac{1}{1+g} A_N^{-1} \Delta A A_N^{-1} \quad (5.60)$$

where g is $\text{tr}(A_N \Delta A)^{-1}$. Then, using (5.60) into (5.59), we can rewrite

$$\dot{V} = -AS \left(A \left(\ddot{q}_d + \Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1)) \right) \left(\frac{1}{1+g} A_N^{-1} \Delta A \right) + \Delta V + \Delta Q + K_f(S) - f_f \right) \quad (5.61)$$

Hence the system can be stable and the sliding surface is attractive if the gain of the switching function is selected as

$$K \geq \|\Delta V\| + \|\Delta Q\| + \|f_f\| + \|h\| \quad (5.62)$$

$$h = A \left(\ddot{q}_d + \Lambda_1 X_2 + \Lambda_3 (CX_2 + \mu f(X_1)) \right) \left(-\frac{1}{1+g} A_N^{-1} \Delta A \right) \quad (5.63)$$

The term h is due to the parameter uncertainty in the inertia matrix, which is mainly due to position of end effector and payload. To avoid the need for excessively high value of gain, which main excite high frequency oscillation or vibration of the platform, friction torque can be separately compensated.

5.3.3. Simulation results and discussion

For the simulation study of the performance of the controller, the Gough-Stewart platform with the geometric parameters given in table I of chapter 3 is implemented using simmechanics tool box of MATLAB. A friction model containing viscous friction and coulomb friction, which is used in most robotic controllers, is included into the simmechanics model to simulate the effect of actuator friction. The model equation is given as

$$f_f = k_v \dot{q} + k_c \text{sign}(\dot{q}) \quad (5.64)$$

and also a random disturbance torque is added to simulate other external disturbance effects. Actuator dynamics is neglected assuming that electric motors with current/torque control are to be used. It is to be reminded that the practical performance of sliding mode controller is

also affected by back lash effects. Therefore, a nonlinear backlash simulator is connected in series with the joint actuator to test the chattering that may occur when the sliding mode controller is implemented digitally. To test for robustness against load variation, the controller is allowed to work with payload variations from no load to 200Kg.

The trajectory considered contains translational motions of heave, surge and sway including rotational motions of roll and yaw. The most important thing to be considered in the trajectory is its speed. It is a fast trajectory where the platform moves heave at (400mm/1.98Hz), surge and sway at (150mm/5.9Hz). The rotational motions are also fast with rate of roll and yaw motions at (10°/2.7Hz, 10°/2.3Hz) and the maximum desired acceleration is 10g. The trajectories are designed to achieve zero velocity and acceleration initially and are as give in (5.16) and repeated here for convenience.

$$\begin{aligned}
 x(t) &= 0.10 \left\{ 1 - \exp\left(-\frac{3\pi t}{10}\right) \right\} \cos(1.88\pi t), m \\
 y(t) &= 0.10 \left\{ 1 - \exp\left(-\frac{3\pi t}{10}\right) \right\} \sin(1.88\pi t), m \\
 z(t) &= 1.563 + \frac{0.15}{1+0.9t} \sin \left\{ 0.2\pi t \left(\frac{0.1+5.9t}{10.5} \right) + \frac{\pi}{24} \right\}, m \\
 \alpha(t) &= 0.15 \left\{ 1 - \exp(-\pi t) \right\} \sin(0.86\pi t), rad \\
 \beta(t) &= 0, deg \\
 \gamma(t) &= 0.15 \left\{ 1 - \exp(-\pi t) \right\} \sin(0.74\pi t), rad
 \end{aligned}$$

To compare the performance of the controller with existing controllers, a standard PID controller, with its parameters tuned using genetic algorithm, is implemented. Moreover, a simple sliding mode controller, which has been suggested by many authors and contains joint error and error rate, is also implemented and compared with the new controller. The simulation results are shown in fig. 5.17-5.33. The performance of the controllers for no load case is shown in figs. 5.17-5.22 while the figs.5.23-5.29 show the performance of the controllers when the platform is caring full load of 200Kg and friction is considered. The friction force considered is given by (5.64) with $k_v=50$ and $k_c=20$ and a disturbance torque taken from a uniform random distribution with maximum value of 50NM is also added to each leg. To see the performance of the controllers in task space, their performance is compared in task space by using a numerical forward kinematic estimation algorithm which is not used for control but only for comparison of outputs.

The controller parameters used and summary of tracking errors in percentage of the maximum translation and rotations of the manipulator are given in table I. As can be seen from the table and the figures, the new sliding mode controller, which employs a new sliding surface containing synchronization error, is performing better than simple sliding mode controller. In the simple sliding mode controller, the tracking errors in x, y and z directions increased by 10, 10 and 5.34 percent respectively when the platform is fully loaded and friction is considered but in the new sliding mode controller, the corresponding values are only 8,6 and 2.66. Generally, in both no load and full load cases, the new sliding mode controller is more than 10% better in tracking error performance. The control effort of the three controllers is given in figs. 14-19 and the figures reveal that the control signals are smooth and practically realizable. The signals for the simple sliding mode control and the new sliding mode control are almost the same and have slight oscillations at the beginning. This is because, both of them make use of the desired acceleration which is very high, 10g.

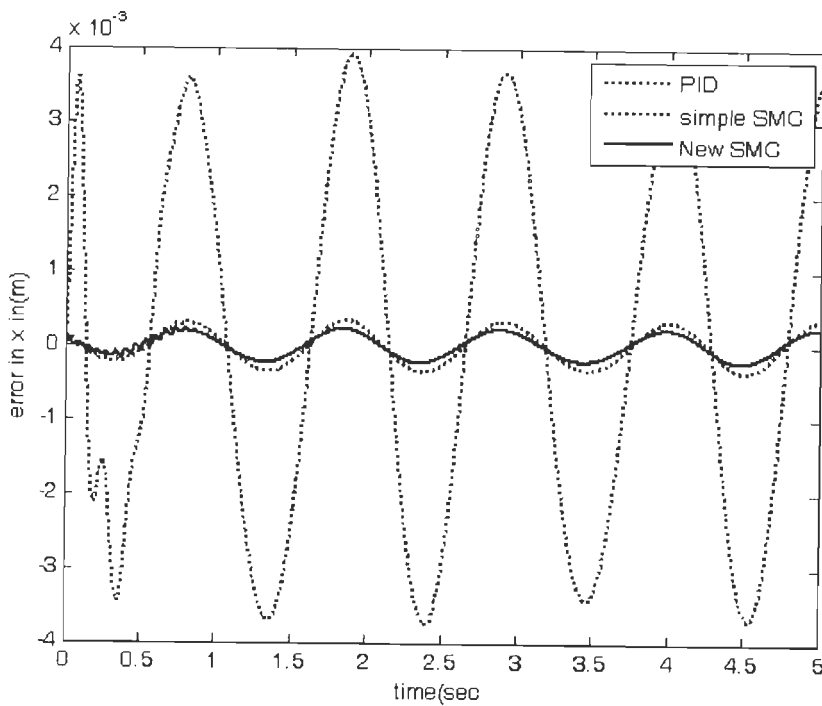


Fig.5.17 Task space tracking control in x direction

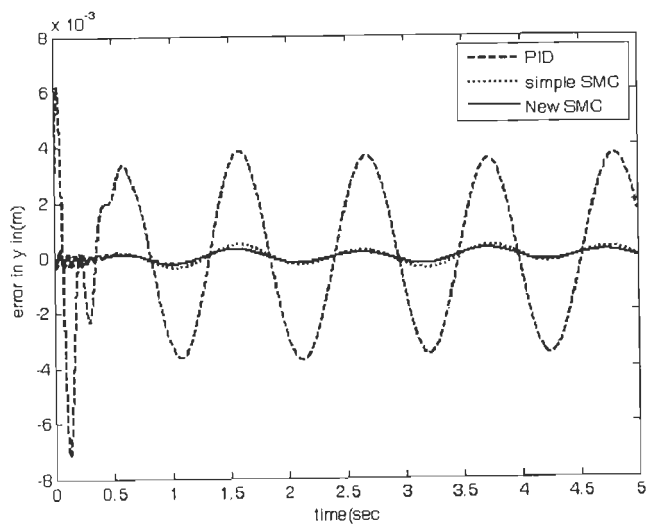


Fig. 5.18 Task space tracking control in y direction

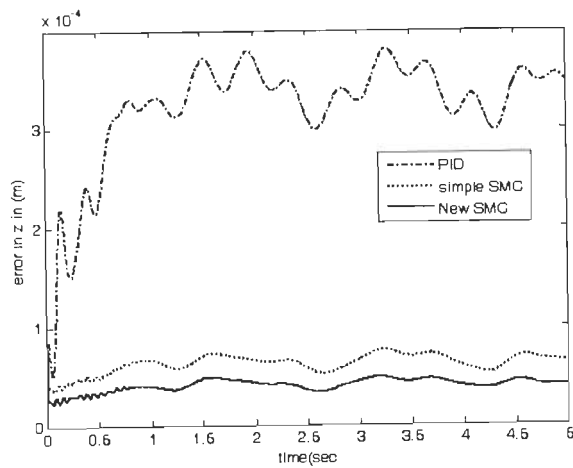


Fig. 5.19 Task space tracking control in z direction

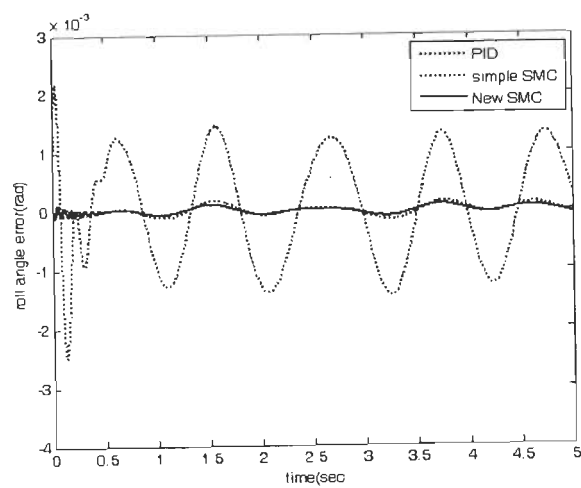


Fig. 5.20 Task space tracking control for roll angle

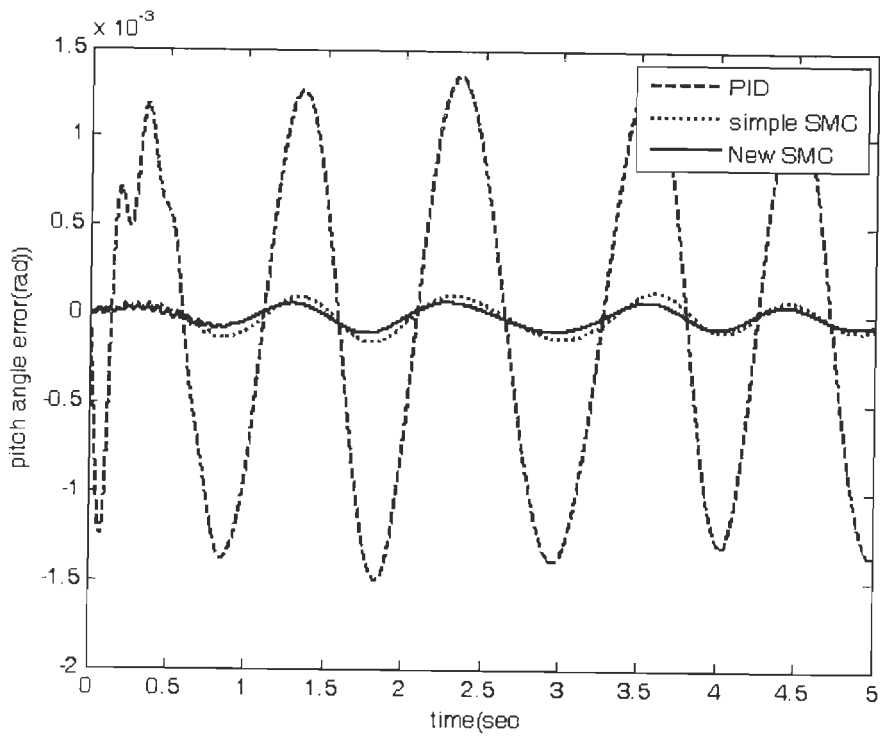


Fig.5.21 Task space tracking control for pitch angle

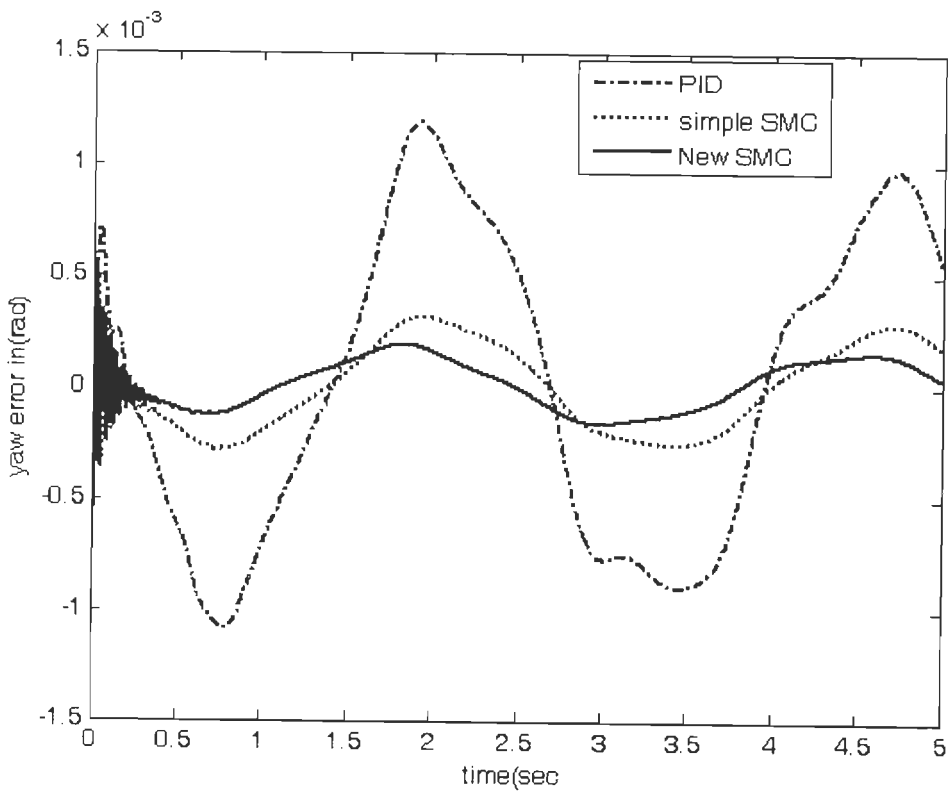


Fig.5.22 Task space tracking control for yaw angle

Table5.2 No load tracking performance of the three controllers

Controller parameters	Translational motion errors	Rotational motion errors	Remarks
PID $K_p=1 \times 10^6$ $K_d=1 \times 10^4$ $K_i=1 \times 10^3$	$e_x=e_y=4\%$ $e_z=0.4\%$	roll=1% pitch=1% yaw=1.2%	Yaw angle error is absolute error
Simple SMC $\Lambda_1=1000$ $\Lambda_2=1$ $\Lambda_3=0$ $K=1000$	$E_x=e_y=0.5\%$ $E_z=0.05\%$	Roll=0.133% Pitch=0.133% Yaw= 1.2×10^{-3}	Yaw angle error is absolute error
New SMC $\Lambda_1=1000$ $\Lambda_2=1$ $\Lambda_3=500$ $K=1000$	$E_x=e_y=0.25\%$ $E_z=0.05\%$	Roll=0.067% Pitch=0.067% Yaw= 2×10^{-4}	Yaw angle error is absolute error

Simple sliding mode control and improvements

Table5.3 Tracking error performance of the three controllers when platform is carrying payload of 200Kg and actuator friction is considered

Controller parameters	Translational motion errors	Rotational motion errors	Remarks
PID Kp=1x10 ⁶ Kd=1x10 ⁴ Ki=1x10 ³	ex=ey=25% ez=2%	roll=6% pitch=6% yaw=5x10 ⁻³	Yaw angle error is absolute error
Simple SMC Λ1=1000 Λ2=1 Λ3=0 K=1000	ex=ey=5% ez=0.267%	Roll=1% Pitch=1% Yaw=1x10 ⁻³	Yaw angle error is absolute error
New SMC Λ1=1000 Λ2=1 Λ3=500 K=1000	ex=3.5% ey=4.5% ez=0.133%	Roll=3% Pitch=3% Yaw=5x10 ⁻⁴	Yaw angle error is absolute error

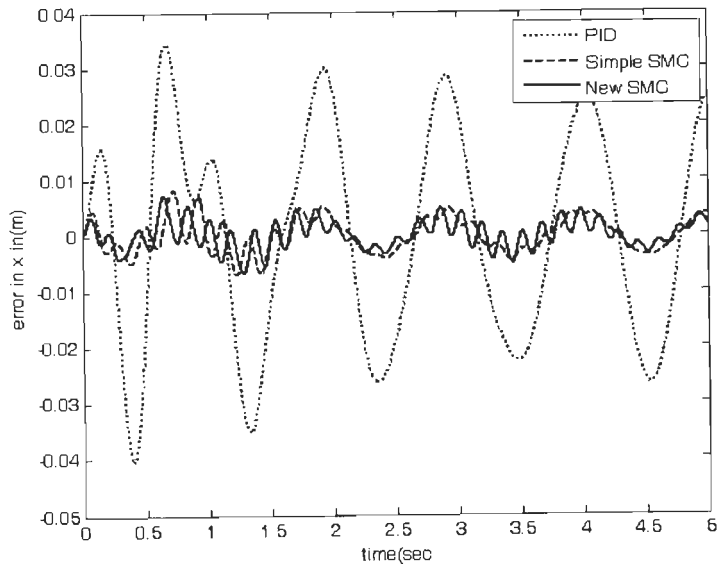


Fig.5.23 Task space tracking control in x direction with full load and friction

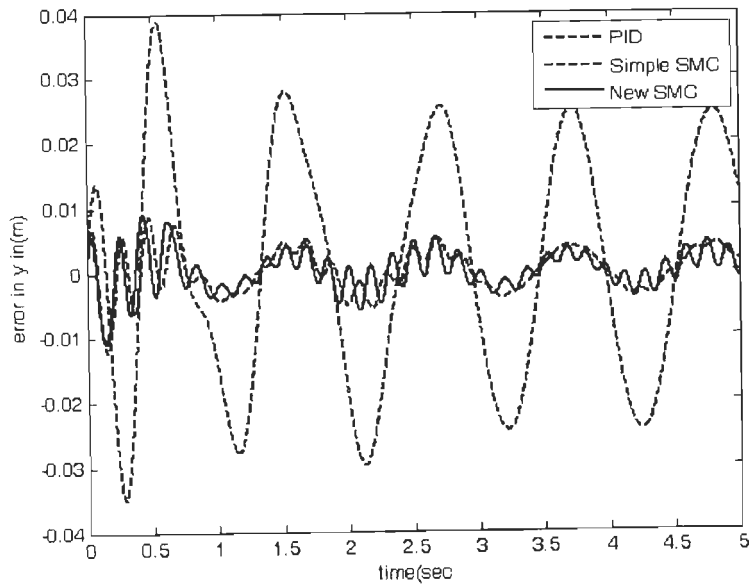


Fig.5.24 Task space tracking control in y direction with full load and friction

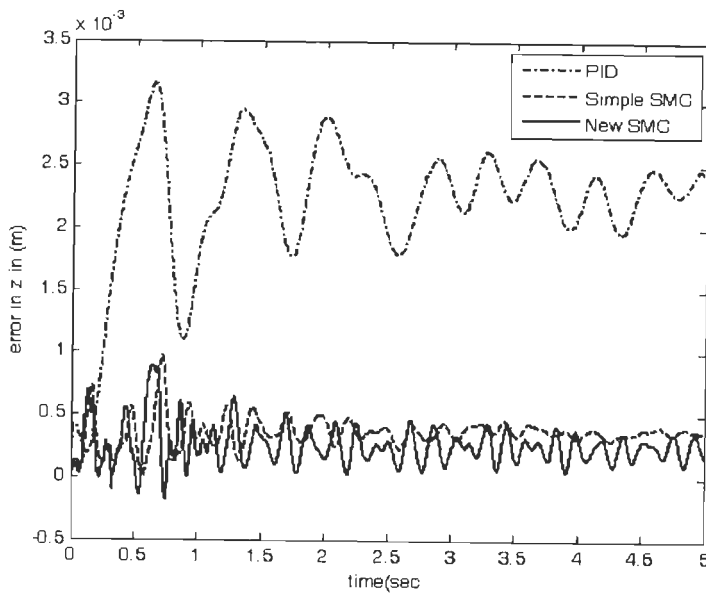


Fig.5.25 Task space tracking control in z direction with full load and friction

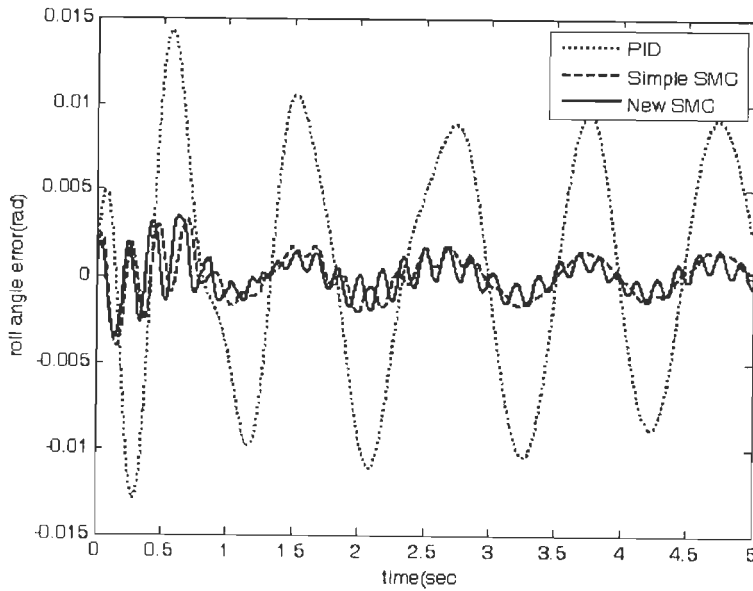


Fig.5.25 Task space tracking control in roll angle with full load and friction

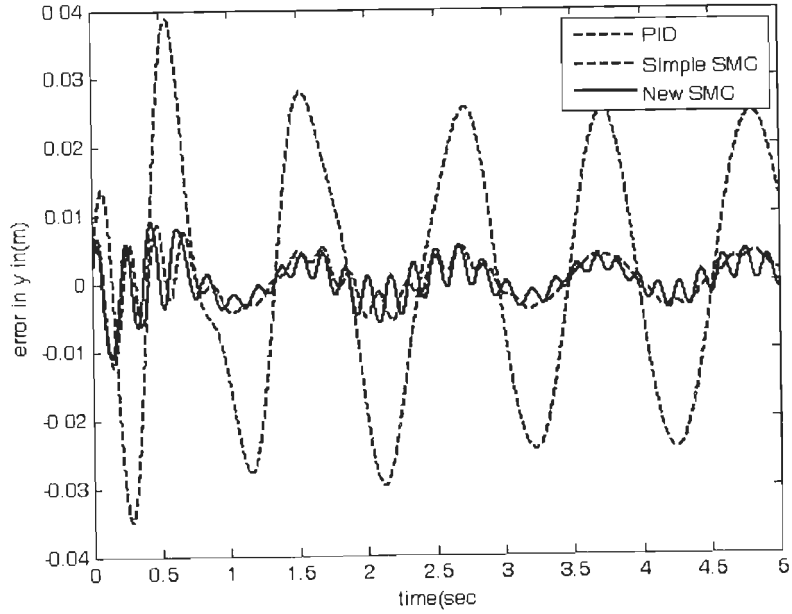


Fig.5.27 Task space tracking control in pitch angle with full load and friction

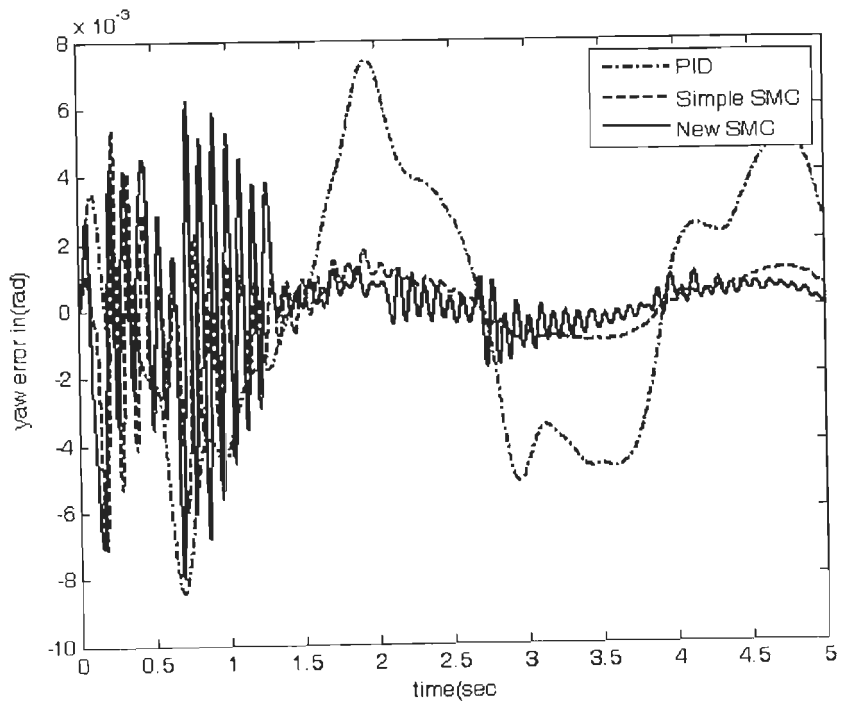


Fig.5.28 Task space tracking control in roll angle with full load and friction

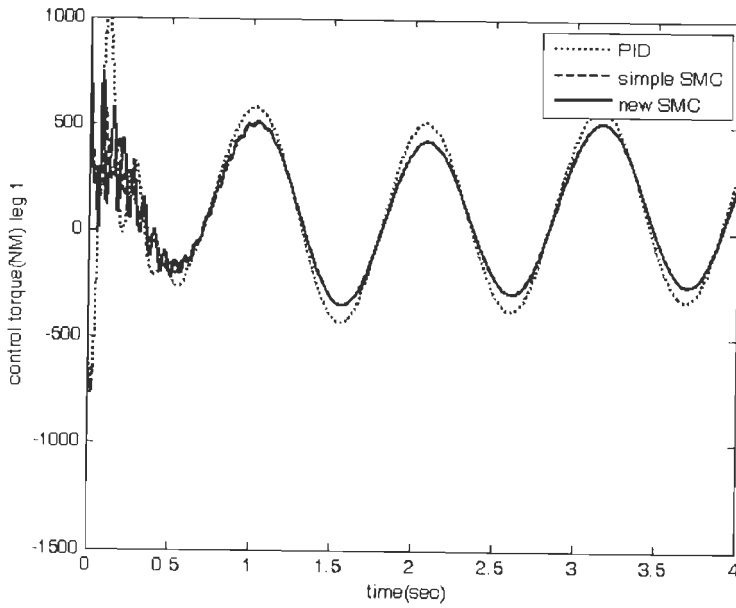


Fig. 5.29 Torque applied to leg 1 when platform is carrying a load

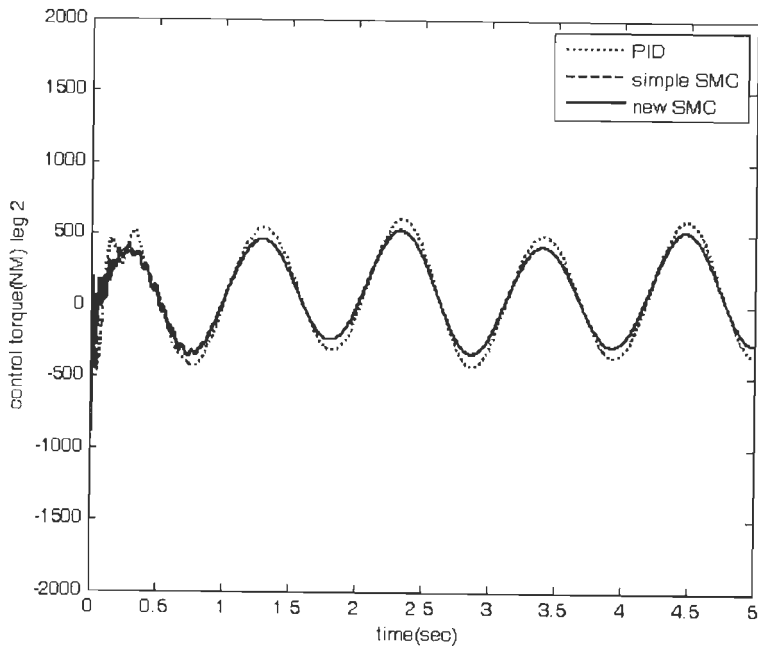


Fig. 5.30 Torque applied to leg 2 when platform is carrying a load

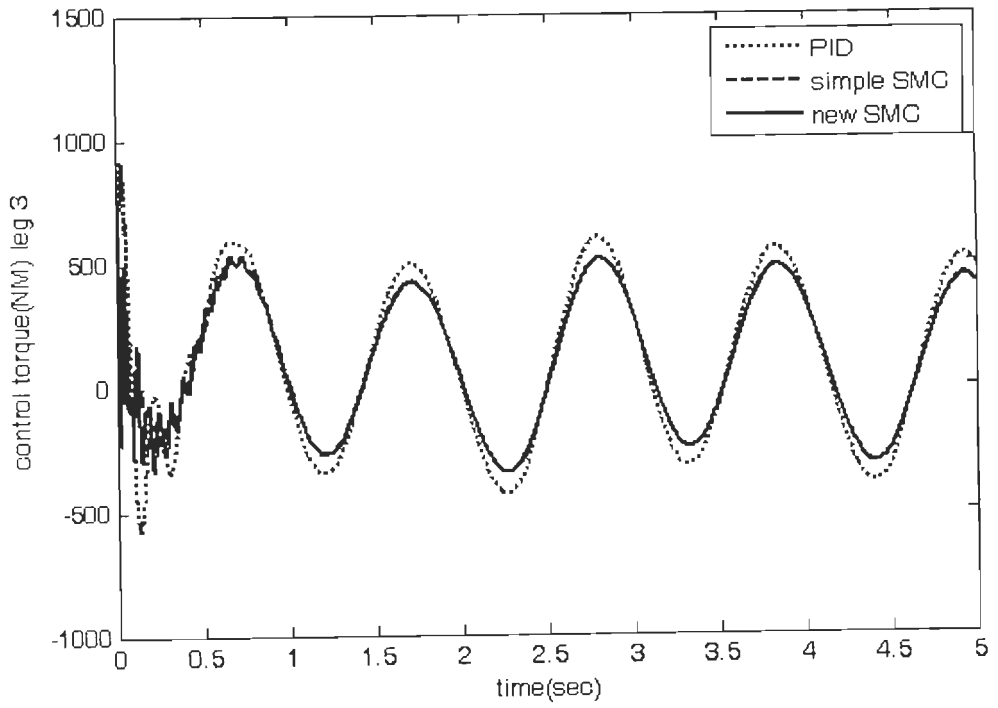


Fig.5.31 Torque applied to leg 3 when platform is carrying a load

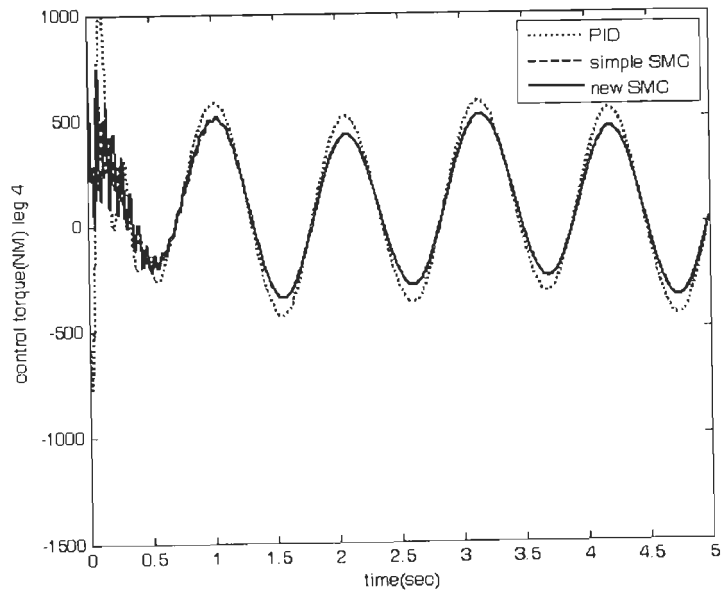


Fig.5.32 Torque applied to leg 4 when platform is carrying a load

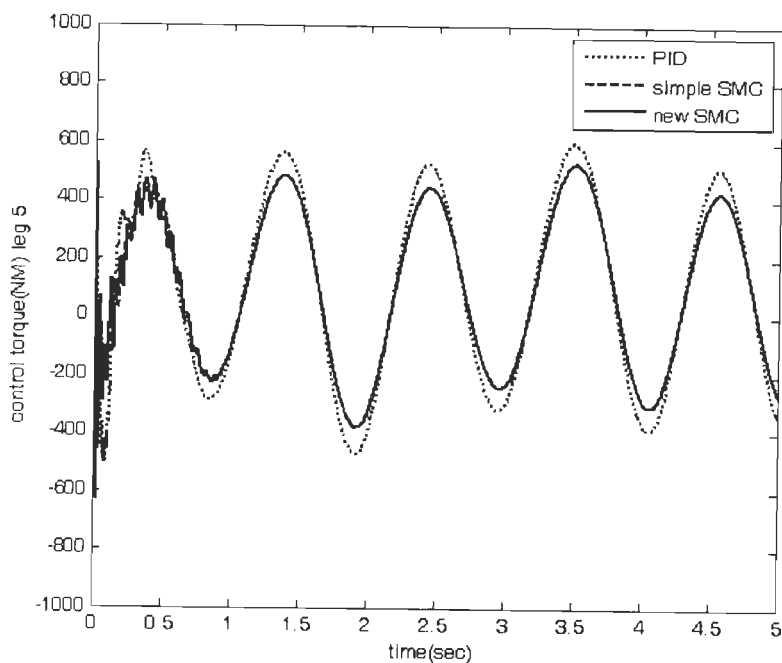


Fig. 5.33 Torque applied to leg 5 when platform is carrying a load

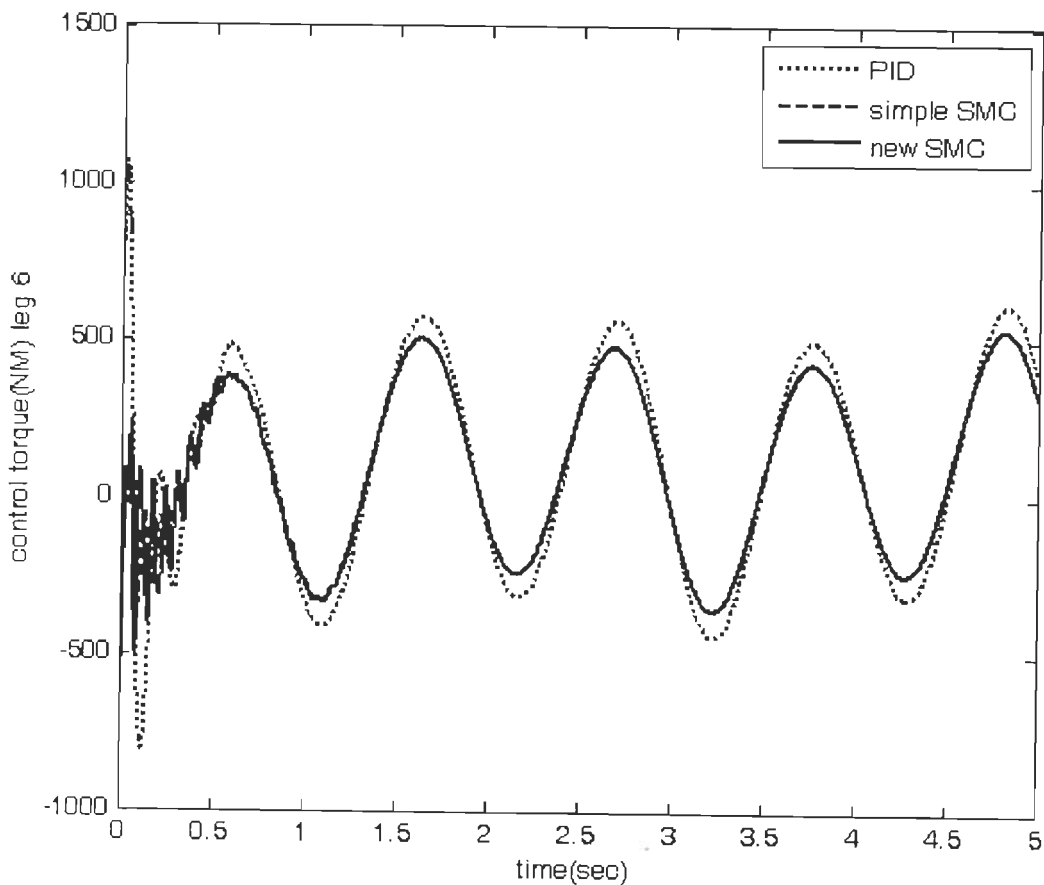


Fig. 5.34 Torque applied to leg 6 when platform is carrying a load

5.4 Discussion and conclusion

In this chapter we have seen three different controllers designed for high performance control of Stewart platform manipulator. The first one is task space fuzzy sliding mode controller (TFSMC); the second one is task space fuzzy sliding mode controller with integral loop (TSFSMCIP) and the last one is a sliding mode controller implemented as hybrid of task space and joint space with a newly proposed sliding surface. The performance of all three controllers is better than joint space PID controller with the performance of the task space fuzzy sliding mode controller with integral loop having slightly better performance than the other two.

The hybrid implementation has advantages of avoiding forward kinematics estimation and is very much cost effective. Its performance is also comparable to the task space controllers. The newly proposed sliding surface is able to drive the synchronization error to zero and hence solves the problem of joint space control. One important point to be noted here is, in all of the three controllers, the sliding mode controller is a classical one and hence has the drawbacks mentioned in chapter 2. To enhance the performance of task space controller's integral sliding controller has to be used. In the next chapter we will discuss the design and implementation of integral sliding mode controllers in detail.

CHAPTER 6

GENETIC ALGORITHM BASED INTEGRAL SLIDING MODE CONTROL

6.1 Introduction

In chapter 5, we discussed various implementations of basic sliding mode controller used to improve the performance of Gough-Stewart platform manipulator. Though it shows better performance compared to joint space independent leg PID controllers, classical sliding mode control has certain drawbacks. These are: (i) chattering or high frequency oscillation of control signal, (ii) lack of robustness during the sliding phase, (iii) reduced life time of actuators and (iv) is unable to compensate unmatched uncertainties [144].

Integral sliding mode controller is an improvement of the conventional SMC that can solve the reaching phase problem and improve the robustness against unmatched uncertainties. Its basic structure and advantages has been discussed in [76] [150] [151] [156]. The reaching phase problem of conventional SMC is solved by integral SMC because, integral SMC has a nonlinear sliding surface with an integral term that constrain the system states to be on sliding mode from the initial time[55]. The other important advantage of the integral sliding surface is it helps to achieve a stable sliding dynamics and gives enables to design a control signal that completely compensates matched uncertainties and minimize unmatched uncertainty to any desired level [42]. This improves robustness to unmatched uncertainty. Therefore, the central point in the design of integral SMC is the design of the integral sliding surface.

In the case of classical SMC, there are various established methods used for the design of sliding surface such as: eigenstructure assignment, Lyapunov based method and pole

assignment method [153]. But, there is no any established method for the design of integral sliding surface. There are few methods suggested in the literature. In [81], Lyapunov's direct method was used, while the authors of [153] suggested matrix fraction description method for MIMO systems described using polynomial matrix. In [78], a design method based on equivalent transfer function was proposed and in [141], fuzzy logic based integral sliding mode controller was implemented but no formal design method is given. In all of the above methods, the systems were assumed to have matched uncertainty only and hence the methods cannot be used for systems with unmatched uncertainty. The only methods proposed for systems having unmatched uncertainty are that of [42] and [55]. The authors of [42] proposed to use the transpose of the output matrix as a gain of integral sliding surface. But, there method needs a system with constant input matrix. In [55], linear matrix inequality (LMI) technique was employed to design an integral sliding mode controller for a system having matched and unmatched uncertainties in both input and state matrix. The method is complex due to the mathematical complexity of LMI. Therefore in this chapter, an important design method used to determine parameters of the integral sliding surface is presented.

In section one, basics of integral sliding mode control is discussed. Then in section two, the application of genetic algorithm for the design of integral sliding mode controllers is presented. In the third section, genetic algorithm based integral sliding mode controller application to the Gough-Stewart platform is given. In the last section, the extension of the method to multi-objective optimization using genetic algorithm and its application to Stewart platform manipulator is discussed.

6.2. Integral sliding mode controller

Consider an uncertain nonlinear system given as

$$\dot{x} = f(x) + g(x)u + d(x, t) \tag{6.1}$$

Where $x \in \mathcal{R}^n$ is state vector, $f(x)$ and $g(x)$ are $nx1$ and nxm dimensional vector valued and matrix valued smooth nonlinear functions, u is $mx1$ dimensional vector of control signal, d is lumped uncertainty term which includes matched and unmatched uncertainties due to parameter variations and unmodeled dynamics. The following assumptions are taken:

Assumption 1: The nominal system $\dot{x} = f(x) + g(x)u_0$ is stabilizable through a nominal controller u_0 .

Assumption 2: The matched and unmatched uncertainties are not known but bounded.

The control signal in integral sliding mode control is given by

$$u = u_0 + u_1 \quad (6.2)$$

where u_0 is the nominal control input designed to stabilize the nominal system and u_1 is a discontinuous control input given by

$$u_1 = K_1 f_s(s) \quad (6.3)$$

$f_s(s)$ being a switching function and K_1 is constant gain value. The integral switching surface is given as follows [119].

$$s(x, t) = G_s \left[x - x(0) - \int_{t=0}^t (f(x) + g(x)u_0) d\tau \right] \quad (6.4)$$

Where s is the sliding parameter and G_s is $m \times n$ gain vector (when system is SISO and $m=1$) or matrix (when the system is MIMO).

In (6.4), the term in the bracket can be seen as the miss match between actual plant output x and desired or nominal response. This can be shown in block diagram as displayed in Fig.6.1.

At $t=0$, $s(x,0)=0$ and hence the system starts on the sliding surface avoiding the non robust reaching phase and sliding mode exists for all time. If the controller (6.2) is able to drive the actual system as desired, the deviation becomes zero and $s=0$ for all time. But due to the presence of disturbances, this may not occur and there will be some dynamics of sliding. The effect of the integral sliding surface on the sliding dynamics can be analyzed as follows.

Taking the derivative of s ,

$$\dot{s} = G_s \dot{x} - G_s (f(x) + g(x)u_0) \quad (6.5)$$

Substituting the system dynamics from (6.1) into (6.5)

$$\dot{s} = G_s f(x) + G_s g(x)u + G_s d(x, t) - G_s f(x) - G_s g(x)u_0 \quad (6.6)$$

And then the equivalent control signal becomes,

$$u_{eq} = -(G_s g)^{-1} G_s d(x, t) + u_0 \quad (6.7)$$

Substituting the equivalent controller into the system dynamics (6.1), the sliding dynamics or sliding manifold becomes

$$\dot{x} = f(x) + g(x) \left\{ -(G_s g(x))^{-1} d + u_0 \right\} + d \quad (6.8)$$

This can be rewritten as,

$$\dot{x} = f(x) + g(x)u_0 - \left[g(x)(G_s g(x))^{-1} G_s \right] d + d \quad (6.9)$$

From (6.9) one can see that, if the gain G_s of the integral sliding surface is optimally selected so that the term in the square bracket is 1, then the uncertainty term will be totally rejected and a stable sliding surface is obtained.

Hence the objective of this chapter is to show how genetic algorithm and multi-objective optimization can be used to optimally select the gains G_s and K_1 of integral sliding mode controller to stabilize and control an MIMO uncertain nonlinear system having unmatched uncertainty. Note that (6.7) and (6.9) need the condition that $G_s g$ is nonsingular and this is always true for mechanical systems.

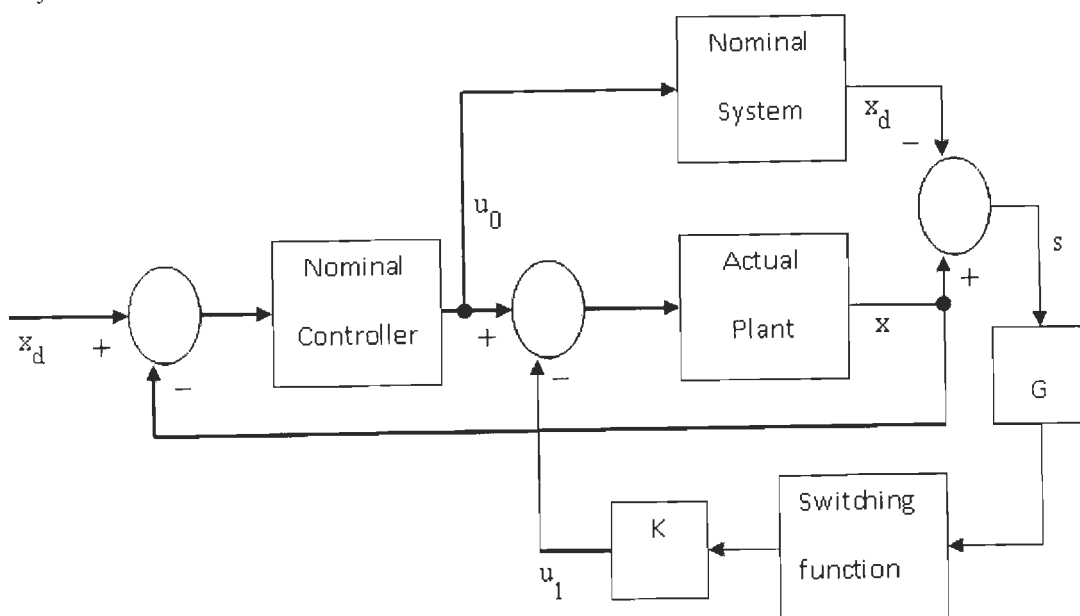


Fig.6.1 Block diagram representation of integral sliding mode control

6.3 Integral sliding mode controller design using genetic algorithm

6.3.1 Genetic Algorithm

Genetic algorithm (GA) is an evolutionary algorithm based on Darwin's theory of selection of the fittest. It is a multidimensional search algorithm which solves the local minima problem of classical algorithms. In the literature, the basic element of a genetic algorithm is known the chromosome as it is based on the evolutionary theory of Darwin [9]. The chromosome contains the genetic information for a given solution and can be coded by using either binary or real string. The algorithm starts by generating randomly some number of

chromosomes as candidate solutions to a given problem. A fitness function which in effect is a performance index is used to select the best solution in the population to be parents to the offspring that comprises the next generation. The more fit the parent is, it will have greater probability of selection.

The selection of parent chromosomes is done using various methods including the roulette wheel method. Then offsprings are produced by selecting parent chromosomes for breeding and crossing over some of the genetic material. This process is known as crossover. Another operator which is used to introduce some element of randomness into the solution is mutation. In the process of mutation a randomly selected gene of an offspring is changed. Mutation occurs in not all offsprings but very few ones and it is used to introduce some randomness. This process continuous until a global solution is obtained. Figure 2 shows the algorithm described above. Therefore, in a GA optimization, parameters such as, the initial population size, crossover rate and mutation rate, coding size of chromosomes and fitness function, have to be selected. The most important one which determines the problem at hand is the fitness function and below we will study how to select the fitness function.

6.3.2 Fitness function

The most important parameter in a GA optimization is the fitness function since it determines the objective of the optimization itself. A poorly selected objective function may give a completely wrong result. In [81], the main objective was to decrease chattering and obtain fast response and the authors used fitness function given by

$$f_{obj} = e^{-(t_h/w_1)^2} \times e^{-(c_m/w_2)^2} \quad (6.10)$$

where t_h is the time to hit the sliding surface, c_m is the amount of chattering and w_1 and w_2 are weight factors. In the current discussion, the first term is not needed because the system is on the sliding surface from the initial time. The main objective here is to keep the system on the sliding surface from the initial time. Another objective is to minimize chattering and the effect of the unmatched uncertainty as shown in (6.9). Hence for the fitness function, the product of two terms is taken: the first one used to constrain the states on the sliding surface and the second one is to minimize the effect of unmatched uncertainty. The fitness function is formulated as in (6.11) below.

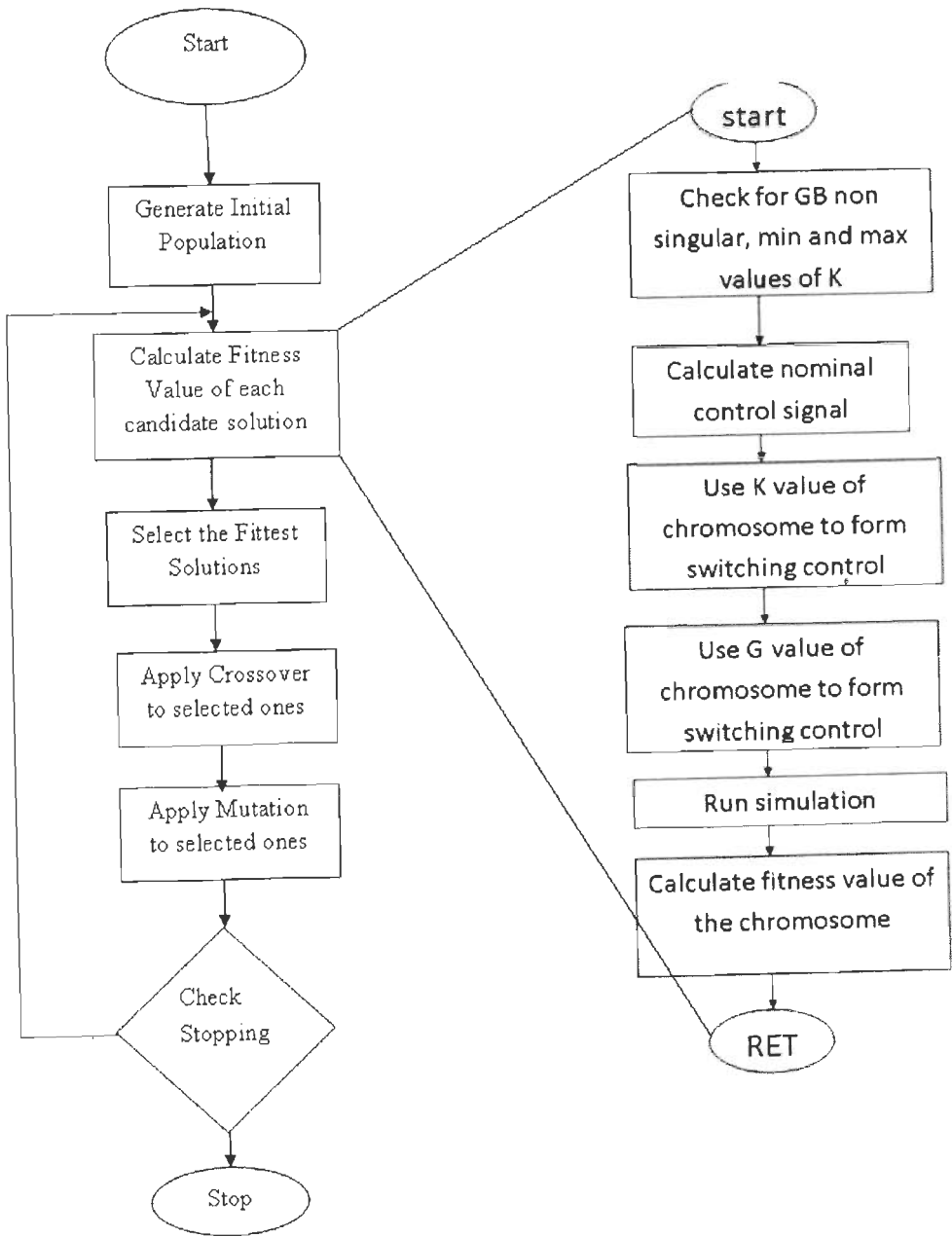


Fig.6.2 Steps of Genetic Algorithm tuning for integral sliding mode controller design

$$f_{obj} = e^{-(c_s/w_1)^2} \times e^{-(u_m/w_2)^2} \quad (6.11)$$

Where c_s is the term used to constrain the states to the sliding surface and is taken as

$$c_s = \sum_{i=1}^N \|s_i\| \quad (6.12)$$

s_i is the value of the sliding variable in iteration i . In the implementation, a chromosome which has an initial value of s greater than zero by some upper limit is given a big penalty. Similarly the u_m is the term which is used to minimize the effect of unmatched uncertainty and is taken as the norm of

$$u = \left\| I + g(x) \left(G_s g(x) \right)^{-1} G_s \right\| \quad (6.13)$$

6.3.3 Coding of Genes

In searching for the optimal value of gain, the genetic algorithm will be initialized by candidate values of the gain matrix G . For a general nonlinear system given in (6.1), the gain matrix G is n by m . Each chromosome in the GA optimization will have a certain value for each of the nm parameters. The gain parameters or values have to be coded by some number of genes, which can be either binary or real. While binary coding results in a long chromosome, real coding results in a shorter length. Hence each chromosome will have nmp genes, where p is the number of digits used to represent each gain parameter. For the current problem, we consider coding the genes using real numbers.

Hence the i^{th} chromosome containing nm gains each coded by p digits can be represented as

$$G_{si} = \left[g_{11} g_{12} g_{13} \cdots g_{(n-1)m} g_{n1} g_{n2} \cdots g_{nm} \right] \quad (6.14)$$

Where each g is coded using p decimal digits as

$$g_{jk} = d_{jk1} d_{jk2} d_{jk3} \cdots d_{jkp} \quad (6.15)$$

Remark: The gain K_1 of the discontinuous control signal (6.3) can also be tuned together with the parameters of the sliding surface. This is another advantage of the GA tuning method.

6.3.4 Determining range of values for the parameters

The range of values for the parameters to be tuned is an important factor in the application of GA and hence it has to be analyzed carefully. Some combinations may result in positive feedback and result in unstable system. The range of values to be taken depends on the particular application for which the GA is used. Generally the range of values can be decided from prior information on system performance or from some preliminary tests. For example for robotic control, the range of values can be decided from step response test of individual joints.

In particular, the minimum value of the gain K_1 of the discontinuous control (3) can be determined from stability analysis of the closed loop system. Lyapunov's direct method and the uncertainty bounds of the uncertain terms can be used to determine the lower bound while the upper bound is determined by the maximum value of the control signal or by the saturation limit of actuators. Hence, taking a Lyapunov function

$$V = \frac{1}{2} s^T s \quad (6.16)$$

Taking the derivative along the trajectory where s is zero,

$$\dot{V} = s \dot{s} \quad (6.17)$$

$$= s \left\{ G \left(\dot{x} - f(x) - g(x)u_0 \right) \right\} \quad (6.18)$$

This can be simplified by substituting the expression (6.1). For asymptotic stability of the system,

$$\dot{V} \leq -\gamma \quad (6.19)$$

where γ is a positive constant. Therefore, to achieve the above stability condition, the gain of the discontinuous control u_1 has to be selected so that (6.21) is fulfilled. If the discontinuous control signal is taken as

$$u_1 = \begin{cases} -K \frac{(D_s g)^T s}{\|(D_s g)\| \|s\|} & \text{if } s \neq 0 \\ 0 & \text{if } s=0 \end{cases} \quad (6.20)$$

Then the gain K has to be

$$K > G_s \|d_m\| \quad (6.21)$$

where d_m is the maximum value of the lumped uncertainty containing both matched and unmatched uncertainties.

6.3.5 SISO system example with constant input and state matrix

Consider an uncertain system with the following dynamic equation as given in [55].

$$\dot{x} = (f(x) + \Delta f(x))x + (g(x) + \Delta g(x))(u + h(t))$$

The nominal system parameters are

$$f(x) = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } g(x) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The uncertainties in $f(x)$ and input matrix $B(x)$ are given as

$$\Delta f = \begin{bmatrix} 0 & p_1 \cos(u) & 0 \\ p_2 \cos(x_1) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and}$$

$$\Delta g(x) = \begin{bmatrix} 0 \\ p_3 \cos(x_1) \\ 0 \end{bmatrix}$$

Where p_1 , p_2 and p_3 are given by

$$p_1 = 0.2 \sin(10\pi t), p_2 = 0.2 \sin(20\pi t) \text{ and } p_3 = 0.1 \sin(30\pi t)$$

The disturbance d is also given as

$$h(t) = 0.6 \sin(60\pi t)$$

and the initial value of the states is $x_1(0) = 0$, $x_2(0) = 0$ and $x_3(0) = 1$. The above dynamic system has both matched and unmatched uncertainties and cannot be controlled by classical sliding mode controller. In [55] an integral sliding mode controller was designed for this system using LMI method. Here we use genetic algorithm to design an efficient integral sliding mode controller. The dynamic system can be rewritten in the form of (6.1) by taking

$$d = \Delta f(x)x + \Delta B(x)h + h$$

Note: In the expression above, the first two terms are the unmatched uncertainty parts and the last term h is the matched uncertainty.

The nominal controller u_0 is designed using LQR method by taking the linear nominal system. In the LQR design Q and R matrixes are taken as

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and $R=1$. Then the gain for the nominal control $u_0 = -Kx$ is obtained as $K=[0.1176 \ 1.2966 \ 1.8956]$. Then based on the response of the nominal control and disturbance bounds, the range of values for the gain of the sliding surface G and the gain of the discontinuous control are set in the genetic algorithm as

$$\text{Range of values for } G = \begin{bmatrix} -2 & 2 \\ -2 & 2 \\ -2 & 2 \end{bmatrix} \text{ and for the gain of the discontinuous control } K_1 = [1 \ 20].$$

The parameters of the genetic algorithm were set as crossover rate 0.8, mutation rate 0.2, initial population 20 and generation 100. The optimal values obtained are

$$G_s = [1.12 \ 0.188 \ 0.577]$$

$K_1 = 6.9109$. The simulation result using the above values is given in Fig. 6.3-Fig.6.5. The decay rate for the states is very much close to the desired compared with the LMI design of [55]. However, the sliding parameter is not zero initially but it comes to zero with in a negligible time and remains constantly zero as shown in Fig.6.6. The control signal shown in Fig.6.7 is smooth and chatter free.

6.4 Multi-objective genetic algorithm optimization and integral sliding mode controller

6.4.1 Multi-objective optimization

Multi-objective optimization is an optimization problem where there are multiple conflicting objectives to be fulfilled and usually a single solution that satisfies all objectives may not be obtained [9]. But a set of optimal solutions, which are optimal in the sense that no improvement can be made in any objective without sacrificing the others, can be found. A practical example of multi objective optimization is the robust control of Stewart platform manipulator. Specifically in integral sliding mode control, robustness against unmatched

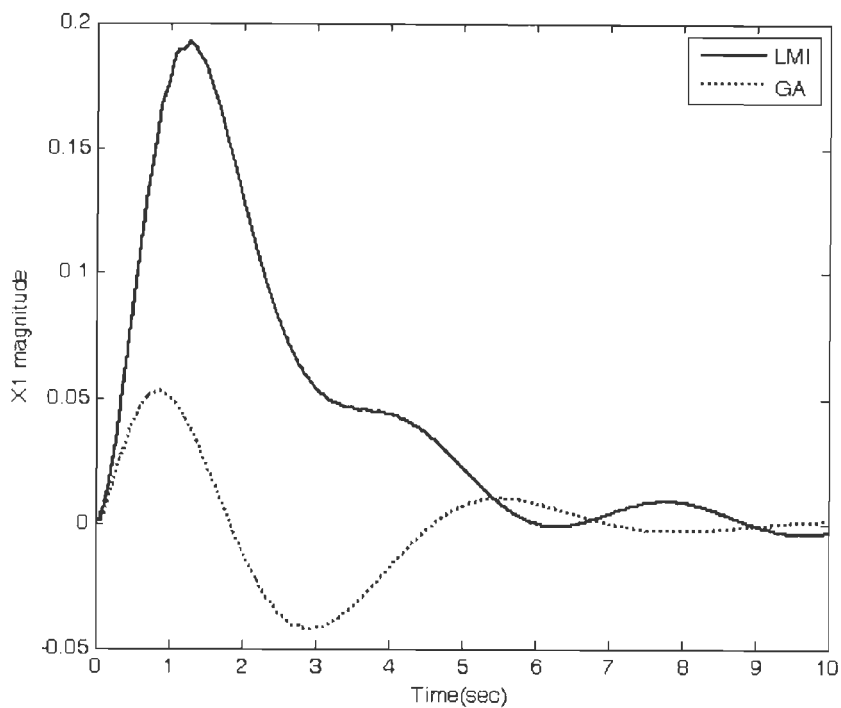


Fig.6.3 Closed loop response, X_1

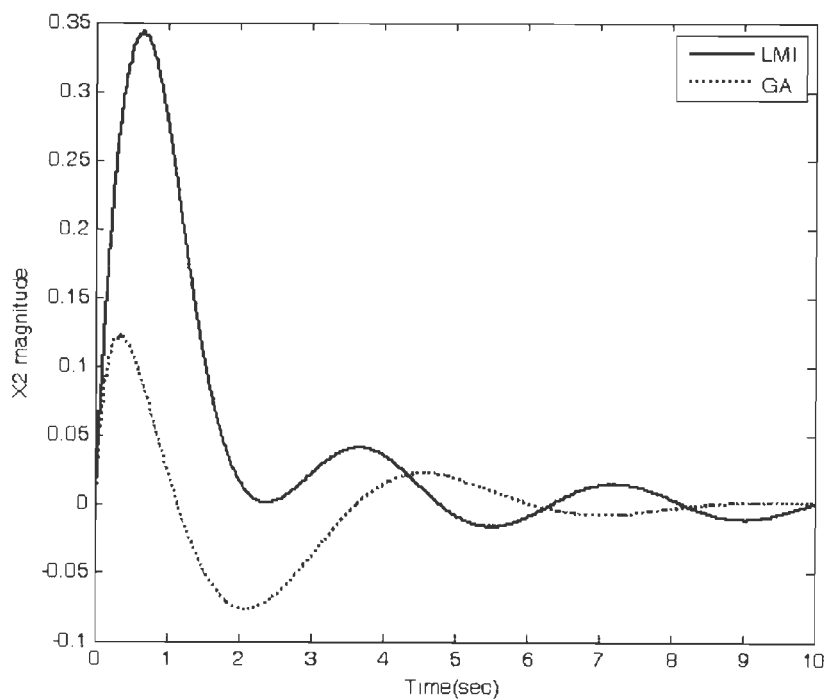


Fig.6.4 Closed response for X_2

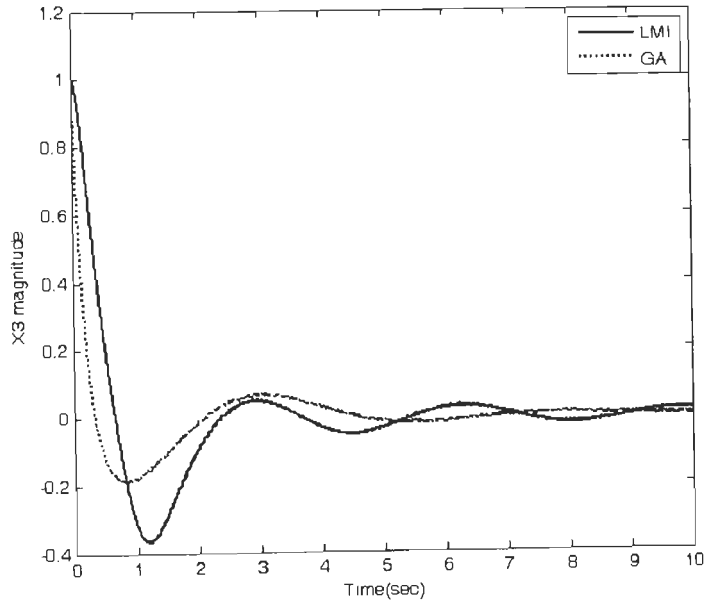


Fig.6.5 Closed loop response X3

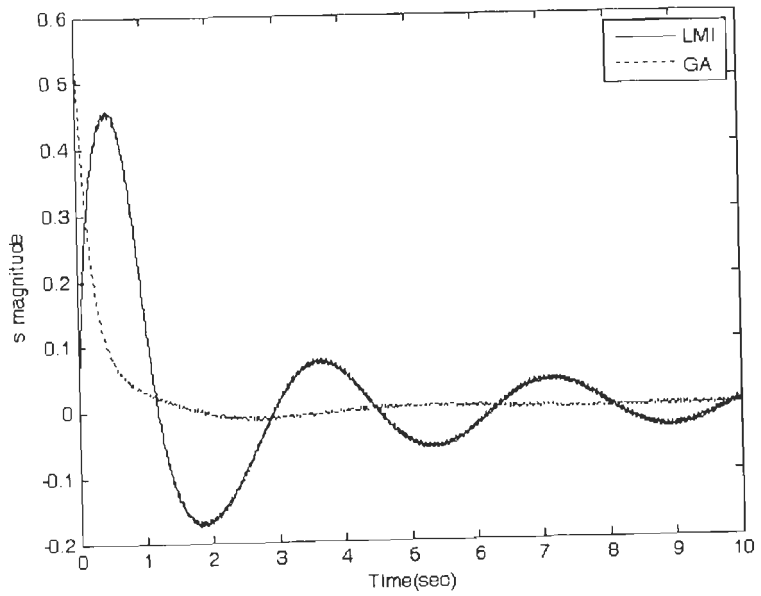


Fig.6.6 The sliding parameter s is constantly maintained at zero

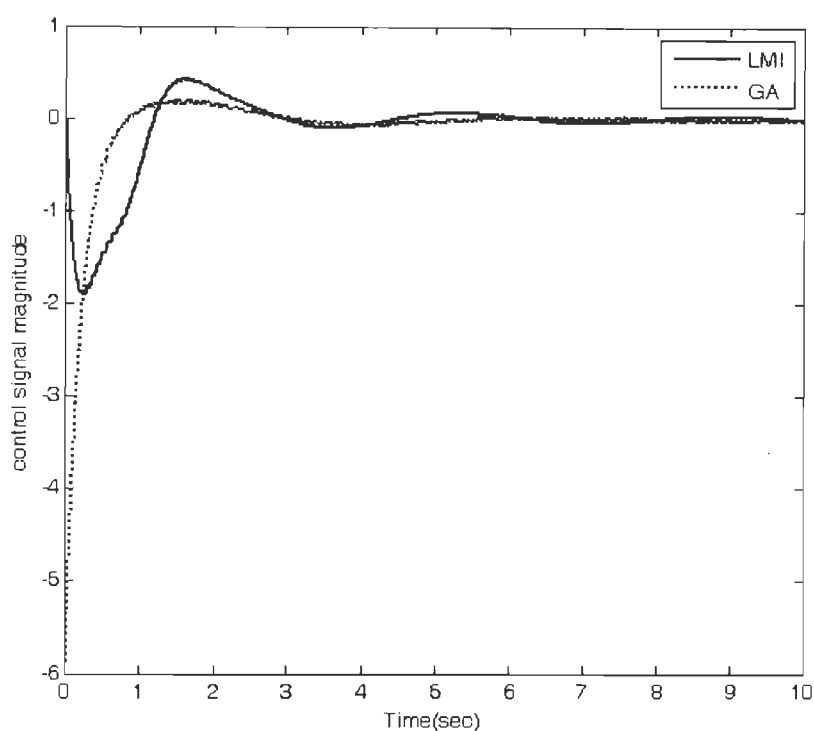


Fig.6.7 Control signal is smooth and chattering is almost zero

uncertainty has to be improved and non robust reaching phase of classical SMC should also be avoided. At the same time good performance has to be achieved which is a conflicting requirement with the previous one. There are various methods used to solve multi-objective optimization problems and in [9], a tutorial has been given about the use of genetic algorithm for multi-objective optimization. In this section, we will use genetic algorithm and multi-objective optimization to design integral sliding mode controller for Stewart platform manipulator.

As discussed in section 6.2, in genetic algorithm fitness function is used to select the best solution out of the candidates. Out of the candidate solutions, some of them may be unfit. For example, for the integral sliding mode controller case, the values to be used for K_1 have to fulfill the condition for stability. Hence a chromosome containing a value which makes the system unstable has to be removed.

The most important step in using genetic algorithm for multi-objective optimization is how to formulate the single fitness function required in genetic algorithm out of the multiple objectives. There are various methods used including weighted sum of the individual objective functions [9]. In this section we present the design of the projection matrix G_s and the control law to be used to drive a Stewart platform manipulator.

6.4.2. Selection of objective functions

As explained in section two, in the design of ISMC, the main objective is to avoid the non-robust reaching phase and to minimize the effect of unmatched uncertainty. Moreover, as in any control system, the closed loop stability of the system has to be insured. This is achieved through proper selection of the objective functions and this is done as follows. To remove the non-robust reaching phase, we have to minimize the deviation of the actual system from nominal value, i.e. we should minimize the value of s , which is the sliding parameter. To decrease the effect of unmatched uncertainty and chattering, we should minimize the rate of change of s . Hence the following objective functions are taken. To minimize the deviation of the actual system from nominal value, objective function is taken as:

$$f_1 = e^{-(c_{s1}/w)^2} \quad (6.22)$$

Where w is weight value for the given objective and c_{s1} is the term used to constrain the states to be on the sliding surface and is taken as

$$c_{s1} = \sum_{i=1}^N s_i \quad (6.23)$$

s_i is the value of the sliding variable in iteration i , N is the number of samples in the trajectory used. Note that, evaluating the objective functions needs running the closed loop control system for the given trajectory using the candidate solution values for G_s and K_1 . In the implementation, a chromosome which has an initial value of s greater than zero by some upper limit is given a big penalty. To minimize the rate of change of s , a similar objective function to that of (10) is formulated using the rate of change of the sliding parameter as

$$f_2 = e^{-(c_{s2}/w)^2} \quad (6.24)$$

Where w again is the weight vector and c_{s2} is given by

$$c_{s2} = \sum_{i=1}^N \dot{s}_i \quad (6.25)$$

Therefore, for MIMO system having m control outputs, there will be $2m$ objective functions and a weighted sum of these $2m$ objective functions should be taken as the final objective function for the optimization.

To insure stability, the value of K_1 has to be checked for the condition given in (6.21), i.e. all candidate chromosomes will be checked for their value of K_1 before evaluating the objective function, which needs running the closed-loop system. The non singularity of Gg will also be tested mathematically and all chromosomes which fail to fulfill the stability condition and other conditions will be given high objective function value so that they are not selected for next steps.

6.5 Application to Stewart platform manipulator

In this section we will make use of the above steps of genetic algorithm based integral sliding mode controller design to design an integral sliding mode controller for Stewart platform manipulator. We design the controller both in joint space and task space and compare the performances.

6.5.1 Integral sliding mode controller design in joint space

6.5.1.1 Design and analysis of controller

Consider the dynamic model of a Stewart manipulator given in (2.10) with actuator friction and external disturbance included.

$$A(q)\ddot{q} + B(q, \dot{q})\dot{q} + Q(q) = \tau - f_f - \tau_d \quad (6.26)$$

Where q is vector of joint parameter, i.e. vector of elongations of six legs, A is 6×6 manipulator inertia matrix, B is also 6×6 coriolis and centrifugal torque/force, Q is 6×1 gravitational torque/force, τ is the actuator torque, f_f is torque due to friction and τ_d is the disturbance. The same assumptions on the mass matrix and manipulator Jacobian as given in chapter 2 and other chapters are considered. The assumption on the uncertainty of the dynamic parameters is also same.

The joint space tracking error can be given as

$$e = q_d - q \quad (6.27)$$

where q_d is the desired joint elongation. Then, the joint space error dynamics is given as:

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= \ddot{q}_d - A_N^{-1}(\tau - B_N(q, \dot{q})\dot{q} - Q_N(q)) + d_{21} \end{aligned} \quad (6.28)$$

Where the nominal joint space parameters are obtained from task space values by using

$$A_N = J^T M_N J^{-T} \quad (6.29)$$

$$B_N = J^T (M_N \dot{J}^{-T} + V_N J^{-T}) \quad (6.30)$$

$$Q_N = J^T G \quad (6.31)$$

$$d_{21} = -A_N^{-1} J^T (\Delta M J^{-T} \ddot{q} + (\Delta M \dot{J}^{-T} + \Delta V J^{-T}) \dot{q} + \Delta G) + A_N^{-T} f_f \quad (6.32)$$

Comparing (6.28) and (6.1),

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (6.33)$$

$$f = \begin{bmatrix} e_2 \\ \ddot{q}_d + A_N^{-1} (B_N(q, \dot{q})\dot{q} + Q_N(q)) \end{bmatrix} \quad (6.34)$$

and

$$g = \begin{bmatrix} 0 \\ -A_N^{-1} \end{bmatrix} \quad (6.35)$$

and d is

$$d = \begin{bmatrix} 0_{6 \times 1} \\ d_{21} \end{bmatrix} \quad (6.36)$$

Following the same procedure as in section 6.2, the integral sliding surface for the Stewart platform manipulator is given as

$$S = G_s \left(x - x(0) - \int_0^t (f(x, \omega) + g(x, \omega)\tau_0) d\omega \right), \quad (6.37)$$

where τ_0 is the nominal control torque, $x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$ and $x(0)$ is the initial condition of the error dynamics and f and g are as given in (6.34) and (6.35) above. If the nominal controller to be used is chosen as

$$\tau_0 = D_N(\ddot{q}_d + K_p e + K_d \dot{e}) + B_N(q, \dot{q})\dot{q} + Q_N \quad (6.38)$$

for some positive diagonal matrices K_p and K_d , then the sliding dynamics of the system becomes,

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= -K_p e - K_d \dot{e} \end{aligned} \quad (6.39)$$

which shows a stable sliding dynamics. The complete control signal is given as

$$\tau = \tau_0 + Kf_s(S) \quad (6.40)$$

where τ_0 is the nominal control signal given by (6.38), K is gain of switching function, $f_s(S)$ is switching function.

The magnitude of K required to achieve stability is

$$K \geq \left\| -A_N^{-1}J^T \left(\Delta M J^{-T} \ddot{q} + (\Delta M J^{-T} + \Delta V J^{-T}) \dot{q} + \Delta G \right) \right\| + \left\| A_N^{-1}f_r \right\| \quad (6.41)$$

6.5.1.2 Simulation results and discussion

The integral sliding surface given in (6.37) is implemented after the expressions (6.33)-(6.36) are substituted and after simplification it becomes

$$S = G_{s1}(e - e(0) - \int \dot{e}_N) + G_{s2}(\dot{e} - \dot{e}(0) + \int K_p e + K_d \dot{e})$$

where G_{s1} and G_{s2} are 6x6 diagonal matrices forming the 6x12 matrix G_s given in (27). There initial values are determined from step response and there values is

$$G_{s1} = \text{diag}(500 \ 500 \ 500 \ 500 \ 500 \ 500)$$

$$G_{s2} = \text{diag}(1 \ 1 \ 1 \ 1 \ 1 \ 1)$$

K_p and K_d are diagonal matrices of the proportional and derivative gains used in the nominal controller. Assuming a damping factor of 0.7 so that

$$K_d = 2\sqrt{K_p}$$

and The values used are

$$K_p = \text{diag}(4 \ 4 \ 4 \ 4 \ 4 \ 4) \times 10^4$$

$$K_d = \text{diag}(400 \ 400 \ 400 \ 400 \ 400 \ 400)$$

The simple sliding mode controller used for comparison has a sliding surface given by

$$S = G_{s1}e + G_{s2}\dot{e}$$

With G_{s1} and G_{s2} being as given above. For both simple sliding mode controller and integral sliding mode controllers, the gain of the switching control signal is calculated from the maximum values of uncertainties and is taken as

$$K = \text{diag}(6 \ 6 \ 6 \ 6 \ 6 \ 6) \times 10^3$$

The smoothness and magnitude of the control signal for different values of the sliding surface parameters and the switching control gain K is studied and the effect of adding the fuzzy friction estimator is investigated.

The tracking error in task space for the integral sliding mode controller and the other two controllers mentioned above is shown in Fig.6.8-Fig.6.25. The task space error is obtained using numerical estimation for forward kinematics. The results reveal that the task space tracking performance of the integral sliding mode controller with friction estimation is very much better than that of the simple sliding mode controller and the well known PID controller. The results are summarized in table 6.1 and table 6.2. The tables show no load tracking performance of the three controllers, which is also given in Fig.6.8-Fig.6.13. Generally, the no load tracking error improvement obtained by the integral sliding mode controller is more than 15 times compared with PID and 12 times compared with simple SMC. Especially the tracking error in the Z axis has been improved drastically and is in the order of μm . This indicates that the integral sliding mode controller is able to compensate gravitational torque better than the simple SMC and PID controllers. Compared with X and Y directions, the error in Z is small and this is due to the fact that the desired speed of vertical oscillation is small compared to x and y directions and tracking error generally increase when the speed of motion increase. But for the given frequency, the integral sliding mode controller is able to track the trajectory in all directions as shown in the other figures also.

The most important characteristic of the controller is its robustness against parameter variation and actuator friction. Robustness against parameter uncertainty is studied by varying the payload mass from no load to 200Kg and effect of actuator friction is studied by simulating viscous and coulomb frictions in the simmechanics model.

Table6.1 Task space % error comparison (no load and no friction case)

Controller parameters	Translational motion errors (%)	Rotational motion errors (%)	Remarks
PID $K_p=1 \times 10^6$ $K_d=5 \times 10^3$ $K_i=1 \times 10^3$	$E_y=3.65\%$ $E_x=3.5\%$ $E_z=2\%$	roll=1% pitch=1% $yaw=\pm 1 \times 10^{-3}$	Yaw angle error is absolute error
Simple SMC $G_{s1}=500$ $G_{s2}=1$ $K=6000$	$E_x=e_y=0.5\%$ $E_z=0.05\%$	Roll=0.133% Pitch=0.133% $Yaw=1.2 \times 10^{-3}$	Yaw angle error is absolute error
Integral SMC $G_{s1}=500$ $G_{s2}=1$ $K=6000$ $K_p=4 \times 10^4$ $K_d=400$	$E_x=e_y=0.25\%$ $E_z=0.05\%$	Roll=0.067% Pitch=0.067% $Yaw=2 \times 10^{-4}$	Yaw angle error is absolute error

Table6.2 Task space % error comparison (full load and friction case)

Controller parameters	Translational motion errors (%)	Rotational motion errors (%)	Remarks
PID K _p =1x10 ⁶ K _d =1x10 ⁴ K _i =1x10 ³	E _y =27.7 E _x =26.8 E _z =21.33	roll=21.2% pitch=25.4 yaw=±0.007	Yaw angle error is absolute error
Simple SMC Λ ₁ =500 Λ ₂ =1 K=6000	E _x =4% e _y =4.7% E _z =0.33%	Roll=1.27% Pitch=1.27% Yaw=2x10 ⁻³	Yaw angle error is absolute error
Integral SMC G _{s1} =1000 G _{s2} =1 K=6000 K _p =4x10 ⁴ K _d =400	E _x =e _y =0.31% E _z =0.0978%	Roll=0.202% Pitch=0.084% Yaw=1.78x10 ⁻⁴	Yaw angle error is absolute error

*G_{s1}, G_{s2}, K_p, K_d, K_i and K are 6x6 diagonal matrices and the values given are diagonal elements

As can be seen from Fig.6.14-Fig.6.19, the performance of the integral sliding mode controller is better than simple sliding mode and PID controllers. The result is also summarized in table II and it can be seen that in the case of PID controller and simple sliding mode controller, tracking error increased 9 times and 8 times respectively while the tracking error has increased by only 1.24 times in case of the integral sliding mode controller.

Another important performance measure to be noted, in addition to the tracking error, is the smoothness of the motion of the manipulator when it is carrying a load. Stewart platform manipulator is mostly used for precision applications and vibration of the platform when it is

carrying a load is undesired and a controller has to drive the manipulator smoothly. This smooth operation has also been achieved by the integral sliding mode controller. This can be seen from the smooth and bounded oscillation of the error signal as shown in Fig.6.14- Fig.6.19. The control torque applied at the joints for full load condition is shown in Fig.6.20- Fig.6.25. The simulation results show that the control signal in the integral sliding mode controller is smooth and chattering has been reduced.

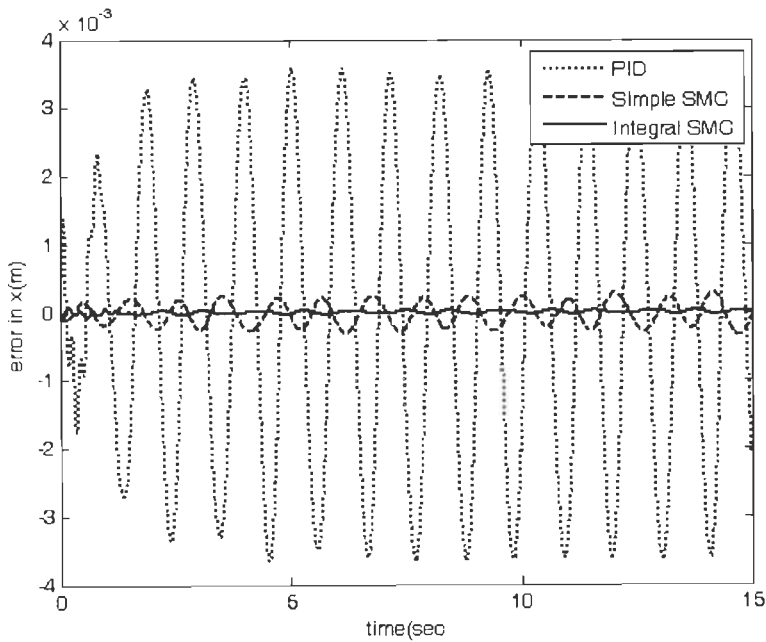


Fig.6.8 Task space error in x direction, without load

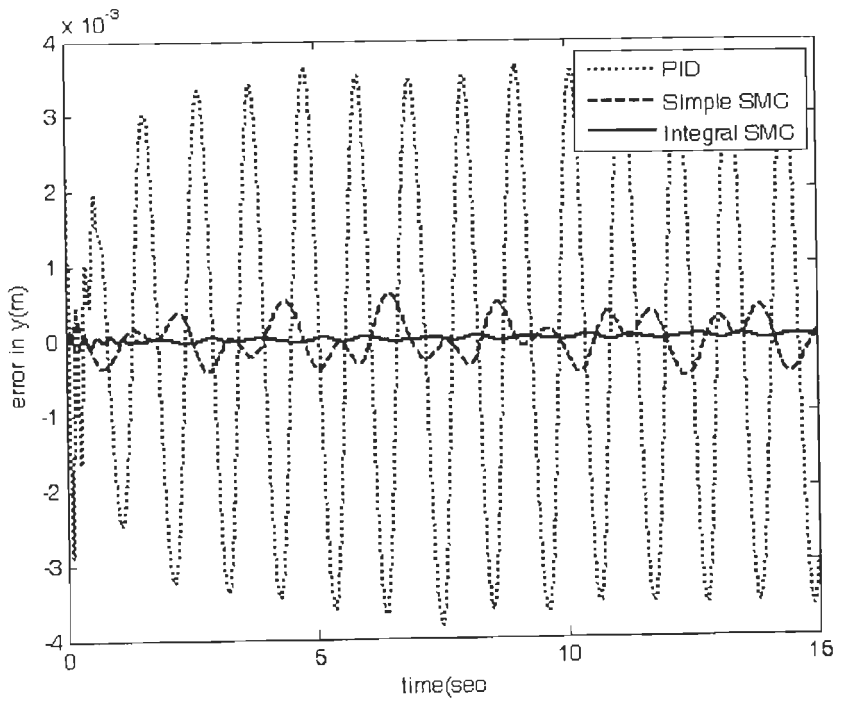


Fig.6.9 Task space error in y direction, without load

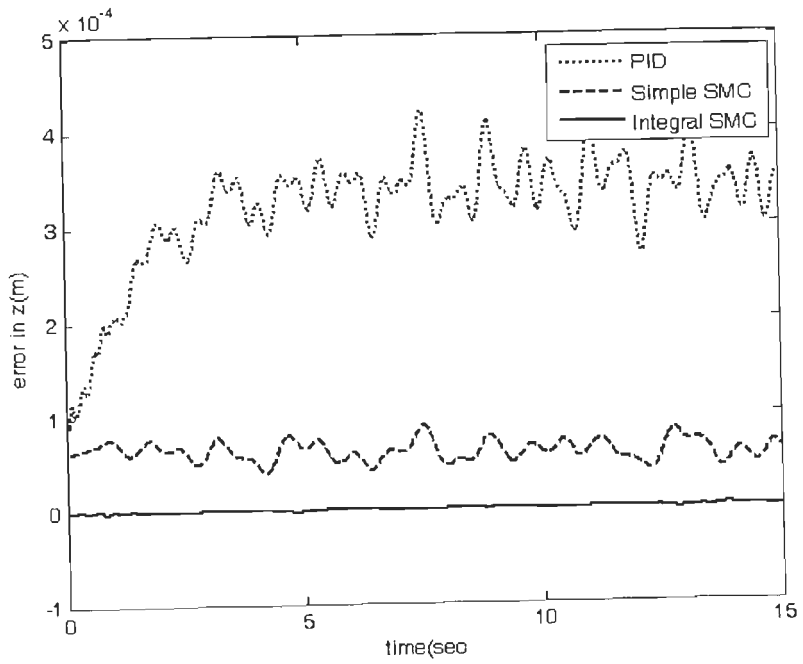


Fig.6.10 Task space error in z direction, without load

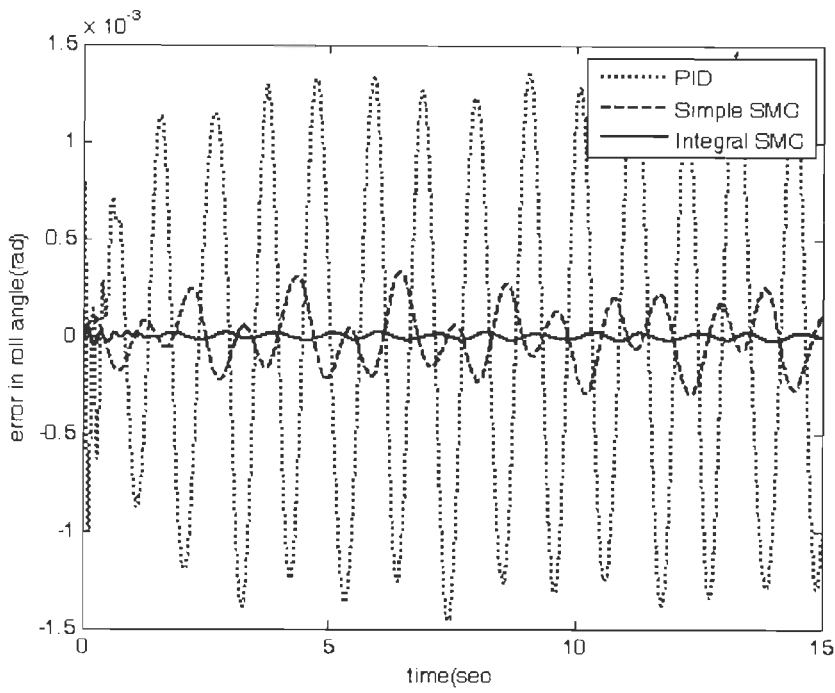


Fig.6.11 Task space error in roll angle, without load

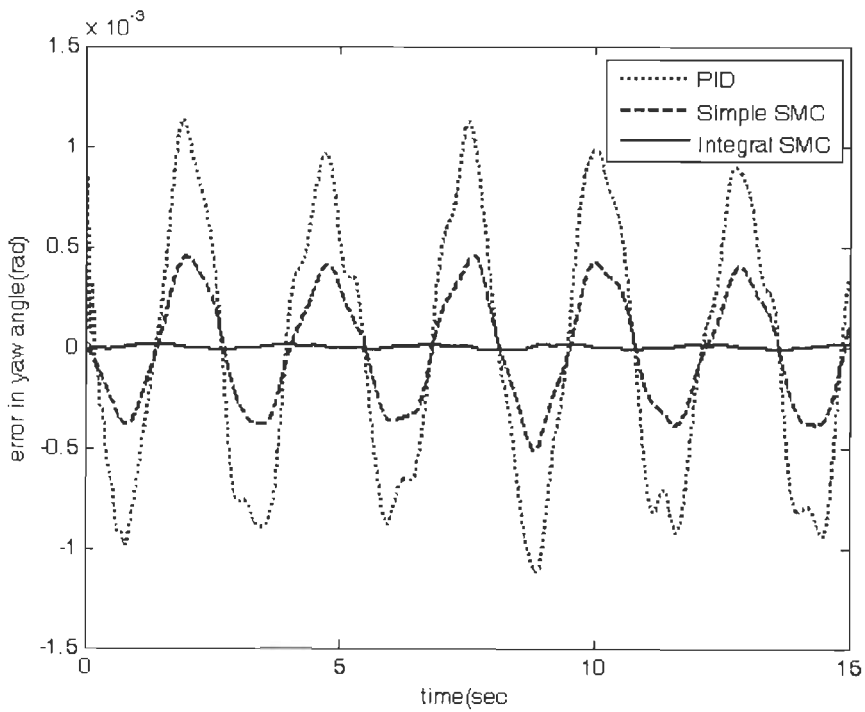


Fig. 6.12 Task space error in pitch angle, without load

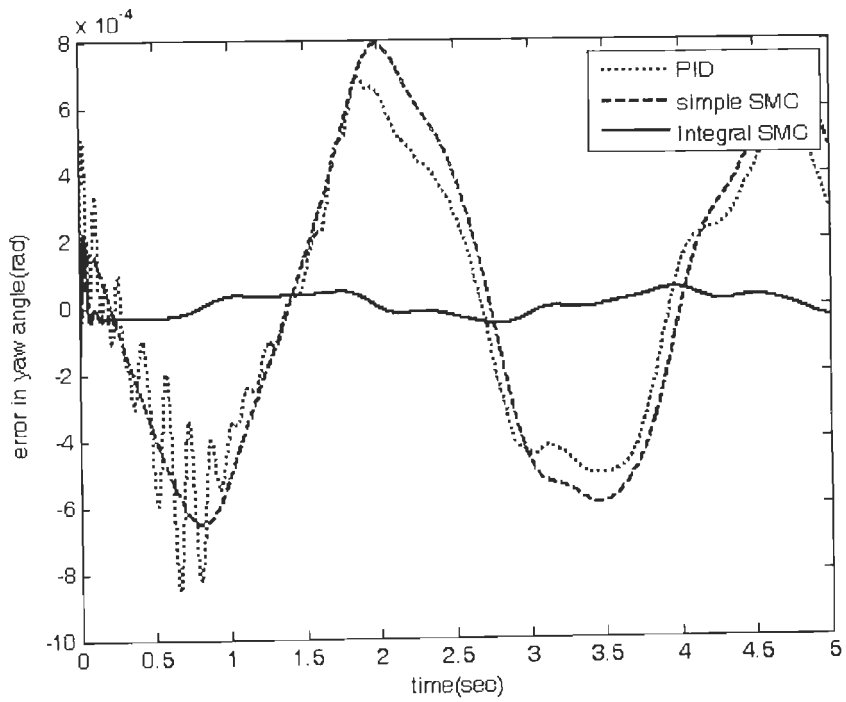


Fig.6.13 Task space error in yaw angle, without load

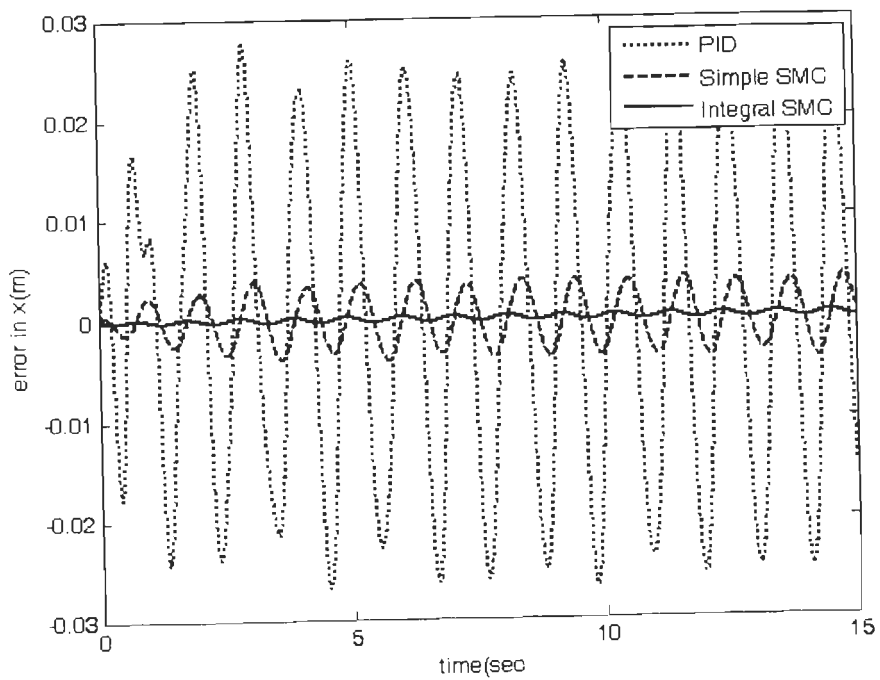


Fig.6.14 Task space error in x direction with 200 Kg load

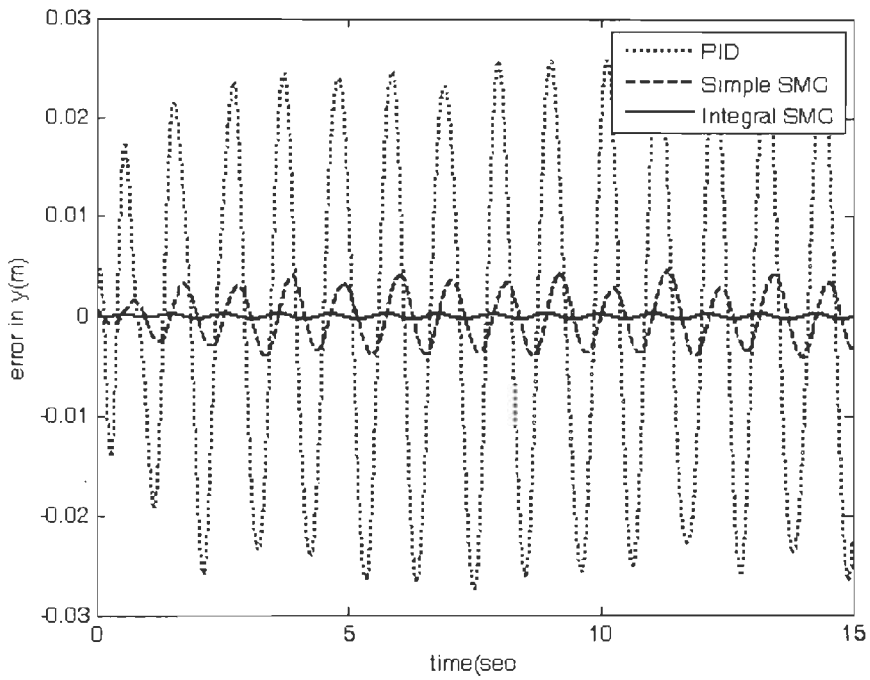


Fig.6.15 Task space error in y direction with 200 Kg load

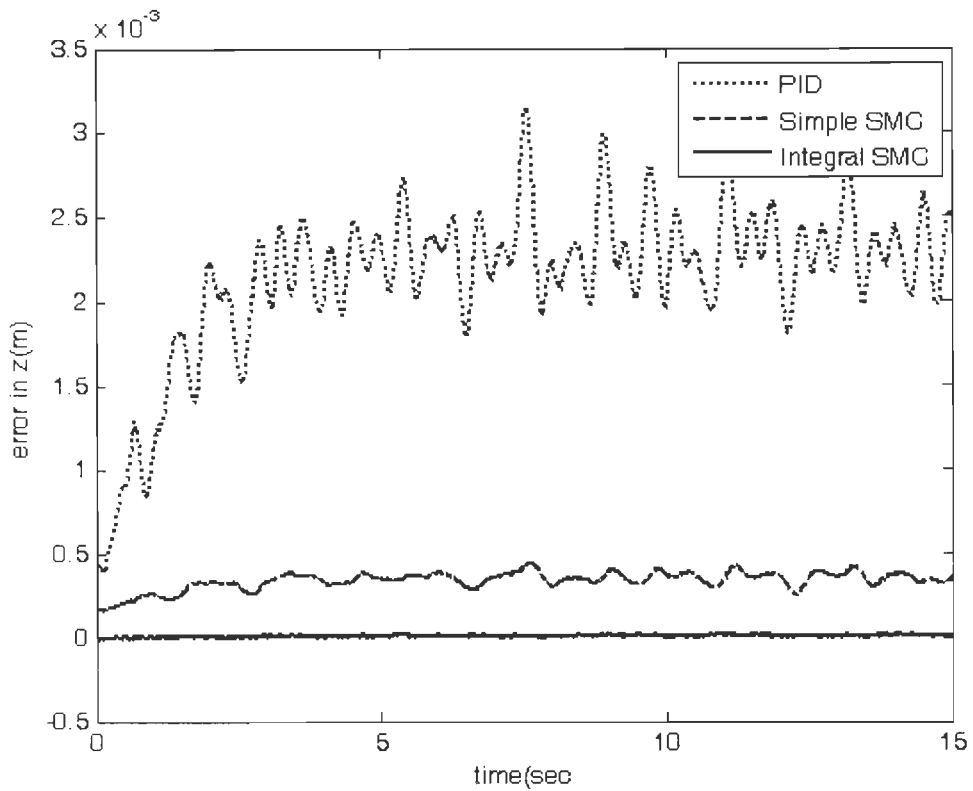


Fig.6.16 Task space error in z direction with 200 Kg load

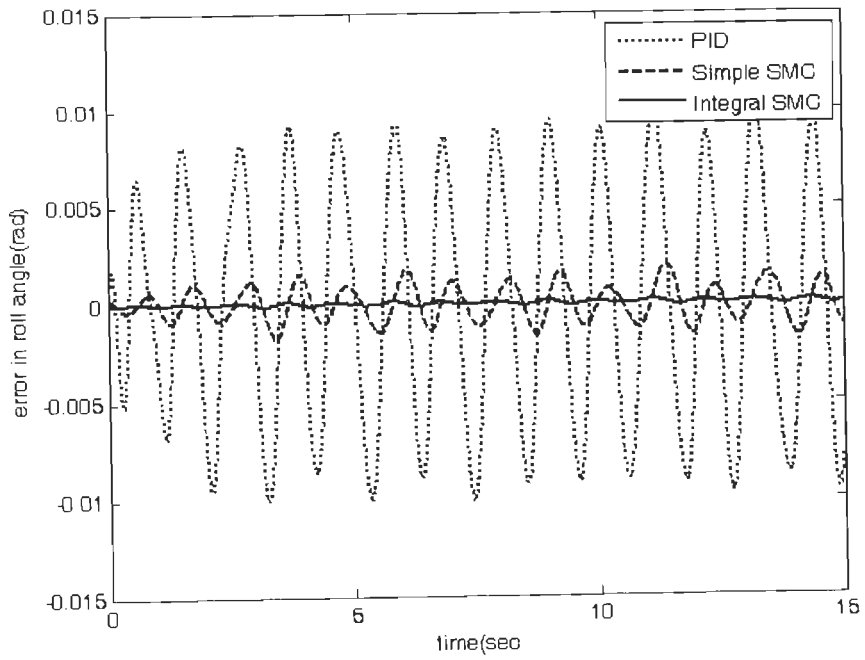


Fig.6.17 Task space error in roll angle with 200 Kg load

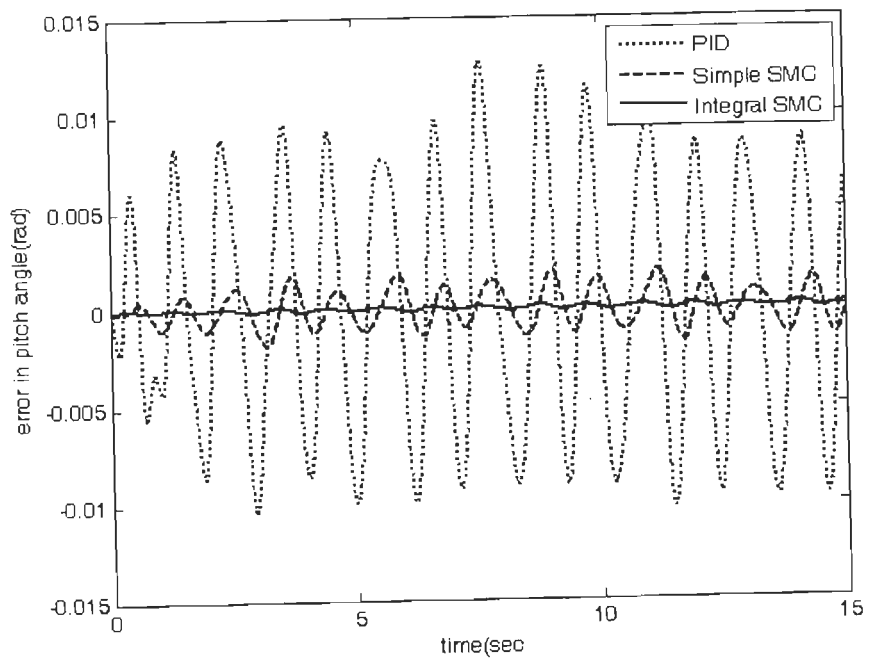


Fig.6.18 Task space error in pitch angle with 200 Kg load

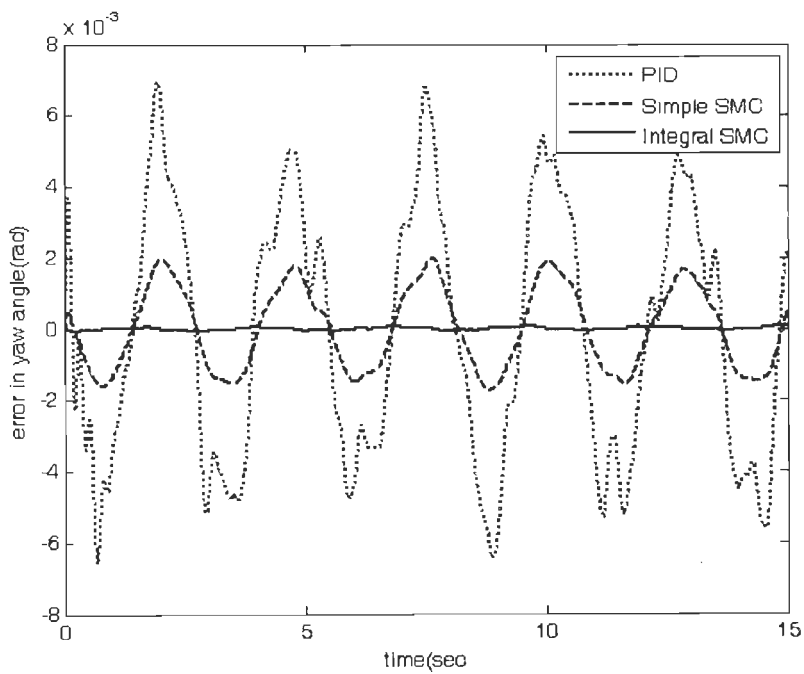


Fig.6.19 Task space error in yaw angle with 200 Kg load

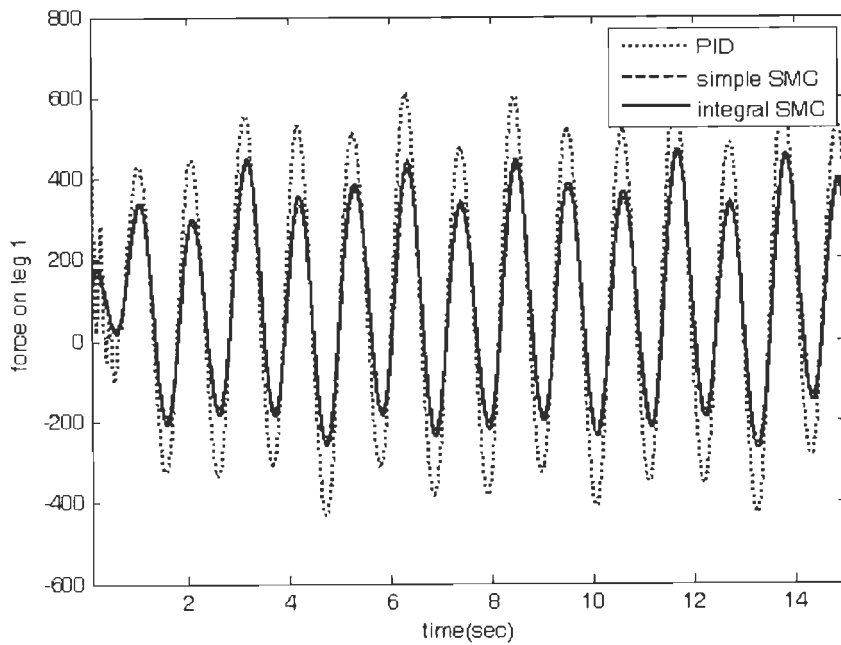


Fig.6.20 Control torque for leg 1 in the three controllers

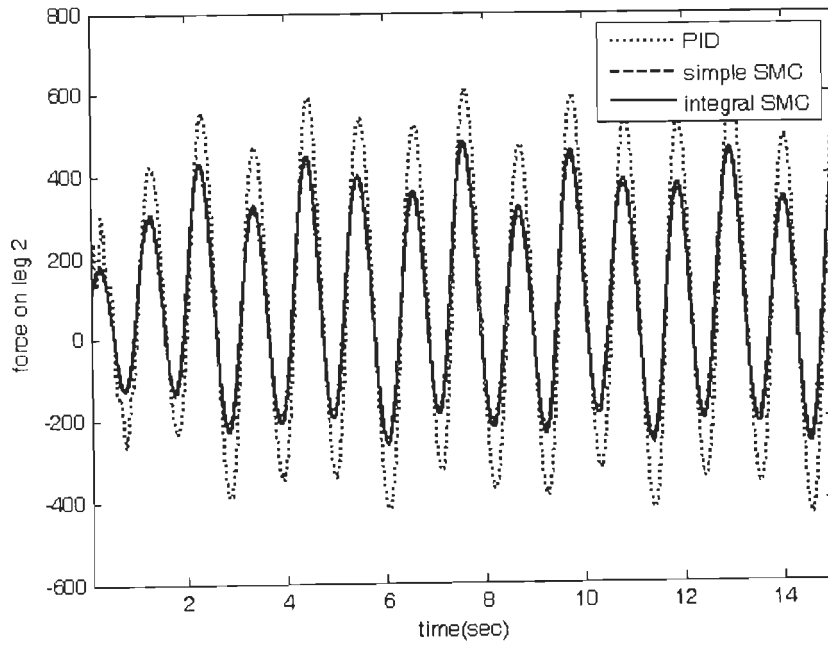


Fig.6.21 Control torque for leg 2 in the three controllers

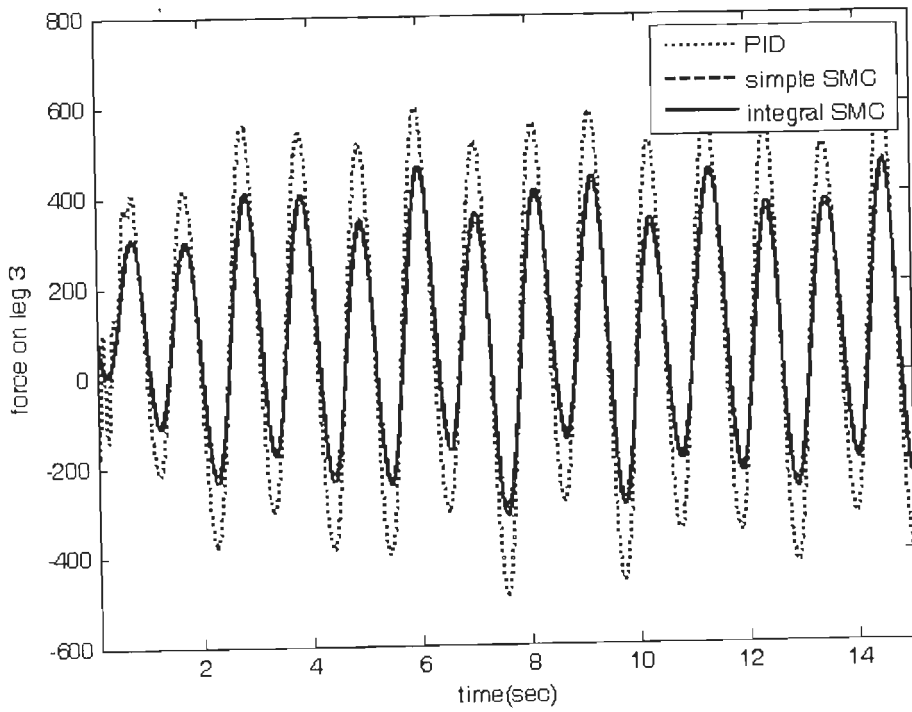


Fig.6.22 Control torque for leg 3 in the three controllers

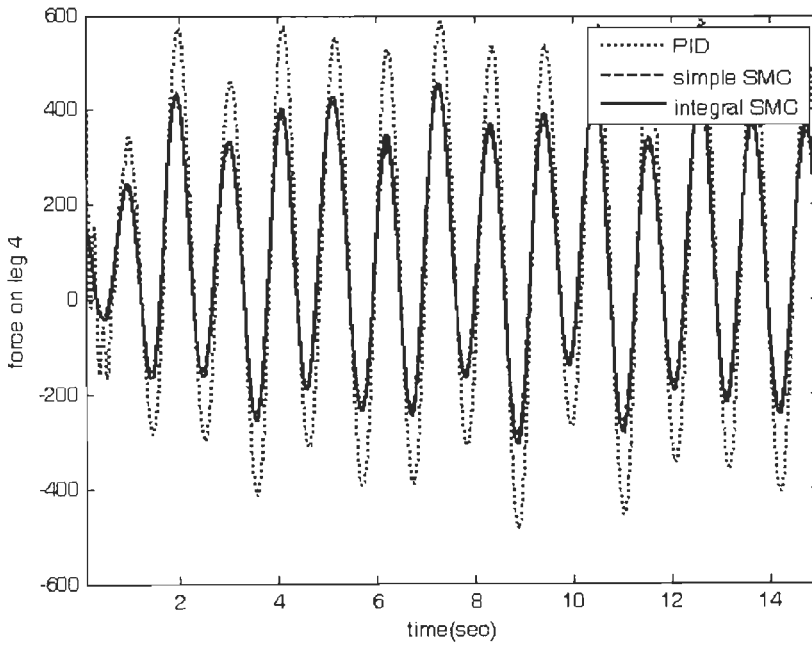


Fig.6.23 Control torque for leg 4 in the three controllers

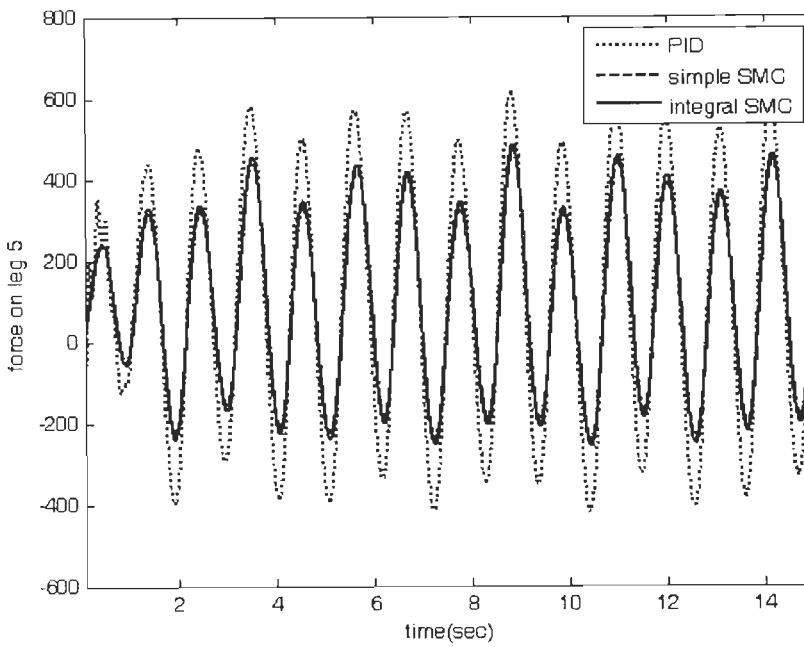


Fig.6.24 Control torque for leg 5 in the three controllers

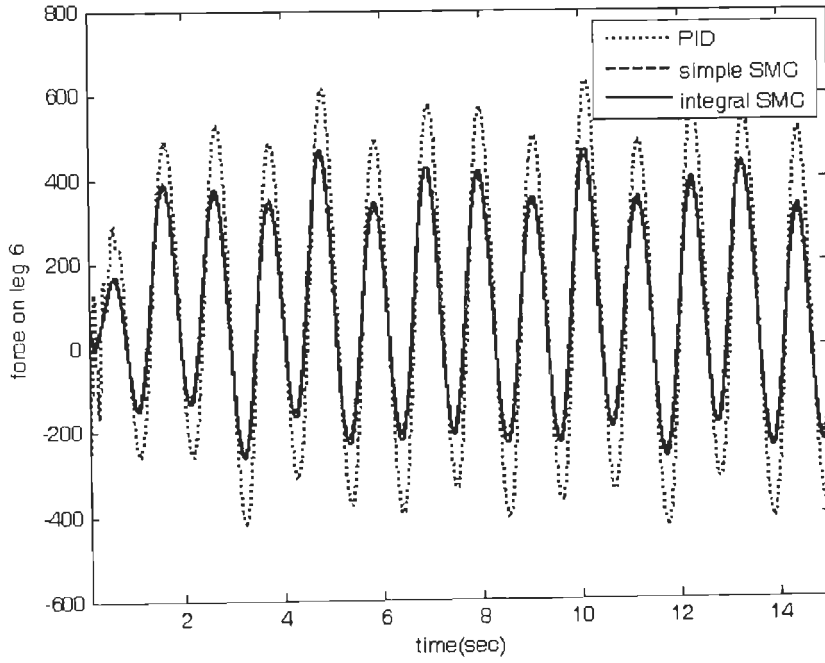


Fig.6.25 Control torque for leg 6 in the three controller

6.5.2. Integral sliding mode control for Stewart platform in task space

6.5.2.1. Design and analysis of the controller

Consider the task space dynamic equation given in 2.11,

$$M(X)\ddot{X} + C(X, \dot{X})\dot{X} + G(X) = J^T (\tau - f_f - \tau_d) \quad (6.42a)$$

The same assumptions are again used.

Assumption 6.1: The uncertainties in the inertia, Coriolis and centrifugal and gravitational matrices are bounded and can be given as nominal and deviation as:

$$M = M_N + \Delta M, \quad (6.42b)$$

$$C = C_N + \Delta C, \quad (6.42c)$$

$$G = G_N + \Delta G, \quad (6.42d)$$

with the perturbations ΔM , ΔC and ΔG having bounds, $\|\Delta M\| \leq M_m$, $\|\Delta C\| \leq C_m$, $\|\Delta G\| \leq G_m$.

Using (6.42b-d) into (6.42a), the dynamic equation can be rewritten as

$$M_N(X)\ddot{X} + C_N(X, \dot{X})\dot{X} + G_N(X) = J_p^{-T} (\tau - f_f - \tau_d) + h \quad (6.43)$$

where h is given by

$$h = -\Delta M\ddot{X} - \Delta C\dot{X} - \Delta G \quad (6.44)$$

Let X_d be (6×1) vector of desired task space trajectories. Then, the task space tracking error vector and its rate vector are given as

$$e = X_d - X \quad (6.45)$$

$$\dot{e} = \dot{X}_d - \dot{X} \quad (6.46)$$

Using (6.45) and (6.46) into (6.43) and changing to state space form, the error dynamics of the system in state space is given as

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= \ddot{X}_d - M_N^{-1} \left(J_p^{-T} (\tau - f_f - \tau_d) - C_N(X, \dot{X})\dot{q} - G_N(X) + h \right) \end{aligned} \quad (6.47)$$

Where $X_1=e$ is (6×1) state vector of Cartesian space tracking error in positions and orientations and $X_2=\dot{e}$ is (6×1) state vector of the Cartesian space velocity errors. Equation (6.47) can be rewritten as in (6.1) by using $x=[X_1 \ X_2]^T$, $x(0)=[X_1(0) \ X_2(0)]$ and rearranging the terms. This gives

$$f = \begin{bmatrix} X_2 \\ \ddot{X}_d + M_N^{-1} (C_N(X, \dot{X})\dot{X} + G_N) \end{bmatrix} \quad g = \begin{bmatrix} 0_{6 \times 1} \\ -M_N^{-1} J_p^{-T} \end{bmatrix} \quad (6.48)$$

and combining all the disturbance terms: parameter uncertainties, friction and external disturbance terms, the lumped disturbance term is given by

$$d = \begin{bmatrix} 0_{6 \times 1} \\ M_N^{-1} (-\Delta M\ddot{X} - \Delta C\dot{X} - \Delta G + J_p^{-T} (f_f + \tau_d)) \end{bmatrix} \quad (6.49)$$

In the case of the Stewart platform manipulator, in task space also the input matrix g is a function of the states as given by (6.48) and is not constant. Hence the method given in [42] cannot be used and hence we use genetic algorithm optimization to obtain G_s . This is done as follows. From (6.4), the integral sliding surface for the Stewart platform manipulator is given as

$$S = G_s \left(X - X_0 - \int_{t_0}^t (f(x, \tau) + g(x, v)\tau_0) d\tau \right), \quad (6.50)$$

where X_0 is the initial condition of the states and τ_0 is the nominal control torque obtained using the computed torque control method and is given by

$$\tau_0 = J_p^T \left(M_N (\ddot{X}_d + K_p X_1 + K_d X_2) + C_N (X, \dot{X}) \dot{X} + G_N \right) \quad (6.51)$$

In (6.51), K_p and K_d are 6x6 constant diagonal matrixes determined from stiffness of material and desired transient performance. Following a similar procedure as in section 6.3.2, the equivalent controller becomes

$$\tau_{eq} = \tau_0 + (G_s g)^{-1} G_s d \quad (6.52)$$

Using this equivalent controller into (6.30), the sliding mode dynamics of the system in task space becomes

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= \ddot{q}_d - M_N^{-1} \left(J_p^{-T} \left(\tau_0 + (G_s g)^{-1} G_s d \right) - C_N (q, \dot{q}) \dot{q} - G_N (q) \right) + d_{21} \end{aligned} \quad (6.53)$$

Substituting for the nominal controller from (6.51), the sliding dynamics becomes

$$\dot{X}_2 = -K_p X_1 - K_d X_2 - \left(d_{21} - M_N^{-1} J_p^{-T} (G_s g)^{-1} G_s d \right) \quad (6.54)$$

This shows that the uncertainty can be compensated and the sliding mode dynamics is stable if the gain matrix G_s is selected such that the last term in the bracket is made to be zero. However since the disturbance is not exactly known but only its bounds, the equivalent controller given by (6.52) cannot be realized. Moreover the value of the Jacobian matrix varies as the position of the manipulator varies. Therefore, the controller given by (6.52) is replaced by a switching function as follows

$$\tau = \tau_0 + J^{-T} K \frac{s}{\|s + \varphi\|} \quad (6.55)$$

Where φ is a small positive boundary value and K is chosen such that

$$K \geq \|d_m\| = \left\| M^{-1} \left(\Delta m_m \ddot{X} + \Delta C_m \dot{X} + \Delta G_m \right) \right\| \quad (6.56)$$

The nominal control signal τ_0 is calculated from the unperturbed model of the system in a feed forward manner as given in (6.52).

6.5.2.2. Simulation study and discussion

The controller given by (6.55) is implemented using MATLAB and Simulink. The integral sliding surface (6.50) is implemented after it is rewritten as in (6.52) by substituting the nominal controller (6.52) and system dynamics functions into (6.51). The last expression

after the substitutions is:

$$s = G_s \left\{ \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} - \begin{bmatrix} e_1(0) \\ e_2(0) \end{bmatrix} - \begin{bmatrix} \int_0^t e_2 d\omega \\ \int_0^t (-K_p e_1 - K_d e_2) d\omega \end{bmatrix} \right\} \quad (6.57)$$

where G_s is a 6x12 matrix to be determined using the multi objective genetic algorithm of section 6.4.1 and K_p and K_d are 6x6 diagonal matrixes used in the nominal control signal. The gain matrix G_s can be partitioned into two 6x6 square matrices as $G_s = [G_1 \ G_2]$. Without loss of generality, we can assume G_{2s} to be identity matrix and G_{1s} to be a diagonal matrix. This helps in reducing the number of parameters to be optimized by the genetic algorithm and also reduces the time required for computation. Moreover, it decouples the six sliding surfaces and speed up the computation of control signal. Then (6.52) can be rewritten as

$$S = G_{1s} X_1 + X_2 - G_{1s} X_1(0) - X_2(0) + \int_0^t (-G_{1s} X_2 + K_p X_1 + K_d X_2) d\omega \quad (6.58)$$

The range of values used in the genetic algorithm optimization for parameters of the integral sliding surface and gains are determined using step response of each dimension. The controller is first used for single direction regulation control and the step response is observed. Then the controller gains are tuned until a desired step in terms of settling time, overshoot and steady state error is obtained. Then the controller parameters used to obtain the best and worst step responses are used as range of values in the genetic algorithm optimization. Step response obtained when diagonal values of G_{1s} are 200 is shown in Fig.6.14. Hence, the diagonal elements of G_1 are allowed to vary between 100 and 400 and the gain value K_1 of the switching function is allowed to vary between 5000 and 15000. The parameters used in the genetic algorithm are as follows.

Initial population 20

Crossover rate =0.8

Mutation rate =0.01

Maximum number of generations 20

With these parameters, one of the Pareto optimal solutions obtained using multi-objective genetic algorithm is given below.

The parameters of the integral sliding surface are

$$G1 = \text{diag}(117 \ 113 \ 87 \ 155 \ 157 \ 150) \quad (6.59)$$

And the gain value used for the switching function is

$$k = \text{diag}(7983 \quad 7218 \quad 13074 \quad 6563 \quad 8468 \quad 6739) \quad (6.60)$$

The K_p and K_d values used in the nominal controller are

$$K_p = \text{diag}[4 \quad 4 \quad 4 \quad 4 \quad 4 \quad 4] \times 10^4 \quad (6.61)$$

$$K_d = [2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2] \times 10^2 \quad (6.62)$$

The tracking performance of the controller for no load and loaded cases with friction and back lash considered are given in Fig.6.26-Fig.6.40. To compare the performance of the task space integral sliding mode controller with existing standard controller, simple sliding mode controller is implemented. The best gain values obtained and the simulation outputs are compared. From the figures one can observe that, the tracking performance of the integral sliding mode controller is far better than simple SMC not only in terms of the tracking accuracy but also in terms of the smoothness of the control signals. Figures 6.26-Fig.6.31 show the no load task space tracking performance of the ISMC and simple SMC in the 6 DOF's for the given trajectory. In all cases the error is not asymptotically decaying but is bounded. In the case of the integral sliding mode controller, the error is much smaller. Especially the integral SMC is able to compensate gravitational torque which can be easily seen from the tracking performance in z direction. In all directions, the tracking error for the integral SMC is below 0.05mm and considering the speed of operation taken, which is 600mm/sec, this tracking error is very good. The orientation angle errors are slightly greater than the translational motion error of x, y and z. This is due to the error in the numerical algorithm. The initial guess taken for the numerical algorithm is same for all trajectory positions.

The performance of the controller for loaded conditions is also checked using a 200Kg load. The result is shown in Figs.6.32-Fig.6.38. From the figures it can be seen that the controller performance has not been degraded much, even in the presence of 200Kg load, simulated friction torque in actuator and random external disturbance torque. The tracking error is bounded within ± 0.1 mm. The most important aspect in control of Gough-Stewart platform manipulator carrying a load is to maintain stiffness and avoid vibration. The results show that the system is moving without vibration as the tracking error is smooth. The control force of the task space controller, after it is transformed to joint space, is shown in Fig.6.39

and Fig.6.40. It is clear that the magnitude of the control torque is reasonable and is also smooth without any chattering. The initial swing is having frequency less than 50Hz and can be implemented by actuator motors.

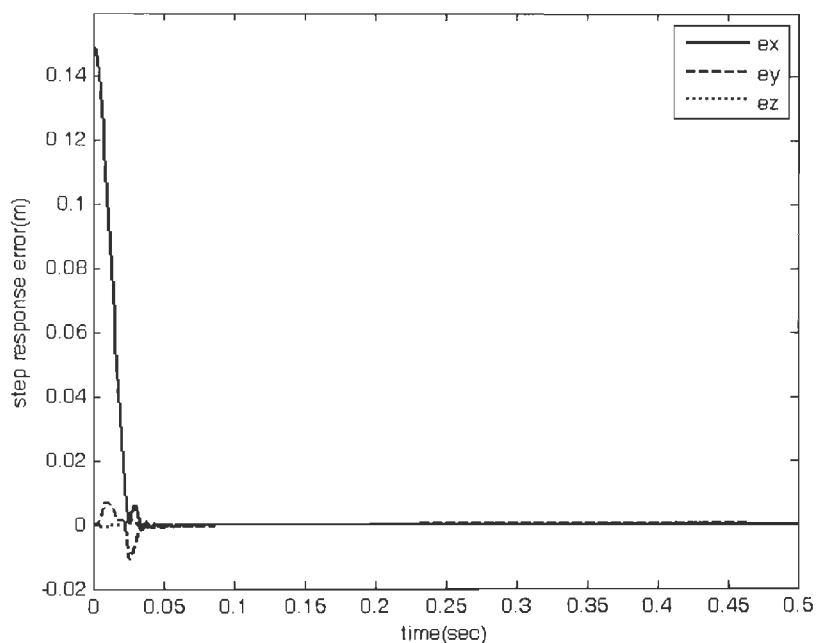


Fig. 6.26 Step response error in x direction used for parameter selection, desired is rise time less than 50msec with no overshoot

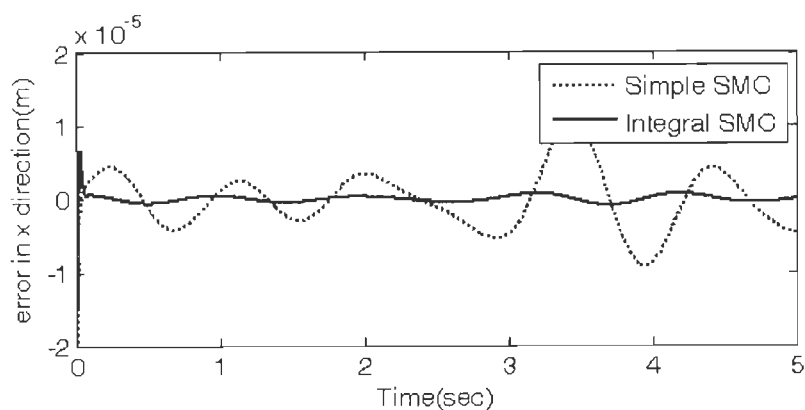


Fig.6.27 Trajectory tracking performance in x direction of simple task space sliding mode controller and integral sliding mode controller with no load

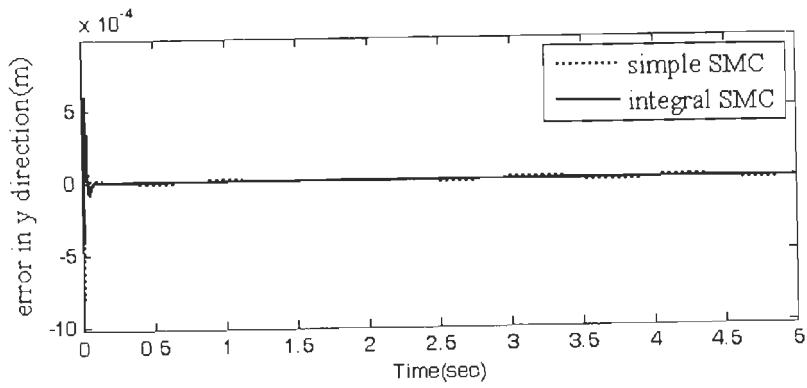


Fig.6.28 Trajectory tracking performance in y direction of simple sliding mode controller and integral sliding mode controller with no load

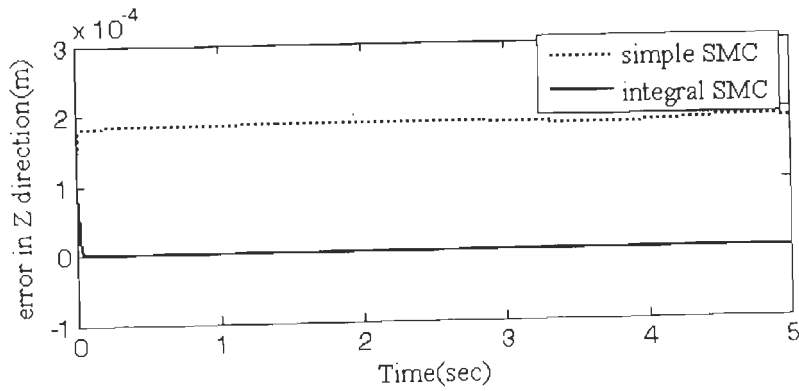


Fig.6.29 Trajectory tracking performance in z direction of the simple task space sliding mode controller integral sliding mode controller with no load

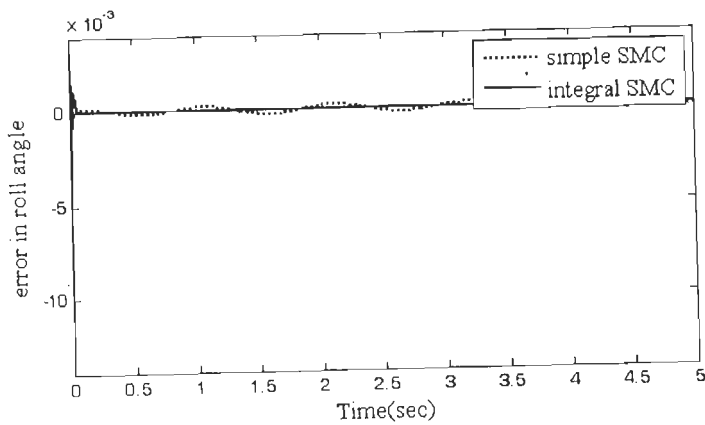


Fig.6.30 Trajectory tracking performance for roll angle of simple sliding mode controller and integral sliding mode controller with no load

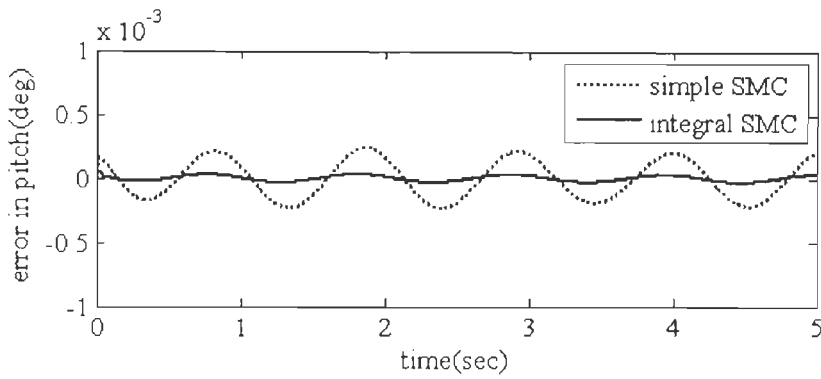


Fig.6.31 Trajectory tracking performance for pitch angle of simple sliding mode controller and integral sliding mode controller with no load

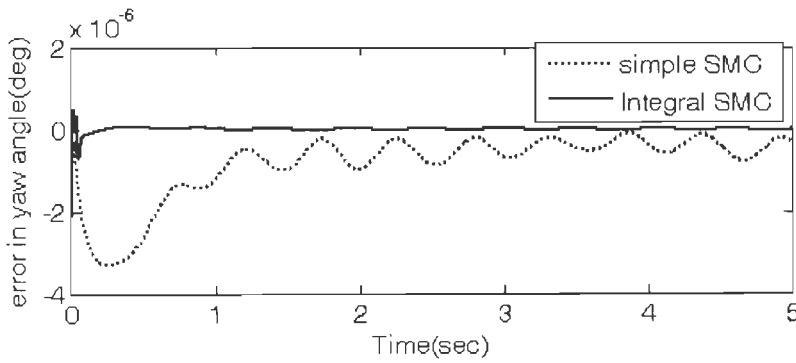


Fig.6.32 Trajectory tracking performance for yaw angle of simple task space sliding mode controller and integral sliding mode controller with no load

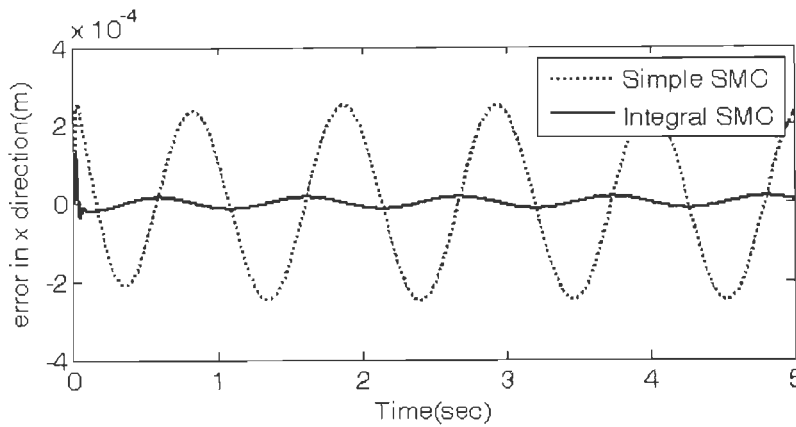


Fig.6.33 X direction trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg

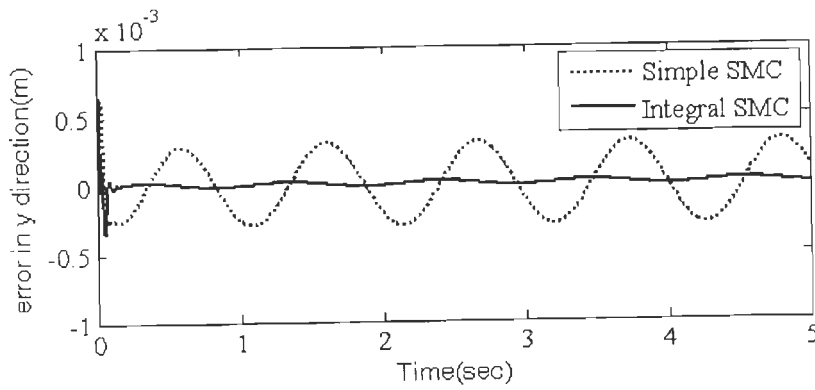


Fig.6.34 Y direction trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg

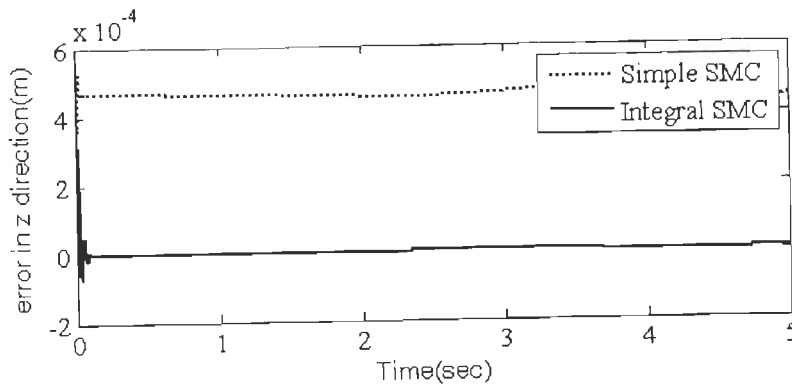


Fig.6.35 Z direction trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg

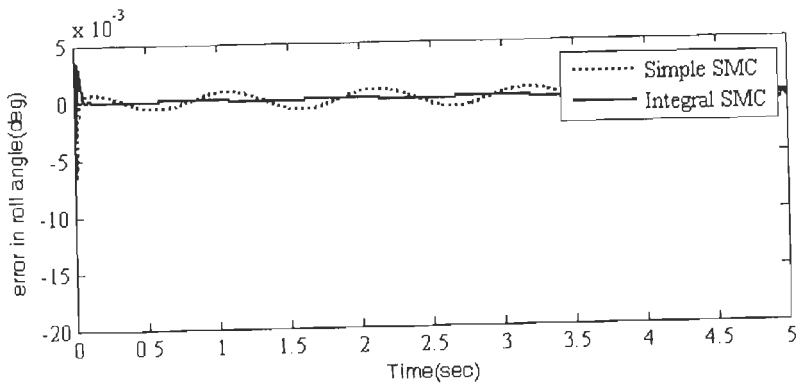


Fig.6.36 Roll angle trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg

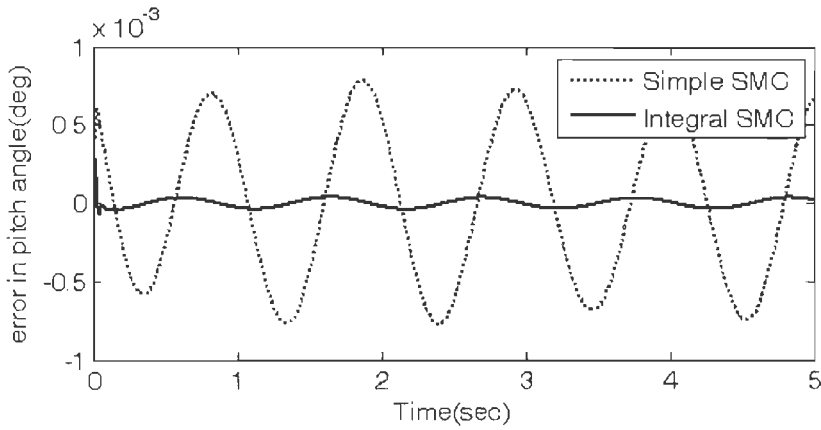


Fig.6.37 Pitch angle trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg

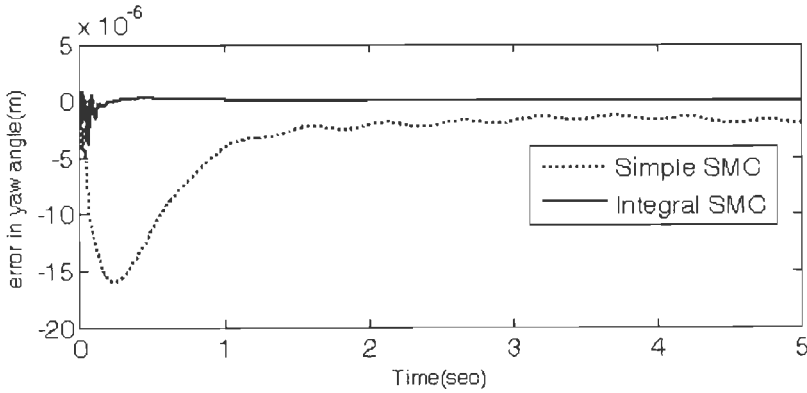


Fig.6.38 Yaw angle trajectory tracking performance of simple task space sliding mode controller and integral sliding mode controller when carrying load of 200Kg

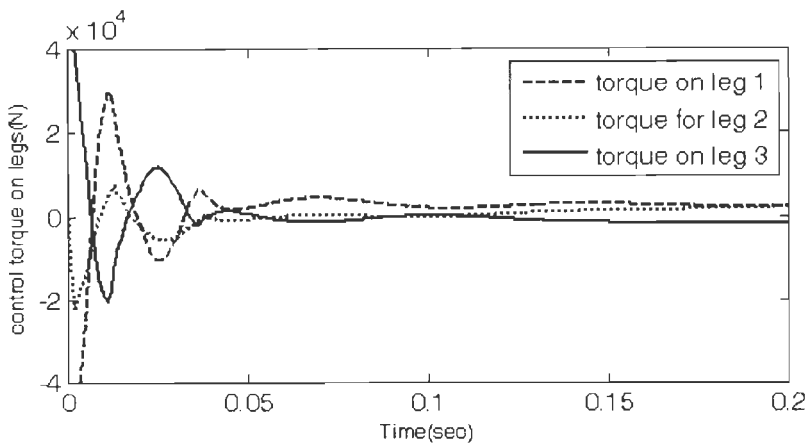


Fig.6.39 Control forces in joint space

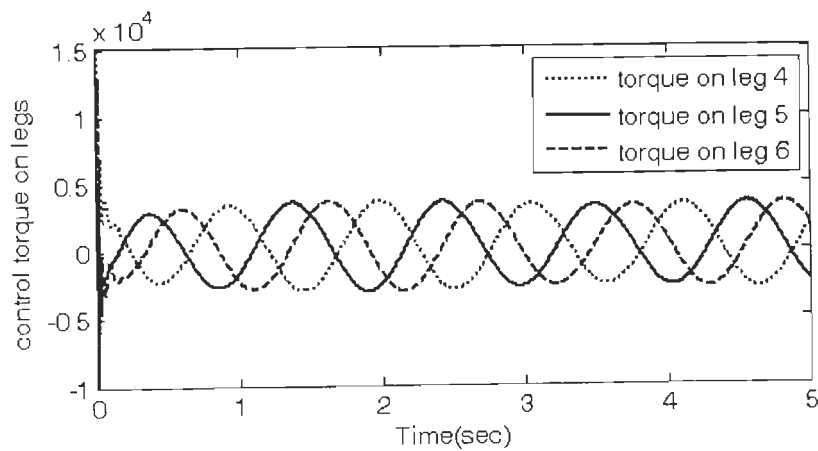


Fig.6.40 Control forces in joint space

6.6 Comparison between joint space and task space ISMC

The results of the integral sliding mode controllers designed in joint space and task space have been in Fig.6.8-Fig.6.25 and Fig.6.26-Fig.6.40 respectively. In both cases the trajectory tracking results are given in task space and the control forces are given in joint space. Actually, the nominal control part of the joint space ISMC has been implemented in task space and only the switching controller is implemented in joint space. This is done to make the simulation more close to practical implementation since implementing dynamic parameters in joint space is difficult and in practical application it is not used. Another point to be noted when comparing the results is the number of controllers compared in each case. In the case of the joint space three controllers are compared: PID, simple SMC and integral SMC. But in the task space, only two controllers are compared. This is so because, due to the big difference between the PID and task space integral SMC, trying to plot them together makes the comparison between simple SMC and integral SMC invisible. So in the task space case only two controllers are plotted.

Comparing the respective trajectory tracking performances, the task space controller generally shows better tracking performance than the joint space one. The order of tracking errors is in micro. This is as expected. But a point to be noted is the magnitude of the control force in the two cases. The joint space control signals are very small. This shows that the joint space approach has more practical advantages than the task space. As stated above, in

reality the joint space is a hybride of task space and joint space and it has better result than the purely task space implementation.

6.7 Conclusion

In this chapter, the design of integral sliding model controller using genetic algorithm and we implemented the method for simple SISO system and for a 6 DOF Stewart platform manipulator. The details of the design of the controller are given and its stability has been analyzed. The comparison between the LMI method and the genetic algorithm based methods shows that genetic algorithm design method gives better results. Because of its easiness and considering the results obtained, we can conclude that genetic algorithm can be effectively used for the design of integral sliding mode controller for any system.

NEURO-FUZZY SLIDING MODE CONTROLLER

In the previous chapters, we have seen various types of sliding mode controllers used together with genetic algorithm and fuzzy logic system. The improvements obtained are encouraging. In this chapter we will discuss a more robust and better controller, which is made from hybrid of fuzzy and neural networks. In the following sections, the design and stability analysis of neuro-fuzzy sliding mode controller is discussed. The controller has two parts: fuzzy logic system and neural network. They are used concurrently but each part is responsible for one phase of sliding mode controller. The fuzzy logic system is utilized to control reaching phase dynamics and the feedforward neural network is employed to keep the system states on the sliding surface. The neural network is trained online using modified back propagation algorithm. Initially, fuzzy logic system is dominant and as the system moves from reaching phase to sliding phase, neural network becomes more active and hence a hybrid computing paradigm is achieved. The stability of the system is analyzed using Lyapunov's direct method. The proposed controller is implemented to regulate a second order nonlinear uncertain system and simulation results confirm that the proposed system reduces chattering and improves transient response.

7.1. Neuro-fuzzy hybridization and sliding mode control

It is very well known that fuzzy logic systems are very good in decision making, while neural networks are good at function approximation. Therefore, combining neural and fuzzy logic systems with conventional sliding mode controllers can result in a more robust controller. For this kind of research, the studies of [24][77] and [102] are worth mentioning. In [24] and [77], a fuzzy logic system had been used to replace the switching function, while in [102] fuzzy logic system was used to generate nonlinear sliding surface. Similarly, in [149], a neural network was employed to estimate an optimal sliding surface and in [148], a radial basis function network was utilized to vary the gain of the switching function of a sliding mode controller. In all the above studies, either neural network or fuzzy logic system was used to improve the performance of basic sliding mode controller. But none of the implementations could solve the chattering

problem completely. This is because using fuzzy logic to replace switching function results in a similar effect to that of saturation control. Therefore, a more important development could be the use of two artificial intelligence techniques together.

In an attempt to use the said two artificial intelligence techniques with sliding mode controller, neuro-fuzzy sliding mode controllers had been proposed [2][3][122]. In [122], a neuro-fuzzy system was used to approximate equivalent control part of conventional sliding mode controller but the switching function part was the same with the conventional one. The work reported by [2] used an ANFIS network to replace the switching function which resulted into adjustable boundary layer but in the same paper, it was reported that the system was sensitive to parameter variation and external disturbance. In the work of [3], the controller had two parts: a neural network part which was employed to estimate the equivalent control signal and a fuzzy logic system, which was utilized as switching function. The output of the fuzzy logic is used to train the neural network.

Our approach is similar to the work of [3]. But, there are three main differences. First, in the former case, the signal used to train the neural network was the output of the fuzzy logic. This might lead to a delay in synchronizing the outputs of the neural network and fuzzy logic systems. In our proposal, we have used the sliding variable to train the neural network. The second main difference is in the sliding surface used. In [3], a nonlinear surface was used, while in the present case a linear surface was used. One basic advantage of sliding mode controllers is the system order reduction achieved when the sliding surface is linear. Moreover, a linear sliding surface gives a stable and robust performance with respect to parameter variation and external disturbance, which is the basic objective of sliding mode control. The third and main difference lies in the fact that our system is designed for a general second order nonlinear uncertain system, while the work of [3] is shown for a specific system.

Therefore, the objective of this thesis is to design and analyze the stability and performance of a neuro-fuzzy sliding mode controller for a general second order uncertain system. In the controller, neural network and fuzzy logic system are used concurrently but each part is mainly responsible to control one of the two phases of sliding mode controller. The fuzzy logic system was utilized as a switching function to control the reaching phase dynamics and the neural network was used to estimate the equivalent

control signal. Initially fuzzy logic system is dominant and drives the system towards the sliding surface. As the system moves from reaching phase to sliding phase, the output of the fuzzy decreases and the neural network weights are adjusted so that an exact equivalent controller is estimated to keep the system on the sliding surface. The neural network learning rate is inversely related to the sliding parameter and therefore the neural network becomes more dominant in the controller in the sliding phase. Because of this smooth transformation in control action, a hybrid computing structure is achieved. The sliding surface used is linear and the online weight adjustment is done using modified back propagation algorithm. Stability analysis and simulation results have shown that the system is stable and has superior performance than other types of sliding mode controllers.

7.2. Problem statement

Consider a general second order uncertain nonlinear system given by

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= (f + \Delta f) + (g + \Delta g)(u + d(t)),\end{aligned}\tag{7.1}$$

where $X = [x_1 \ x_2]^T$ is the state vector, f and g are nonlinear functions in R^2 , u is the control input, Δf and Δg are bounded uncertainties and d is disturbance. The following assumptions are taken.

Assumption 1: The nominal system

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x) + g(x)u\end{aligned}\tag{7.2}$$

is known and stabilizable.

Assumption 2: The uncertainty Δf is matched and Δf , Δg and disturbance signal d are assumed to have the following bounds.

$$\begin{aligned}\|\Delta f\| &\leq f_m \\ \|\Delta g\| &\leq g_m \\ \|d\| &\leq d_m\end{aligned}\tag{7.3}$$

Then, the objective here is to show that the control law,

$$u = u_{NN}(X, x_d) + u_f\tag{7.4}$$

u_{NN} being the equivalent control signal obtained from neural network and u_f being the output of fuzzy logic system, are smooth and stabilize system (7.1) with tracking error decaying asymptotically to zero as given below.

Consider a smooth and twice differentiable trajectory x_d , then the trajectory tracking error e is given by

$$e = x_d - x_1. \quad (7.5)$$

Consider also a linear sliding surface given by

$$s = \lambda e + \dot{e}, \quad (7.6)$$

where λ is a strictly positive design parameter. The derivative of the sliding surface s will be

$$\dot{s} = \lambda \dot{e} + \ddot{e}. \quad (7.7)$$

From (7.5), $\ddot{e} = \ddot{x}_d - \ddot{x}_1$ and using (7.1) $\ddot{e} = \ddot{x}_d - \dot{x}_2$. Substituting for \dot{x}_2 from (7.1), (7.7) can be written as

$$\dot{s} = \lambda \dot{e} + \ddot{x}_d - (f + \Delta f + (g + \Delta g)u + d(g + \Delta g)) \quad (7.8)$$

On the sliding surface both s and its time derivative are zero and hence the equivalent controller is:

$$u_{eq} = (g + \Delta g)^{-1} (\lambda \dot{e} + \ddot{x}_d - f - \Delta f - d(g + \Delta g)). \quad (7.9)$$

This equivalent controller cannot be generated directly because of the unknown uncertainties. But (7.9) can be considered as a mapping from $\mathfrak{R}^2 \rightarrow \mathfrak{R}$. It is to be noted that neural networks are universal approximators which can learn such mappings. We can readily generate the equivalent control signal using a neural network and adjust its weights online such that the control signal keeps the system on the sliding surface. Hence, the equivalent controller can be written as follows:

$$u_{eq} = u_{NN} = f_o \left(\sum_{j=1}^N W_{jo}^h f_j \left(\sum_{i=1}^p W_{ij}^l X_i + b_j \right) + b_o \right), \quad (7.10)$$

where N is the number of hidden layer neurons, p is the number of input units, b_j is the bias for hidden layer neurons, b_o is the bias of the output neuron, f_j is the activation function of hidden layer neurons, W^h and W^l are ideal network connection weight matrices for hidden layer to output and input to hidden layer, respectively. The universal approximation theory

confirms that we can obtain weight matrices, \widehat{W}^h and \widehat{W}^l , which are very close to the ideal weight matrices such that (7.10) sufficiently approximates (7.9). This means that there exists a small real number ε , such that

$$(g + \Delta g)^{-1} (\lambda \dot{e} + \ddot{x}_d - f - \Delta f - d(g + \Delta g)) = f_o \left(\sum_{j=1}^N W_{jo}^h f_j \left(\sum_{i=1}^p W_{ij}^l X_i + b_j \right) + b_o \right) + \varepsilon \quad (7.11)$$

The weight estimation error is given by

$$\begin{bmatrix} \widetilde{W}_1 & \widetilde{W}_2 \end{bmatrix} = \begin{bmatrix} W^l & W^h \end{bmatrix} - \begin{bmatrix} \widehat{W}^l & \widehat{W}^h \end{bmatrix} \quad (7.12)$$

This implies that, if the weight estimation error is zero, the estimated weight matrices will be the same as the ideal weight matrices and the neural network estimation error reduces to zero. This confirms that the neural network gives the desired equivalent control signal, which will keep the system on the sliding surface rejecting all disturbances. The complete block diagram of the controller system is shown in Fig. 1 and stability and reaching condition of the system under the given control law (7.4) is analyzed as follows.

7.3. Stability Analysis

Let us chose a Lyapunov function of the form:

$$V = \frac{1}{2} s^2 \quad (7.13)$$

Then, its time derivative will be

$$\dot{V} = s \dot{s} \quad (7.14)$$

Using (7.1), (7.5) and (7.6), we have

$$\dot{V} = s (\lambda \dot{e} + \ddot{x}_d - f - \Delta f - (g + \Delta g)(u_{NN} + u_f) - d(g + \Delta g)) \quad (7.15)$$

Now, using (7.11) into (7.15), we can rewrite

$$\dot{V} = s (\lambda \dot{e} + \ddot{x}_d - f - \Delta f - (g + \Delta g) \left((g + \Delta g)^{-1} (\lambda \dot{e} + \ddot{x}_d - f - \Delta f - d(g + \Delta g)) - \varepsilon + u_f \right) - d(g + \Delta g)) \quad (7.16)$$

Where ε is the neural network estimation error. After simplifying the terms, we have

$$\dot{V} = -s ((g + \Delta g)(\varepsilon + u_f)) \quad (7.17)$$

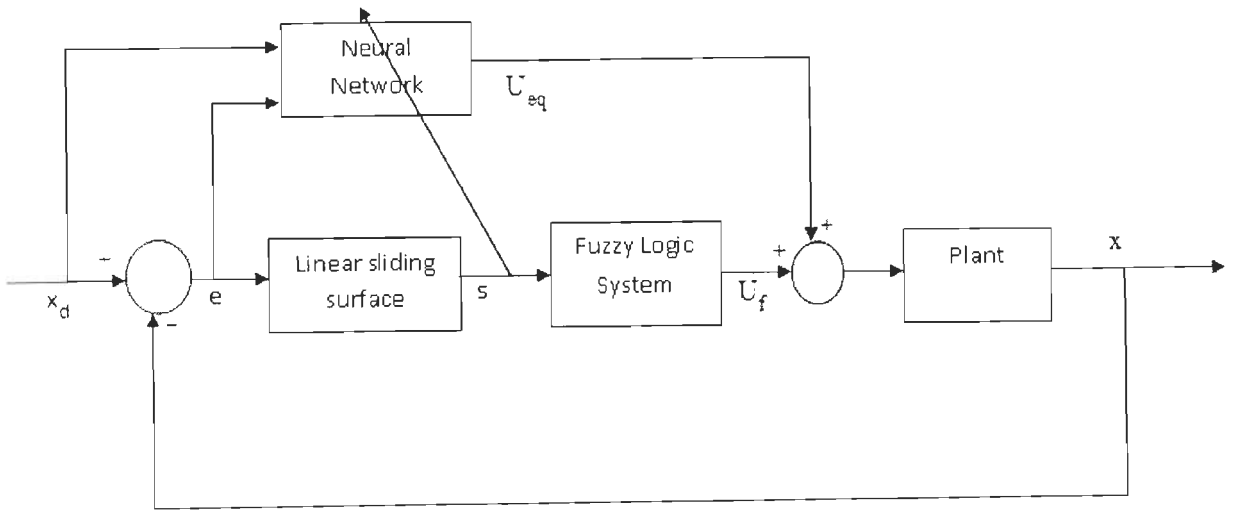


Fig.7.1 Structure of the neuro-fuzzy controller

From (7.17), we can see that $\dot{V} < 0 \forall t > 0$, if the neural network is trained to approximate the equivalent controller with minimum error so that $\|\varepsilon\| < \|u_f\|$. This can be achieved if the fuzzy logic output is designed to fulfill

$$\|u_f\| \geq f_m + g_m d_m + \|g\| d_m \quad (7.18)$$

7.4. Neural network equivalent control estimation

Neural networks are massively parallel distributed processing systems made up of highly interconnected processing elements that have the ability to learn and acquire knowledge. They have been used for function approximation and pattern classification problems. Recently, a radial basis function network has been used in conventional sliding mode controller to estimate the gain of the switching function. The motivation for the use of neural networks in sliding mode control is their ability to estimate unknown functions. Generally, a feedforward network with sufficient number of neurons in hidden layer can approximate any continuous function to any desired accuracy. With this background, a two-layer feedforward network is used to estimate the equivalent control signal of conventional sliding mode controller, as shown in fig.7.2. The output of the neural network is given by (7.9) and can be written as

$$u_{NN} = f_o(\text{net}_i) \quad (7.19)$$

where net_i is the net input to the output layer given by

$$\text{net}_i = \sum_{j=1}^N W_j^h f_j \left(\sum_{i=1}^p W_{ij}^l X_i + b_j \right) + b_0 \quad (7.20)$$

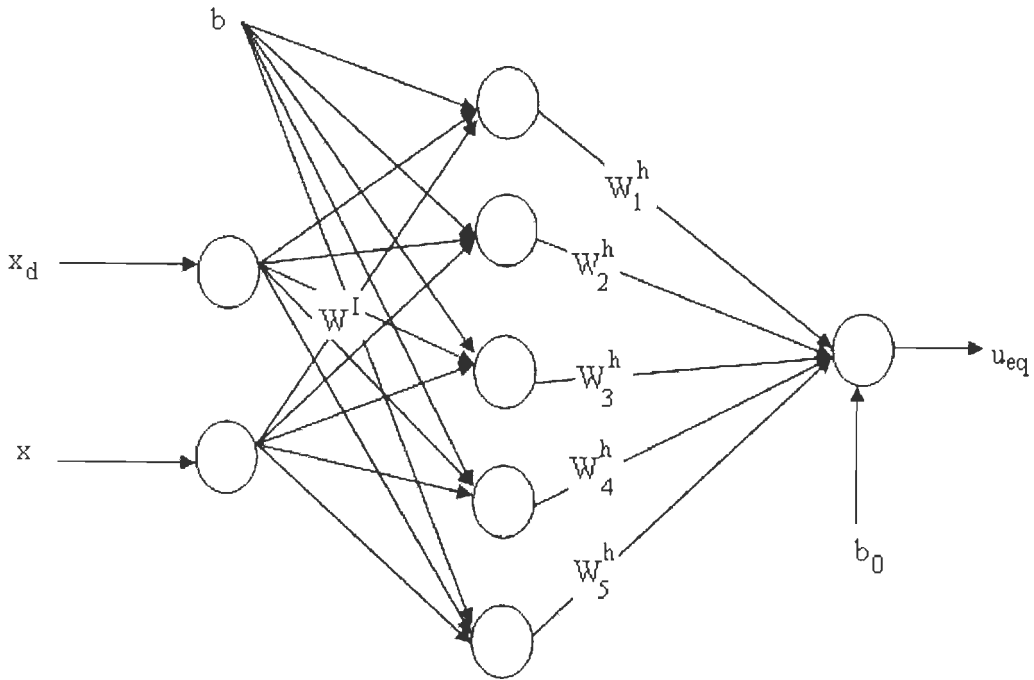


Fig. 7.2 Neural network used for equivalent control estimation

Sliding mode controllers in general have two phases: a reaching phase where the system states are driven towards the sliding surface and a sliding phase where the states are constrained on the sliding surface. The drawback of classical sliding mode controllers is the states come close to the sliding surface and remain within a small region of the sliding surface, which is known as boundary region. The dynamics inside the boundary region is mostly unknown and the high gain of the switching signal may trigger high frequency oscillation of the system, when there are disturbances [65]. Therefore, one of the advantages of neural network is to tune the system in this boundary region, so that the system asymptotically moves towards the sliding surface. In this boundary region the fuzzy system has small gain. Hence we consider two cases, as explained below.

Case 1: reaching phase

In this phase, the distance term s is high and the output of the fuzzy controller is also high, so that it drives the system towards the sliding surface. Therefore, the output of the fuzzy controller is taken to be proportional to the magnitude of this distance variable. The maximum and minimum values of the output of the fuzzy controller are chosen based on the

bounds of uncertainties and disturbance. During this time, the neural network weight updating rate is very small so that output of neural network is smaller than the output of the fuzzy logic to fulfill the stability condition of (7.17). Therefore, the learning rate η is taken as

$$\eta = \frac{1}{\|s\| + 1}, \quad (7.23)$$

to make it always less than 1 and inversely related to the sliding variable s .

Case 2: near the sliding surface

In this region the fuzzy output is constant and at its minimum value. The neural network weights are updated to drive the system further to the sliding surface as a fine tuning that calculate the exact equivalent control signal, which keeps the system on the sliding surface. This means that as far as the neural network approximation error does not become greater than the gain of the fuzzy controller, the system remains stable. To ensure the stability of the system, the initial gain of u_f has to be greater than the sum of bounded uncertainties and disturbance, refer to equation (7.18).

7.5 Modified backpropagation training using sliding parameter

The training algorithm used to update the weights of the neural network is a modification of the standard back-propagation algorithm. The standard back-propagation algorithm is an offline algorithm, which needs a set of input-output pairs. The goal of tuning the neural network is to keep the system on the sliding surface, and hence avoid chattering, i.e. high frequency oscillation of sliding variable s . Therefore, in the weight updating algorithm, we use \dot{s} as the error for training and its square as the performance criteria to be minimized as given in equation (7.22).

$$J = \frac{1}{2} (\dot{s})^2 \quad (7.22)$$

Then, using the steepest decent algorithm, the weights are updated as

$$W_{\text{new}} = W_{\text{old}} - \eta \frac{\partial J}{\partial W}, \quad (7.23)$$

where η is a learning rate determined using the distance function s , refer equation (7.23).

Applying the chain rule and using (7.8), (7.19) and (7.20), for output layer weights, we have

$$\frac{\partial J}{\partial W} = \frac{\partial J}{\partial \dot{s}} \frac{\partial \dot{s}}{\partial W} = \frac{\partial J}{\partial \dot{s}} \frac{\partial \dot{s}}{\partial \text{net}_i} \frac{\partial \text{net}_i}{\partial W} = -(\dot{s}) f'_o(\text{net}_i) y_h, \quad (7.24)$$

where y_h is the output of hidden layer, f'_o is the derivative of output activation function and net_i is the net input to the output layer. Similarly, for the hidden layer,

$$\frac{\partial J}{\partial W} = -(\dot{s}) f'_h(net_h) f'_o(net_i) w^h x_i \quad (7.25)$$

Where x_i is the input to the neural network, f'_h is the derivative of hidden layer neuron activation function, net_h is net input to hidden layer neuron and w^h is the connection weight matrix from hidden layer to output layer.

7.6. Application to SISO system

The neuro-fuzzy sliding mode controller explained above has been implemented using MATLAB for control of a nonlinear second order system having parameter variation and disturbance. The system dynamics is given as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -bx_2 - a \cos(x_1) + u + d \end{aligned} \quad (7.27)$$

where the parameter variations and the disturbances are given by

$$b = 0.5 + 0.2 \sin(t) \quad (7.28)$$

$$a = 3.5 + 0.5 \sin(t) \quad (7.29)$$

$$d = \sin(3t) \quad (7.30)$$

Below we will compare the performance of conventional sliding mode controller, fuzzy sliding mode controller and neuro-fuzzy sliding mode controller. In all cases, the sliding surface used is linear and is given by (6.6). Assuming a 20% decay rate for the error, the sliding surface parameter will be $\lambda=5$. Then for each of the controllers, the rest of the parameters will be designed as follows:

- i) Classical sliding mode controller- in the classical sliding mode controller, the controller used is given by

$$u = K \text{sat}(s/\Psi) \quad (7.31)$$

where $K>0$ is gain, Ψ is width of boundary layer and sat is the saturation function given by

$$\text{sat}(x) = \begin{cases} 1 & x > 1 \\ x & -1 \leq x \leq 1 \\ -1 & \text{if } x < -1 \end{cases} \quad (7.32)$$

The value of the gain K is determined from the parameter variation and disturbance bound. Taking $\Psi=0.05$ and $K=40$, the performance of the classical sliding mode controller is shown in Figs. 7.6, 7.7 and 7.8.

- ii) For the fuzzy sliding mode controller, we have to determine the membership function types and their parameters for the input and output variables and also we have to determine the rule base. Hence, triangular membership functions are used for ease of calculation. Three parameters have to be decided for each membership function. Particularly, the width of the membership function used to represent the linguistic variable ZERO decides the width of the boundary layer and is very important to avoid chattering. Hence, with the above consideration, the membership functions shown in Fig.7.3 and Fig.7.4 are used. The resulting control surface of the fuzzy logic system is graphically shown in Fig.7.5. The above fuzzy logic system is implemented using fuzzy logic toolbox of MATLAB. The performance of the controller is given in Fig.7.6, Fig.7.7 and Fig.7.8.
- iii) Neuro-fuzzy sliding mode controller- As discussed in section 3, the neural network system is used to implement the equivalent controller. A two layer feedforward neural network controller, trained with the online training algorithm explained in section 3, is implemented in MATLAB and is used in parallel with the fuzzy logic system designed in (ii) above. The performance of the neuro-fuzzy controller is shown in Fig.7.6, Fig.7.7 and Fig.7.8.

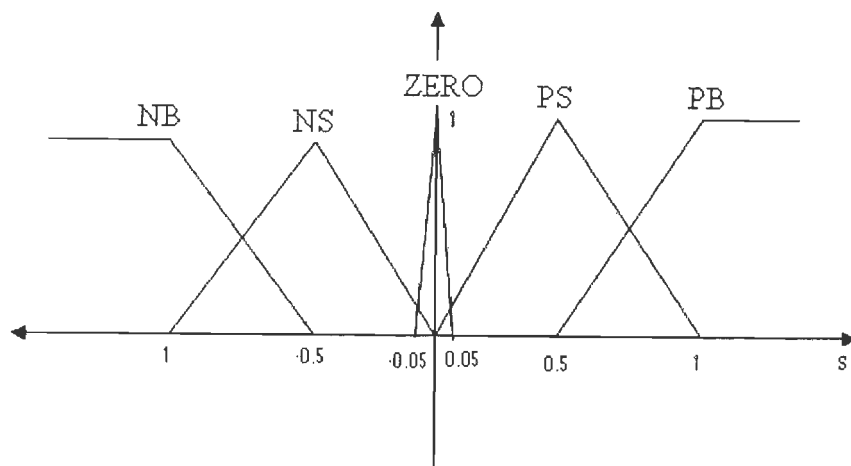


Figure.7.3 Membership functions for input variable

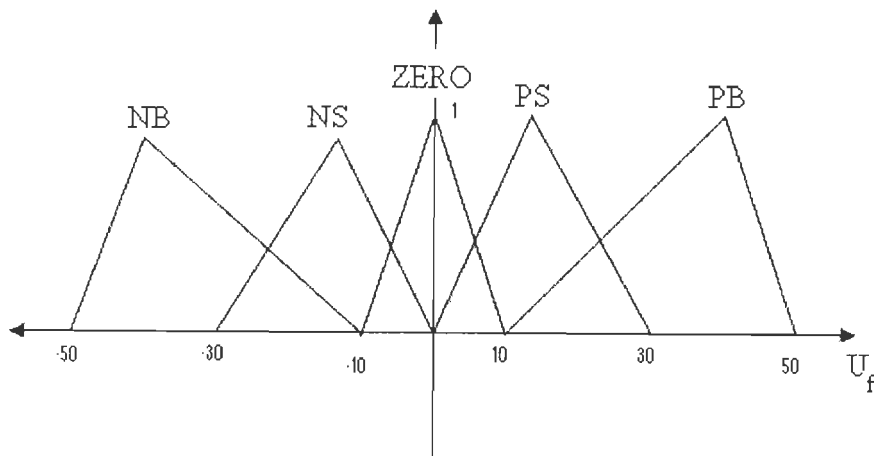


Figure7. 4 Membership functions for the output variable

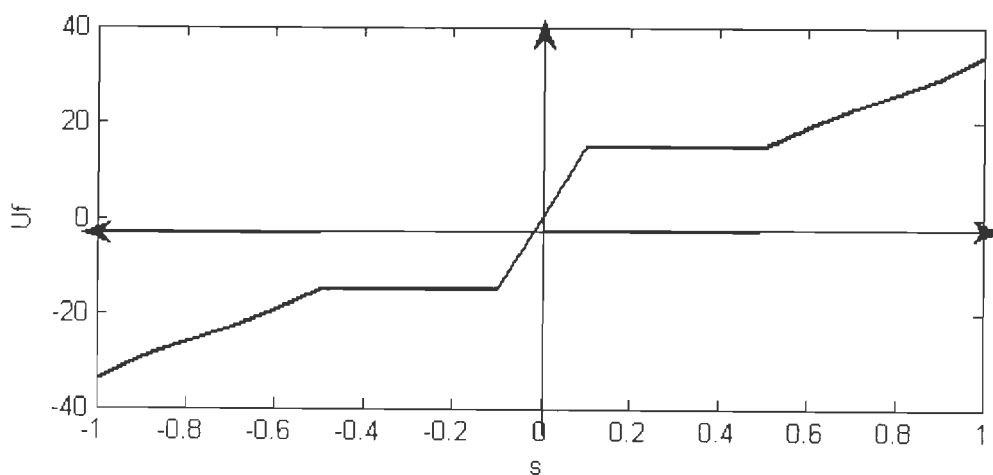


Fig.7.5 Nonlinear control surface of the fuzzy logic system

From Fig.7.6, it is clear that the tracking performance of the controllers goes on improving from classical SMC to FSMC and then to the NFSMC. A more important advantage of the NFSMC is shown in Fig.7.7, which displays the control signals for the three controllers. The figure shows that the FSMC has an appreciable chattering at some points. The same fuzzy logic controller is used with a neural network in the NFSMC and the same figure reveals that the chattering has been suppressed in the NFSMC. Figure7.8 displays the tracking error for the three controllers. From the figure, it is clear that the tracking performance of the NFSMC is better than that of classical SMC and is the same with that of FSMC but with improved control signal. But the cost for the NFSMC is the slight delay in the error decay as it can be seen from Fig.7.8.

7.7. Discussion and Conclusion

In this chapter, we have reported how neural network and fuzzy logic system can be used together to reduce the problem of chattering of classical sliding mode controllers. The design of neural network and fuzzy logic part is given in detail and the stability analysis of the closed loop system is done using Lyapunov's technique. A second order system having parameter variation and disturbance is taken as an example and the controller is implemented using MATLAB and Simulink. Simulation results showed that the system performance is better than classical sliding mode controller and problem of chattering decreased compared to the fuzzy sliding mode controller.

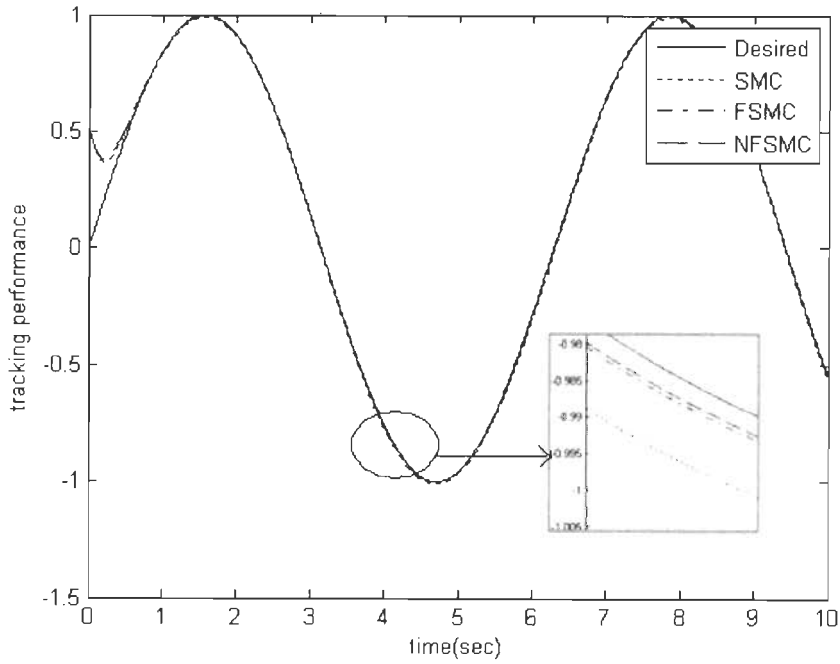


Fig.7.6 Tracking performance of the three controllers (SMC, FSMC and NFSMC) for the position variable x_1 of the inverted pendulum

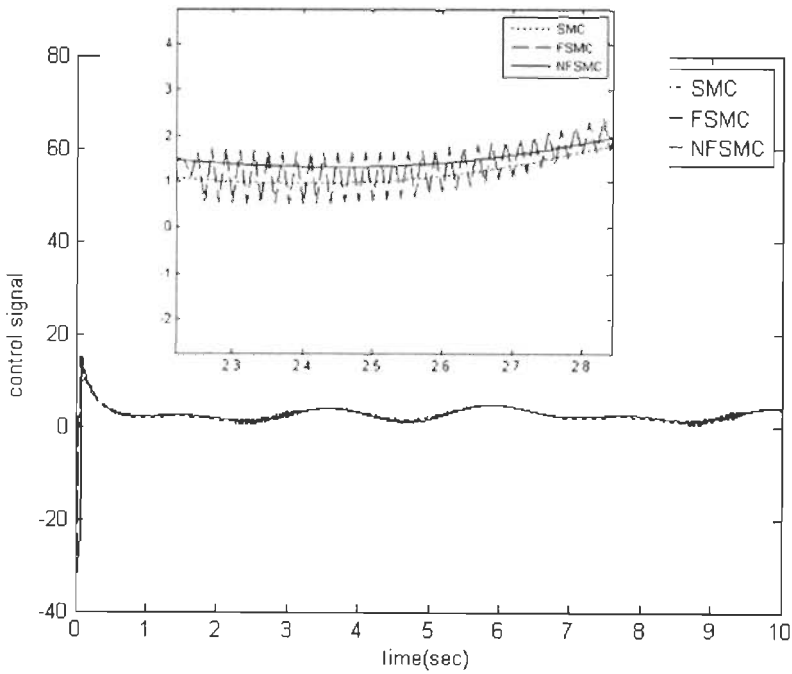


Fig.7.7 The control signals of the three controllers. The Chattering in FSMC is shown in details. The NFSMC has avoided the chattering with improved control performance

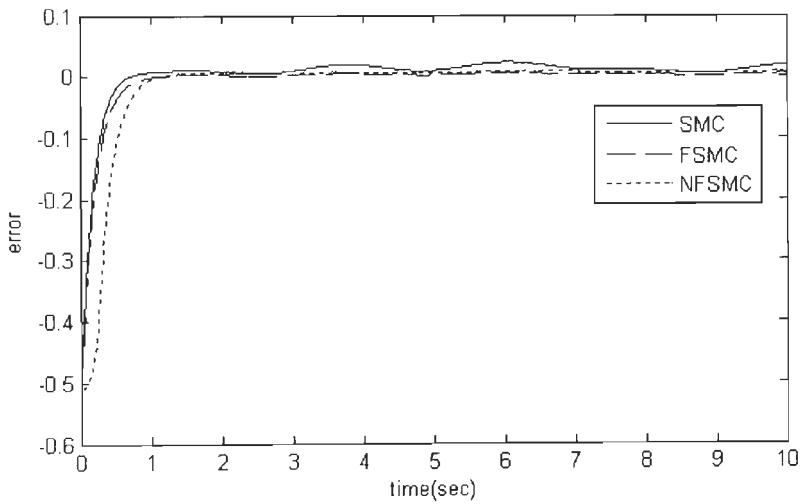


Fig. 7. 8 Tacking error for the three controllers, SMC, FSMC and NFSMC

The structure of the controller may initially seem bulky as both neural network and fuzzy logic are used concurrently and implementation may look uneconomical. However, with the development of high speed microprocessors and software technologies, this problem will vanish. The future work of this research is to analyze the performance of the hybrid system in the presence of unmatched uncertainties. Moreover, we have also a plan to see the system performance for higher order systems.

CONCLUSIONS AND FUTURE WORK

From the research work, we can draw the following conclusions:

1. For Stewart platform used in low speed, low precision applications improving individual leg joint space controllers is effective and fuzzy tuned PID controller works well.
2. For high speed application or when high precision is required, task space control has to be used and sliding mode control in task space gives a better result as seen in fuzzy sliding mode control and integral sliding mode control.
3. Feed forward neural networks result in a better estimation of forward kinematics of Stewart platform and can be used for task space control implementation.
4. We have shown how Genetic algorithm can solve the design problem of ISMC and it is a better and easier design method.
5. To reduce cost of implementation improvements to independent leg PID Joint space controller can be achieved by using hybrid sliding mode controller with our newly proposed SMC.
6. Task space fuzzy sliding mode performs better than joint space SMC or PID but fuzzy sliding mode with an integral loop performs much better.
7. An integral sliding mode controller designed using the GA method we propose has the best performance. It can result in μm precision for a large Stewart platform which can carry a 200Kg load. This means for small sized platforms the precision can be very high.
8. A hybrid implementation which avoids the complex forward kinematics is another alternative which can result in a better performance.
9. Neuro-fuzzy sliding mode controller has the potential to reduce chattering and achieve a better sliding dynamics. Integrating GA designed integral sliding mode controller with neuro-fuzzy design may result in a best controller.
10. We have tried to address all problems of classical SMC by using soft computing.
 - Chattering – Neuro-fuzzy sliding mode controller

Conclusions

- Reaching phase – Integral sliding mode control using GA
- Unmatched uncertainty- using GA
- Smoothness of control signal – Neuro-fuzzy controller

Future work

1. Further investigation regarding the performance of support vector machines and other hybrid algorithms for forward kinematics in order that better reliability and reduced space complexity is achieved a first step for future direction.
2. The study of application of interpretable fuzzy logic system for friction estimation is another good starting point to enhance the performance of the controllers designed.
3. It is clear that all the controllers have finally to be implemented in a digital hard ware where sampling and reconstruction is a must. Hence as a forward step in the direction of practical implementation, study the effect of sampling and delay on tolerance behavior of soft computing techniques and on the sliding mode controllers designed is a third option we are planning for.
4. Rigorous stability analysis of fuzzy logic based sliding mode controllers using Popov's method is one important point that could be added. Stability analysis for robust controllers is mostly performed using Lyapunove based methods but the method needs mathematical representation of fuzzy logic systems. Though mathematical representation of fuzzy logic systems is available, using Popov's method will give a much more intuition.
5. The study of the GA based design method proposed in this thesis has to be further tested with various examples to see its applicability and limitations. A work in this direction is useful both for the soft computing and control system fields and hence is our step in the future.
6. Practical implementation of the algorithms for a Stewart platform manipulator.

Bibliography

- [1] A. A. Goldenberg, "Implementation of force and impedance control in robot manipulators," in *Proc. Robotics and Automation Conf*, Philadelphia, PA, 1988, pp. 1626-1632.
- [2] A. Bagheri A. and J. J. Moghaddam "Decoupled Adaptive Neuro-Fuzzy (DANF) Sliding Mode Control for a Lorenz Chaotic Problem," *Expert Systems with Applications*, vol. 36, pp. 6062-6068, 2009.
- [3] A. Boubakir, F. Boudjema and S. Labiod, "A Neuro-Fuzzy Sliding Mode Controller Using Nonlinear Sliding Surface Applied to the Control of Coupled Tanks System," *International Journal of Automation and Computing*, vol. 6, no.1, pp. 72-80, 2009.
- [4] A. Codourey, "Dynamic modelling and mass matrix evaluation of the DELTA parallel robot for axes decoupling control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, IROS 96*, Osaka, Japan, 1996, vol. 3, pp. 1211-1218.
- [5] A. Ferrara, and M. Rubagotti, "A sub-optimal second order sliding mode controller for systems with saturating actuators," *IEEE Tran. Autom. Control*, vol. 54, no. 5, pp. 1082-1087, May 2009.
- [6] A. Ghobakhloo M. Eghtesad, "Neural network solution for the forward kinematics problem of a redundant hydraulic shoulder," *Proc. 31st Annu. Conference Industrial Electronics Society, IECON*, North Carolina, 2005, pp.1999-2004.
- [7] A. Ghobakhloo, M. Eghtesad and M. Azadi, "Position control of Stewart-Gough platform using inverse dynamics with full dynamics," in *Proc. 9th Int. Workshop on Advanced Motion Control*, IEEE AMC'06, Istanbul, Turkey, 2006, pp.50-55, 2006.
- [8] A. K. Pal and R. K. Mudi, "Self-tuning fuzzy PI controller and its application to HVAC systems," *Int. J. Computational Cognition* ([HTTP://WWW.IJCC.US](http://www.ijcc.us)), vol. 6, no. 1, pp. 25-30, March 2008.
- [9] A. Konaka, D. W. Coitb, A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial" *Reliability Engineering and System Safety*, vol. 91, pp. 992-1107, 2006
- [10] A. Omran, G. El-Bayiumi, M. Bayoumi and A. Kassem, "Genetic algorithm based optimal control for a 6DOF non redundant Stewart platform," *Int. J. Mechanical, Industrial and Aerospace Engineering*, vol. 2, no.2, pp. 73-79, 2008.

- [11] A. Seuret, C. Edwards, S. K. Spurgeon, and E. Fridman, "Static output feedback sliding mode control design via an artificial stabilizing delay," *IEEE Tran. Autom. Control*, vol. 54, no. 2, pp. 256-265, Feb. 2009.
- [12] A. V. Korobeynikov, V. E. Turlapov. (2005, June). "Modeling and evaluating of the Stewart platform," *5th Int. conf. computer Graphics and Vision*. [online]. Available: http://www.graphicon.ru/2005/proceedings/papers/Korobeinikov_turlapov.pdf
- [13] A. Visioli "Tuning of PID controllers using fuzzy logic," *Proc. IEEE-Control Theory Appl.*, vol. 148, no. 1, Jan. 2001.
- [14] B. Bandyopadhyay, F. Deepak and K.-S. Kim, "Sliding mode control using novel sliding surfaces," in *Lecture Notes in Control and Information Science 392*, Berlin: Springer-Verlag, 2009.
- [15] B. Dasgupta and T. S. Mruthyunthaya, "The Stewart Platform Manipulator: a review," *Mechanism and Machine Theory*, vol. 35, no. 1 pp. 15-40, 2000.
- [16] C. C. Nguyen, Z.-L. Zhou and Sami S. Antraz, "Efficient computation of forward kinematics and Jacobian matrix of a Stewart platform based manipulator," in *Proc. IEEE Int. Conf. on Robotics and Autom.*, vol. 1, 1997, pp. 869-874.
- [17] C. M. Gosselin and J. Angeles, "Singularity Analysis of Closed-Loop Kinematic Chains", *IEEE Trans. Robotics and Autom.*, vol. 6, no. 3. pp. 281-290, JUNE 1990
- [18] C. M. Gosselin, "Parallel computational algorithms for the kinematics and dynamics of parallel manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, 1993, vol.1, pp. 883-888.
- [19] C. N. Riviere, J. Gangloff, and M. Mathelin, "Robotic Compensation of Biological Motion to Enhance Surgical Accuracy," in *Proc. IEEE*, vol. 94, no. 9, Sept. 2006.
- [20] C. Pernechele, F. Bortoletto and Enrico Giro, "Neural network algorithm controlling a hexapod platform", in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Networks, IJCNN*, 2000, Vol.4, pp. 349-352.

- [21] C. Reboulet and T. Berthomjeu, "Dynamic models of six degree of freedom parallel manipulators," in Proc. 5th Fifth Int. Con. Advanced Robotics, 'Robots in Unstructured Environments', 91 ICAR., IEEE, Toulouse, France, 1991, pp. 1153-1157.
- [22] C.-C. Hua, Q.-G. Wang, and X.-Ping Guan, "Memoryless state feedback controller design for time delay systems with matched uncertain nonlinearities," *IEEE Tran. Automatic Control*, vol. 53, no. 3, April 2008, pp. 801- 807.
- [23] C.-I. Huang, and L.-C. Fu, "smooth sliding mode tracking control of the Stewart platform," in *Proc. IEEE Conf. on Control Applications*, Toronto, Canada, 2005, PP. 43-48.
- [24] C.-L. Chen and M.-H. Chang, "Optimal design of fuzzy sliding mode control: a comparative study," *Fuzzy Sets and Systems*, vol. 93, pp. 37-48, 1998.
- [25] D. Daney, I. Z. Emiris, Y. Papegay, E. Tsigaridas and J.-P. Merlet, "Calibration of parallel robots: on the elimination of pose-dependent parameters," in *Proc. EUCOMES, the 1st European Conf. on Mechanism Science*, Obergurgl, Austria, Feb. 2006, pp 1-12.
- [26] D. Driankove, H. Hellendroon and M. Reinfrank, *An Introduction to Fuzzy Control*, 2nd Edition, Berlin:Springer Verlag, Hedilberg 1996.
- [27] D. E. Barkana, "Orthopedic Surgery Robotic System," in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, Guilin, China, 2009, pp.1947-1952
- [28] D. K. Pratihari, *Soft computing*, Alpha Science International, 2007.
- [29] D. Li and S. E. Salcudean, "Modeling, simulation, and control of a hydraulic Stewart platform," in *Proc. IEEE Int. Conf. on Robotics and Autom.*, Albuquerque, New Mexico, April 1997, pp.3360-3366.
- [30] D. S. Negash and R. Mitra, "Integral sliding mode control for trajectory tracking control of Stewart platform manipulator," in Proc. Int. Conf. Industrial and Information Systems, IEEE- ICIIS 2010, Surathkal, India, July 2010, pp. 650-654.
- [31] D. Sha, V. B. Bajic and H. Yang, "New model and sliding mode control of hydraulic elevator velocity tracking system," *Simulation Practice and Theory* vol.9, pp. 365-385, 2002

- [32] D. Shiferaw and R. Mitra, "Comparison of numerical and neural network forward kinematics estimation for Stewart platform," in *Proc. Int. Conf. on Computer Applications in Electrical Engineering and Recent Advances*, Roorkee, India, Feb. 2010, pp. 272-257.
- [33] D. Shiferaw and R. Mitra, "Fuzzy logic tuned PID controller for Stewart platform manipulator," *Proc. Int. Conf. on Computer Applications in Electrical Engineering and Recent Advances*, Roorkee, India, pp. 272-257, Feb. 19-21, 2010.
- [34] D. Shiferaw and R. Mitra, "Neuro-Fuzzy Sliding mode control: design and stability analysis," *Int. J. of Computational Intelligence Studies*, vol. 1, no. 3, pp. 242-255, 2010.
- [35] D. Xiao, B. K. Ghosh, N. Xi and T. J. Tarn, "Sensor-Based hybrid position/force control of a robot manipulator in an uncalibrated environment," *IEEE Tran. on Control Systems Technology*, vol. 8, no. 4, pp. 635- 645, July 2000.
- [36] D. Zhao, S. Li and F. Gao, "Fully adaptive feedforward feedback synchronized tracking control for Gough-Stewart platform systems," *Int. Journal of Control Autom. and Systems*, vol. 6, no. 5, pp. 689-701, Oct. 2008.
- [37] D. Zhu and Y. Fang, "Adaptive control of parallel manipulators via fuzzy-neural network algorithm," *Journal of Control Theory and Applications* vol. 5, no. 3, pp. 295-300, 2007.
- [38] E F Fichter, D. R. Kerr, and J Rees-Jones, "The Gough-Stewart platform parallel manipulator: a retrospective appreciation," in *Proc. Institutions of Mechanical Engineers, Part C: J. Mechanical Engineering Science*, Vol. 223, no.1, 2009, DOI 10.1243/09544062JMES1137.
- [39] E. D. Fasse, and C. M. Gosselin, "Spatio-geometric impedance control of Gough-Stewart platforms," *IEEE Tran. Robotics and Autom.*, vol. 15, no. 2, pp. 281-285, April 1999.
- [40] E. Vieira, L. Nunes, L. Hsu and F. Lizarralde, "Global exact tracking for uncertain systems using output-feedback sliding mode control," *IEEE Tran. on Autom. Control*, vol. 54, no. 5, pp. 1141-1147, May 2009.
- [41] F. C. Teng, A. Lotfi and A. C. Tsoi, "Novel fuzzy logic controllers with self-tuning capability," *Journal of Computers*, vol. 3, no. 11, pp. 9-16, Nov. 2008.

- [42] F. Castaños and L. Fridman, "Analysis and design of integral sliding manifolds for systems with unmatched perturbations," *IEEE Tran. on Automatic Control*, vol. 51, no. 5, pp. 853-858, May 2006.
- [43] F. H. Ghorbel, O. Chételat, R. Gunawardana and R. Longchamp, "Modeling and set point control of closed chain mechanisms, theory and experiment," *IEEE Tran. on Control Systems Technology*, vol. 8, no. 5, pp. 801-815, Sept. 2000.
- [44] F. Paccot, N. Andreff, P. Martinet and W. Khali, "Vision-based computed torque control for parallel robots," in *Proc. IEEE Int. Conf. Robotics and Autom.*, 2006, pp. 3851-3856.
- [45] F. Qiao, Q. Zhu, W. FT Alan and C. Melhuish, "Adaptive sliding mode control for MIMO nonlinear systems based on a fuzzy logic scheme," *Int. Journal of Autom. and Computing*, vol. 1, pp. 51-62, 2004.
- [46] F.-Y Hsu and L.-C. Fu, "Intelligent robot deburring using adaptive fuzzy hybrid position/force control," *IEEE Tran. on Robotics and Autom.*, vol. 16, no. 4, pp 325-335, August 2000
- [47] G. Bartolini, A. Ferrara and E. Usai, "Chattering avoidance by second-order sliding mode control," *IEEE Tran. Autom. Control*, vol. 43, no. 2, pp. 241-246, Feb. 1998.
- [48] G. Bartolini, A. Pisano and E. Usai, "An improved second-order sliding-mode control scheme robust against the measurement noise," *IEEE Tran. Autom. Control*, vol. 49, no. 10, pp. 1731-1736, Oct. 2004.
- [49] G. He and Z. Lu, "The Research on the Redundant Actuated Parallel Robot with Full Compliant," in *Proc. IEEE Conf. Robotics, Automation and Mechatronics*, Bangkok, Thailand, 2006, pp.1-6.
- [50] G. Yang, I.-M. Chen, W. Lin and J. Angeles, "Singularity analysis of three- legged parallel robots based on passive-joint velocities," *IEEE Tran. Robotics and Autom.*, vol. 17, no. 4, pp 413-422, August 2001
- [51] H. Abdellatif and B. Heimann, "Advanced Model-Based Control of a 6-DOF Hexapod Robot: A Case Study", *IEEE/ASME Trans. Mechatronics*, vol., 15, no. 2, pp. 269-279, April 2010.
- [52] H. Abdellatif and B. Heimann, "Applying efficient computation of the mass matrix for decoupling control of complex parallel manipulators," in *IFAC 16th Triennial World Congress*, Prague, Czech Republic, 2005,pp. 493-498.

- [53] H. Bruyninckx. (2005, August 20). *Parallel Robots*. [Online]. <http://www.roble.info>.
- [54] H. F. Ho, Y. K. Wong and A.B. Rad, "Adaptive fuzzy sliding mode control with chattering elimination for nonlinear SISO systems," *Simulation Modeling Practice and Theory* vol. 17, pp.1199–1210, 2009.
- [55] H. H. Choi, "LMI-Based sliding surface design for integral sliding mode control of mismatched uncertain systems," *IEEE Tran. Autom. Control*, vol. 52, no. 4, April 2007.
- [56] H. Lin and J. E. McInro, "Adaptive sinusoidal disturbance cancellation for precise pointing of Stewart platforms" *IEEE Tran. Control Systems Technology*, vol. 11, no. 2, pp. 267-272, March 2003.
- [57] H. Q. Thinh Ngo, J.-H. Shin and W.-H. Kim, "Fuzzy sliding mode control for a robot manipulator," *J. Artificial Life and Robotics* vol. 13, pp.124–128, 2008
- [58] H. S. Kim, Y. Shim, Y. M. Cho and K. -I. Lee, "Robust nonlinear control of a 6 dof parallel manipulator: task space approach," *KSME Int. Journal* vol. 16, no. 8, pp. 1053-1063, 2002.
- [59] H. S. Kima, Y. M. Chob and K. -I. Lee, "Robust nonlinear task space control for 6 DOF parallel manipulator," *Automatica* vol. 41, pp. 1591-1600, 2005.
- [60] H. Sadjadian , H.D. Taghirad and A. Fatehi, "Neural networks approaches for computing the forward kinematics of a redundant parallel manipulator," *Int. J. of Computational Intelligence*, vol. 2, no.1, pp 40-47,2005
- [61] H. Wang, C. Xue and W. A. Gruver, "Neural network control of a parallel robot," in *Proc. IEEE Int. Con. Systems, Man and Cybernetics, Intelligent Systems for the 21st Century*, Vancouver, BC, Canada, 1995, pp. 2934-2938.
- [62] H. B. Guo and H.R. Li, "Dynamic Analysis and Simulation of a Six Degree of Freedom Stewart Platform Manipulator ", in *Proc. Institution of Mechanical Engineers, Part C: J. of Mechanical Engineering Science*, vol. 220, no. 1, pp. 61-72, 2006, DOI 10.1243/095440605X32075
- [63] H.-P. Hong, S.-J. Park, S.-J. Han, K.-Y. Cho, Y.-C. Lim, J.-K. Park et al., "A design of auto-tuning PID controller using fuzzy logic", in *Proc. Int. Con. Industrial Electronics, Control, Instrumentation, And Automation, Power Electronics And Motion Control*, San Diego, CA, USA, 1992, pp. 971-976.

- [64] I. A. Bonev, J. Ryut, N.-J. Kim and S.-K. Lees, "A simple new closed-form solution of the direct kinematics of parallel manipulators using three linear extra sensors," in *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Atlanta, USA, 1999, pp. 526-530.
- [65] I. Boiko, L. Fridman, A. Pisano and E. Usai, "Analysis of chattering in systems with second-order sliding modes," *IEEE Tran. Autom. Control*, vol. 52, no. 11, pp. 2085-2102, Nov. 2007.
- [66] I. Boiko, L. Fridman, A. Pisano and E. Usai, "On the transfer properties of the "Generalized Sub-Optimal" second-order sliding mode control algorithm" *IEEE Tran. Autom. Control*, vol. 54, no. 2, pp. 390-403, Feb. 2009.
- [67] I. Davliakos and E. Papadopoulos, "Model-based control of a 6-dof electrohydraulic Stewart-Gough platform," *J. Mechanism and Machine Theory*, vol. 43, no. 11, pp. 1385-1400, Nov. 2008.
- [68] I. Davliakos and E. Papadopoulos, "Model-based position tracking control for a 6-dof electrohydraulic Stewart platform," in *Proc. IEEE Mediterranean Conf. Control and Automation, Athens-Greece*, 2007, pp. 1-6.
- [69] I. Eker and S. A. Akınal, "Sliding mode control with integral augmented sliding surface: design and experimental application to an electromechanical system," in *Electrical Engineering Letters*, vol. 90, pp. 189-197
- [70] I. J. Nagarath and M. Gopal, *Control Systems Engineering*, 5th Edition, New Delhi: New Age Int. Publishers, 2008.
- [71] I.-F. Chung, H.-H. Chang and C.-T. Lin, "Fuzzy control of a six-degree motion platform with stability analysis," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, 1999, vol.1, pp. 325-330.
- [72] J. E. McInroy, J. F. O'Brien, and G. W. Neat, "Precise, Fault-Tolerant Pointing Using a Stewart Platform," *IEEE/ASME Trans. on Mechatronics*, vol. 4, no.1, pp. 91-95, March 1999
- [73] J. P. Merlet http://www-sop.inria.fr/members/Jean-Pierre.Merlet//merlet_eng.html
- [74] J. Wang, A.B. Rad and P.T. Chan, "Indirect adaptive fuzzy slidingmode control: Part I: fuzzy switching," *Fuzzy Sets and Systems* vol.122, pp. 21-30, 2001.

- [75] J. X. Luo and H. H. Shao, "Developing soft sensors using hybrid soft computing methodology: a neurofuzzy system based on rough set theory and genetic algorithms", *J. Soft Computing*, vol. 10, pp. 54-60, 2006.
- [76] J. Y. Hung, W. Gao and J. C. Hung, "Variable Structure Control: A Survey," *IEEE Tran. Industrial Electronics*, vol.40, no.1, pp. 2-19, Feb.1993.
- [77] J.-C. Lo and Y.-H. Kuo, "Decoupled fuzzy sliding mode control," *IEEE Tran. Fuzzy Systems*, vol. 6, no. 3, pp. 426-435, August 1998.
- [78] J.-D. Wang, T.-L. Lee and Y.-T. Juang, "New methods to design an integral variable structure controller," *IEEE Tran. Automatic Control*, vol. 41, no. 1, pp. 140-143, Jan. 1996.
- [79] J.-P. Merlet, "Closed-form resolution of the direct kinematics of parallel manipulators using extra sensors data," in *Proc. IEEE Int. Conf. Robotics and Autom.*, 1993, vol. 1, pp. 200-204.
- [80] J.P. Merlet, *Parallel Robots*, 2nd Ed., Springer 2006
- [81] J.-X. Xu, Y.-J. Pan and T.-H. Lee, "Analysis and design of integral sliding mode control based on lyapunov's direct method," in *Proc. American Control Conf.*, Denver, Colorado, 2003, pp. 192-196.
- [82] J.-Y. Kang, D. H. Kim and K.-I. Lee, "Robust estimator design for forward kinematics solution of a Stewart platform," *J. Robotic Systems* vol. 15, no. 1, pp. 29-42, 1998.
- [83] K. Kiguchi and T. Fukuda, "Position/force control of industrial robot manipulators for geometrically unknown objects using fuzzy neural networks," *IEEE Tran. Industrial Electronics*, vol. 47, no. 3, pp. 641-649, June 2000.
- [84] K. Kiguchi, and T. Fukuda, "Intelligent position/force controller for industrial robot manipulators—application of fuzzy neural networks," *IEEE Tran. Industrial Electronics*, vol. 44, no. 6, pp. 753-761, Dec. 1997.
- [85] K. Liu, J. M. Fitzgerald and F. L. Lewis, "Kinematic analysis of Stewart platform manipulator," *IEEE Tran. Industrial Electronics*, vol. 40, no. 2, pp. 282-293, April 1993.
- [86] K. Rajeswari and P. Lakshmi, "GA tuned distance based fuzzy sliding mode controller for vehicle suspension systems," *Int. J. Engineering and Technology*, vol. 5, no. 1, pp. 36-47, 2008.

- [87] K.-M. Lee and D. K. Shah, "Dynamic analysis of a three-degrees-of- freedom in-parallel actuated manipulator," *IEEE J. Robotics and Autom.*, vol. 4, no. 3, June 1988.
- [88] K.-M. Lee and D. K. Shah, "Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator," *IEEE J. Robotics and Automation*, vol. 4, no. 3, pp. 354-360, June 1988.
- [89] L. Beji, A. Abichou and M. Pascal, "Tracking control of a parallel robot in the task space," in *Proc. IEEE Int. Conf. Robotics and Autom.*, Leuven, Belgium, 1998, pp. 2309-2304.
- [90] L. H. Keel, S. P. Bhattacharyya and J. W. Howze, "Robust control with structured perturbations," *IEEE Tran. Automatic Control*, vol. 33, no. 1, pp. 68-78, January 1988.
- [91] L. H. San and M.-C. Han, "The estimation for forward kinematics of Stewart platform using the neural network," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 1999, pp. 501-506.
- [92] L.-C. Lin and J.-C. Lai, "Stable adaptive fuzzy control with TSK fuzzy friction estimation for linear drive systems", *J. Intelligent and Robotic Systems* vol. 38, pp. 237-253, 2003.
- [93] L.-C. Tommy Wang and C. C. Chen, "On the numerical kinematic analysis of general parallel robotic manipulators," *IEEE Tran. Robotics and Automation*, vol. 9, no. 3, pp 272-285, June 1993.
- [94] L.E. Bruzzone, R.M. Molfino, M. Zoppi and G. Zurlo, "The PRIDE prototype: control layout of a parallel robot for assembly tasks," in *Proc. 22nd IASTED Int. Conf. modeling identification control*, Innsbruck, Austria, pp. 606-611, 2003
- [95] M. Basin, J. Rodriguez-Gonzalez, L. Fridman and P. Acosta, "Integral sliding mode design for robust filtering and control of linear stochastic time-delay systems," *Int. J. Robust and Nonlinear Control*, vol. 15, pp. 407-421, April 2005
- [96] M. C. Turner and D. G. Bate, "Mathematical methods for robust and nonlinear control," in *Lecture Notes in Control and Information Science 367*, Berlin :Springer-Verlag Heidelberg, 2007.
- [97] M. Dotoli B. Maione, D. Naso and B. Turchiano, "Fuzzy sliding mode control for inverted pendulum swing-up with restricted travel," in *Proc. IEEE Int. Conf. Fuzzy Systems*, Bari, Italy, 2001, pp. 753-756.

- [98] M. Dotoli, B. Maione, D. Naso and B. Turchiano, "Fuzzy sliding mode control for inverted pendulum swing-up with restricted travel", 2001 IEEE Int. Symposium on Fuzzy Systems, pp. 753-756, 2-5 Dec. 2001
- [99] M. Durali and E. Shameli, "Full order neural velocity and acceleration observer for a general 6-6 Stewart platform," in *Proc. IEEE Int. Conf. on Networking Sensing and Control*, Taipei-Taiwan, 2004, pp. 333-338.
- [100] M. Girone, G. Burdea, M. Bouzit, V. Popescu and J.E. Deutsch, "A Stewart platform-based system for ankle telerehabilitation" *J. Autonomous Robots* vol. 10, pp. 203-212, 2001.
- [101] M. R. Sirouspour, and S. E. Salcudean, "Nonlinear Control of Hydraulic Robots," *IEEE Tran. Robotics and Automation*, vol. 17, no. 2, pp. 173-192, April 2001.
- [102] M. Roopaei and M. Z. Jahromi, "Chattering-free fuzzy sliding mode control in MIMO uncertain systems," *Journal of Nonlinear Analysis* vol. 71, pp. 4430-4437, 2009.
- [103] M. Tarokh, "Real time forward kinematics solution for general Stewart platform," in *Proc. IEEE Int. Conf. Robotics and Autom.*, Roma-Italy, 2007, pp. 901-906.
- [104] M. Wapler, V. Urban, T. Weisener, J. Stallkamp, M. Durr and A. Hiller, "A Gough-Stewart platform for precision surgery", *Tran. Institute of Measurement and Control*, vol. 25, no. 4, pp. 329-334, 2003.
- [105] N. Andreff and P. Martinet, "Unifying kinematic modeling, identification, and control of a Gough-Stewart parallel robot into a vision-based framework," *IEEE Tran. Robotics*, vol. 22, no. 6, pp.1077-1086, Dec. 2006.
- [106] N. H. Jo, J. Jin, S. Joo and J. H. Seo, "Generalized Luenberger-like observer for nonlinear systems," in *Proc. American Control Conference*, Albuquerque, New Mexico, 1997, pp. 2180-2183.
- [107] N. Hogan, "On the stability of manipulators performing contact tasks," in *Proc. IEEE Int. Con. Robotics and Autom.*, Sacramento, California, 1991, pp. 677-686
- [108] N. Leroy, A. M. Kokosy and W. Perruquetti, "Dynamic modeling of a parallel robot, Application to a surgical simulator," in *Proc. IEEE Int. Conf. Robotics & Autom.*, Taipei, Taiwan, 2003, pp. 4330-4335.

- [109] N. Ramdani and P. Poignet, "experimental parallel robot dynamic model evaluation with set membership estimation," in *14th IFAC Symposium on System Identification*, Newcastle, Australia, 2006, pp. 967-972.
- [110] N. Sima'an and M. Shoham, "Singularity analysis of a class of composite serial in-parallel robots," *IEEE Tran. Robotics and Automation*, vol. 17, no. 3, pp. 301-311, June 2001.
- [111] N. Sima'an, D. Gloman and M. Shoham, "Design considerations of new six degrees-of-freedom parallel robots," in *Proc. IEEE Int. Conf. Robotics & Automation*, Leuven, Belgium, 1998, pp. 1327-1333, May 1998
- [112] N. Yagiz, I. Yuksek and T. Kepceler "Sliding mode control of a planar flexible single-arm robot," *J. Polytechnic*, vol. 4, no. 2, pp. 15-19, 2001.
- [113] N.-I. Kim and C.-W. Lee, "High speed tracking control of Gough-Stewart platform manipulator via enhanced sliding model controller," in *Proc. Int. Conf. Robotics and Autom.*, pp. 2716-2721, Leuven Belgium, 1998.
- [114] O. Dirdrit, M. Petitot and E. Walter, "Guaranteed solution of direct kinematic problem for general configuration of parallel manipulators," *IEEE Tran. Robotics and Autom.*, vol. 14, no. 2, pp.259-266, April 1998.
- [115] O. Khatib and J. Burdick, "Motion and force control of robot manipulators," in *Proc. IEEE Int. Conf Robotics and Autom.*, 1986, pp 1381-1386.
- [116] O. M. Almeida, L. L. N. Reis, L. D. S. Bezerra and S. E. U. Lima, "A MIMO fuzzy logic autotuning PID controller: Method and application" in *Applied Soft Computing Technologies: The Challenge Of Complexities* (Advances In Soft Computing Vol. 34), pp. 569-580, 2006, DOI: 10.1007/3-540-31662-0_44.
- [117] P. Nanua, K. J. Waldron and V. Murthy, "Direct kinematic solution of a Stewart platform," *IEEE Trans. Robotics and Autom.*, vol. 6, no. 4, pp. 438-444, August 1990
- [118] P. C. Chen, C. W. Chen and W.L. Chiang,, "GA-based modified adaptive fuzzy sliding mode controller for nonlinear systems," *Expert Systems with Applications*, vol. 36, pp. 5872-5879, 2009.

- [119] P. Chiacchio, F. Pierrot, L. Sciavicco and B. Siciliano, "Robust design of independent joint controllers with experimentation on a high speed parallel robot," *IEEE Trans. Industrial Electronics*, vol. 40, no. 4, pp. 393-403, 1993.
- [120] P. Ji and H. Wu, "A closed-form forward kinematics solution for the 6-6 Stewart platform", *IEEE Trans. Robotics and Autom.*, vol. 17 no. 4, pp. 522-526, August 2001.
- [121] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure", *IEEE Tran. Automatic Control*, vol. 53, no. 9, pp. 2180-2185, Oct. 2008.
- [122] Q. Xu, J. Yu, W. Gu and D. Tang, "Fuzzy sliding-mode controller design based on W-stability theorem," *Int. J. Information Technology* vol. 11, no. 6, pp. 61-67, 2005.
- [123] Q. Zhou, P. Kallio and H. N. Koivo, "Modelling of a Piezohydraulic Actuator for Control of a Parallel Micromanipulator," in *Proc. IEEE Int. Conf. Robotics & Automation*, Detroit, Michigan, 1999, pp. 2750-2755
- [124] R. A. Freeman and P. V. Kokotovic, *Robust nonlinear control design: state space and Lyapunov techniques*, Birkhauser, 2008.
- [125] R. D. Klafter, T. A. Chmielewski and M. Negin, *Robotic Engineering an Integrated Approach*, New Delhi: Prentice Hall of India, 2001.
- [126] R. Featherstone and D. Orin, "Robot dynamics: equations and algorithms," in *Proc. IEEE Int. Conf. Robotics and Autom.*, San Francisco, CA, pp. 826-834, April 2000.
- [127] R. J. Anderson and M. W. Spong, "Hybrid impedance control of robotic manipulators," *IEEE J. Robotics and Autom.*, vol. 4, no. 5, pp. 549-556, Oct. 1988.
- [128] R. J. Schilling, *Fundamentals of Robotics: Analysis and Control*, New Delhi: Prentice Hall of India Private Limited, 2006.
- [129] R. Isermann, "Some Thoughts on Automatic Control Engineering", in *Proc. IMechE Part I: J. Systems and Control Engineering*, vol. 223, pp.131-133, 2009.
- [130] S. Alavandar, M. J. Nigam, "Neuro-Fuzzy based approach for inverse kinematics solution of industrial robot manipulators," *Int. J. of Computers, Communications and Control*, vol. 3, no. 3, pp. 224-234, 2008.

- [131] S. Fu and Y. Yao, "Non-linear robust control with partial inverse dynamic compensation for a Stewart platform manipulator," *Int. J. Modelling, Identification and Control*, vol. 1, no. 1, pp. 44-51, 2006.
- [132] S. Iqbal and A. I. Bhatti, "Robust sliding-mode controller design for a Stewart platform," in *Proc. Int. Bhurban Conf. Applied Sciences & Technology*, Islamabad, Pakistan, pp. 155-160, Jan. 2007.
- [133] S. Iqbal, A. I. Bhatti and Q. Ahmed, "Dynamic analysis and robust control design for stewart platform with moving payloads," in *Proc. 17th World Congress Int. Federation of Automatic Control*, Seoul, Korea, pp.5324-5329, Jul. 2008
- [134] S. Janardhanan and B. Bandyopadhyay, "On discretization of continuous-time terminal sliding mode," *IEEE Tran. Automatic Control*, vol. 51, no. 9, pp. 1532- 1536, Sept. 2006.
- [135] S. Jung and T. C. Hsia, "Neural network impedance force control of robot manipulator," *IEEE Tran. Industrial Electronics*, vol. 45, no. 3, pp. 451-461, June 1998
- [136] S. Jung and T. C. Hsia, "Robust neural force control scheme under uncertainties in robot dynamics and unknown environment," *IEEE Tran. Industrial Electronics*, vol. 47, no. 2, pp. 403- 412, April 2000.
- [137] S. Kock, and W. Schumacher, "Control of a fast parallel robot with a redundant chain and gearboxes: experimental results," in *Proc. IEEE Int. Conf. Robotics & Autom.*, San Francisco, CA, pp. 1924-1929, April 2000.
- [138] S. Mitra, and Y. Hayashi, "Neuro-Fuzzy rule generation: survey in soft computing framework" *IEEE Tran. Neural Networks*, vol. 11, no. 3, pp. 748-768, May 2000.
- [139] S. N. Huang, K. K. Tan and T. H. Lee, "Adaptive friction compensation using neural network approximations," *IEEE Tran. Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 30, no. 4, pp. 551-557, Nov. 2000.
- [140] S. P. Batacharaya, H. Chappellat, L. H. Keel, *Robust control: the parameteric approach*, Prentice Hall, 1995
- [141] S. Q. Xie, Y. L. Tu, A. Shaw and Z. C. Duan, "A fuzzy integral sliding mode control algorithm for high-speed laser beam focus tracking control," *Int. J. Advanced Manufacturing Technology*, vol. 20, pp. 296-302, 2002.

- [142] S.-R. Hebertt, "Nonlinear variable structure systems in sliding mode: the general case" *IEEE Tran. Autom. Control*, vol. 34, no. 11, pp. 1186-1188, Nov. 1989.
- [143] S. R. Ploen and F. C. Park, "Coordinate-invariant algorithms for robot dynamics," *IEEE Tran. Robotics and Autom.*, vol. 15, no. 6, pp. 1130-1135, Dec. 1999.
- [144] S. W. Wang and D. L. Yu, "Neural network based integral sliding mode control for nonlinear uncertain systems" in *Advances in industrial engineering and operation research (Lecture Notes in Electrical Engineering 5)*, Alan H. S. Chan and Sio-long Ao, Eds, Berlin: Springer 2008, pp. 245-258, DOI: 10.1007/978-0-387-74905-1.
- [145] S.-H. Lee, J.-B. Song, W.-C. Choi and D. Hong, "Position control of a Stewart platform using inverse dynamics control with approximate dynamics," *Mechatronics*, vol. 13, pp. 605-619, 2003.
- [146] S.-J. Huang and K.-C. Chiou, "An adaptive neural sliding mode controller for mimo systems" *Int. J. Robot Systems*, vol. 46, pp. 285-301, 2006.
- [147] S.-K. Song and D.-S. Kwon, "Efficient formulation approach for the forward kinematics of the 3-6 Stewart-Gough platform," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Mani, Hawaii, USA, pp. 1688-1693, 2001.
- [148] T. C. Kuo and Y. C. Huang. (2008, Feb.). Global stabilization of robot control with neural network and sliding mode. *IAENG Engineering Letters*. [Online]. 16(1).
http://www.engineeringletters.com/issues_v16/issue
- [149] T. Chatchanayuenyong and M. Pranichkun, "Neural network based time optimal sliding mode control for an autonomous underwater robot," *Mechatronics*, vol. 16, pp. 471-478, 2006.
- [150] T. L. Chern and Y. C. Wu, "Design of integral variable structure controller and application to electro hydraulic velocity servo systems," in *Proc. IEEE Control Theory Appl.*, vol. 41, pp. 439-444, 1991.
- [151] T. L. Chern and Y. C. Wu, "Integral variable structure control approach for robot manipulators," in *Proc. IEEE Control Theory Appl.*, vol. 139, pp. 161-166, 1992.
- [152] T. L. Seng, M. Khalid and R. Yusof, "Tuning of a neuro-fuzzy controller by genetic algorithm," *IEEE Tran. Systems, Man and Cybernetics*, vol. 29, no.2, April 1999.

- [153] T.-Z. Wu, J.-D. Wang and Y.-T. Juang, "Decoupled integral variable structure control for MIMO systems," *J. Franklin Institute*, vol. 344, pp. 1006–1020, 2007.
- [154] U. Seibold, B. Kuebler, and G. Hirzinger. (2008, January). "Prototypic force feedback instrument for minimally invasive robotic surgery," Available [Online] http://www.intechopen.com/articles/show/title/prototypic_force_feedback_instrument_for_minimally_invasive_robotic_surgery
- [155] V. I. Utkin "Sliding mode control design principles and applications to electric drives" *IEEE Tran. Industrial Electronics*, vol. 40, no. 1, pp. 23-36, Feb. 1993
- [156] V. I. Utkin and J. Shi, "Integral sliding mode in systems operating under uncertainty conditions," in *Proc. Int. Conf. Decision and Control*, pp.4591-4596, Kobe, Japan, Dec. 1996.
- [157] V. I. Utkin, "Variable structure systems with sliding modes," *IEEE Tran. Autom. Control*, vol. AC-22,2 no. 2, pp. 212-222, April 1977.
- [158] V. Y. Glizer, L. M. Fridman and V. Turetsky, "Cheap suboptimal control of an integral sliding mode for uncertain systems with delays" *IEEE Tran. Automatic Control*, vol. 52, no. 10, pp. 1892-1898, Oct. 2000.
- [159] W. Khalil and J. F. Kleinfinger, "A new geometric notation for open and closed-loop robots," in *Proc. IEEE Int. Conf. Robotics and Autom.*, 1986, pp. 1174-1179.
- [160] W. Khalil and J.-F. Kleinfinger, "Minimum operations and minimum parameters of the dynamic models of tree structure robots," *J. Robotics and Automation*, vol. Ra-3, no. 6, pp. 517-526, Dec. 1987.
- [161] W. Khalil and O. Ibrahim, "General solution for the dynamic modeling of parallel robots", *J. Intelligent Robot Systems*, vol. 49, pp.19-37, 2007.
- [162] W. Khalil and S. Guegan, "Inverse and direct dynamic modeling of Gough- Stewart robots," *IEEE Trans. Robotics*, vol. 20, no. 4, Aug. 2004.
- [163] W.-J. Cao and J.-X. Xu, "Nonlinear integral-type sliding surface for both matched and unmatched uncertain systems," *IEEE Tran. Automatic Control*, vol. 49, no. 8, pp. 1355-1360, August 2004.

- [164] W.-S. Yu, "Adaptive fuzzy PID control for nonlinear systems with h-infinity tracking performance", *IEEE Int. Conference Fuzzy Systems*, Vancouver, Canada, 2006, pp. 1010-1015.
- [165] X. Yu, B. Wang, Z. Galias, and G. Chen, "Discretization Effect on Equivalent Control-Based Multi-Input Sliding-Mode Control Systems," *IEEE Tran. Automatic Control*, vol. 53, no. 6, pp. 1563- 1569, Jul. 2008.
- [166] Y. Bolea, P. Dot, D. Pujol and A. Grau, "A parametric robust approach PID control for a laparoscopic surgery robot," in *Proc. IFAC 16th Triennial World Congress*, Prague, Czech Republic, 2005, pp 121-126.
- [167] Y. Dote and S. J. Ovaska, "Industrial application of soft computing: a review," in *proc. IEEE*, vol. 89, no.9, Sept. 2001
- [168] Y. Li and Q. Xu, "Dynamic analysis of a modified delta parallel robot for cardiopulmonary resuscitation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2005, pp. 233-238.
- [169] Y. Li, Sheng Qiang, X. Zhuang, and O. Kaynak, "Robust and adaptive backstepping control for nonlinear systems using RBF neural networks" *IEEE Tran. Neural Networks*, vol. 15, no. 3, pp. 693-701, May 2004.
- [170] Y. Nakamura and M. Ghodoussi, "A computational scheme of closed link robot dynamics derived by d'alembert principle," in *Proc. IEEE Int. Conf. Robotics and Autom.*, Philadelphia, PA, 1988, vol.3, pp. 1354-1360
- [171] Y. Pan, K. Dev Kumar, G. Liu and K. Furuta, "Design of variable structure control system with nonlinear time-varying sliding sector" *IEEE Tran. Automatic Control*, vol. 54, no. 8, pp. 1981-1986, Aug. 2009.
- [172] Y. Su, D. Sun, L. Ren and J. K. Mills, "Integration of saturated PI synchronous control and pd feedback for control of parallel manipulators," *IEEE Tran. Robotics*, vol. 22, no. 1, pp. 202-207, Feb. 2006.
- [173] Y. Ting, Y.-S. Chen and H.-C. Jar, " Modeling and control for a Gough-Stewart platform CNC machine," *J. Robotic Systems*, vol. 21, no. 11, pp. 609-623, 2004, DOI: 10.1002/rob.20039
- [174] Y. Ting, Y.-S. Chen and S.-M. Wang, "Task space control algorithm for Gough-Stewart platform," in *Proc. 38th Conf. Decision and Control*, Phoenix, Arizona-USA, 1999, pp 3857-3862.

- [175] Y. Wan and S. Wang, "Kinematic analysis and simulation system realization of Stewart platform manipulator," in *Proc. 4th Int. Conf. Control and Automation (ICCA '03)*, Montreal-Canada, 2003, pp.780-784.
- [176] Y. X. Su and B. Y. Duan, "The Application of the Stewart Platform in Large Spherical Radio Telescopes," *J. of Robotic Systems*, vol. 17, no.7, pp. 375-383, Feb. 2000.
- [177] Y. X. Su, B. Y. Duan, C. H. Zheng, Y. F. Zhang, G. D. Chen and J. W. Mi, "Disturbance-rejection high-precision motion control of a Stewart platform," *IEEE Tran. Control Systems Technology*, vol. 12, no. 3, May 2004.
- [178] Z. Gang and L. Hynes, "Neural network solution for forward kinematics problem of a Stewart Platform," in *Proc. IEEE Int. Conf. Robotics and Autom.*, Sacramento-California, 1991, pp.2650-2655.
- [179] Z. Ranko, V. F. Angel, G. G. Pedro, L. G. Angel, "An architecture for robot force and impact control," in *Proc. IEEE Cont. Robotics Automation and Mechatronics*, 2006, pp. 1-6.

Replies to examiners queries and suggestions

Report A queries and replies

- i) **Eq. (3.3) why does not it have direct solution? What solutions are you getting using Newton Raphson technique! What model you are trying to obtain from this solution?**

Answer: Equation (3.3) is as follows

$$l_i = \|Rp_i + r - b_i\| \quad (3.3)$$

And the statement in the thesis about the equation is as follows.

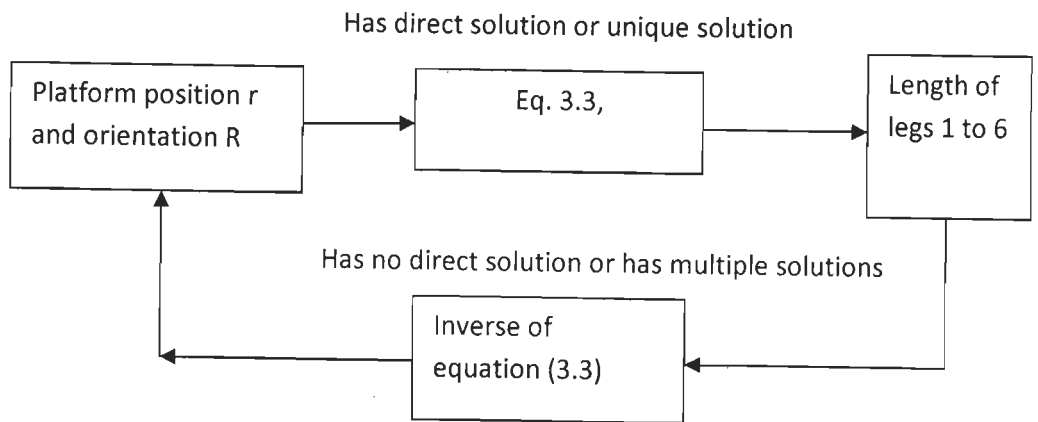
The solution of (3.3) is unique for a given platform position r and orientation R and can be directly calculated. This constitutes the solution of the inverse kinematics problem. The forward kinematics problem is finding the actual Cartesian space position and orientation $X=[r_x, r_y, r_z, \alpha, \beta, \gamma]$ given a set of leg lengths, $l_1, l_2, l_3, \dots, l_6$. This is a nonlinear equation and it has no direct solution.

With great excuse for the expression and miss understanding I created, the sense of the statement is not to say that equation (3.3) has no direct solution. Equation (3.3) is the inverse kinematics problem which computes the length of legs given platform center r and orientation R . So this has a unique solution and has direct solution.

The inverse of equation (3.3) is what we call the forward kinematics problem. It is to find the value of orientation matrix R and position of platform center r given six leg lengths. It is this problem which is nonlinear and which has no direct solution. By direct solution, we mean unique solution. For a given measurement of legs, there are many possible orientations and platform center positions. And analytical methods have found that there are around 40 possible solutions for a given set of leg length measurements. However, some of these solutions are not practically possible and hence out of the analytical solutions one has to select the practically feasible ones.

Using Newton Raphson technique, we get an approximate solution and it is not unique. If we take a different starting point, the solution will be slightly different. This is what we mean by 'it does not have direct solution'.

From the solution of Eq. (3.3) we obtain leg lengths for a given position and orientation of platform. However, from the inverse of equation 3.3, we get orientation and position of the platform for a given measured leg lengths. This is used as a feedback signal in control loops and hence it is very important.



ii) **Ref. to page 44, what is so important about the less time?**

The total time taken for the closed loop control system depends on the time used for the forward and backward paths. The less time in the neural network means, a less time in the backward loop which decreases the closed loop time and this again means a fast control algorithm which can track fast trajectory with a better accuracy

iii) **What is meant by leg PID controller?**

The Stewart manipulator has a fixed base and moveable platform which are joined by six prismatic and actuated joints. These prismatic joints which connect the base to the top platform are known as legs. A PID controller which controls the length of a single leg is what we call leg PID controller. Hence there are six PID controllers. The PID controllers used to control each leg are independent from each other, i.e. the input to each PID controller depends on the concerned leg only.

iv) **How do you guarantee the stability of the system in this method?**

The stability of the system is guaranteed by keeping the range of variation of the PID parameters within a specified limit. The range of variation of each parameter is separately checked with the PID controllers alone. The detail code of the S-function which implements the tuning algorithm takes care that the tuned value is not out of these range. The code used in the s-function `sfunfuzzy` which is used to insure that K_p always remains within bound is shown below. Similar code is used in K_d and K_i .

```

function sys=mdlDerivatives(t,x,u)
temp=x+u(1)*u(2)*Kp0;
if (temp>Kpmax)
    sys=[Kpmax];
else if (temp<Kpmin)
    sys=[Kpmin];
else
    sys = [temp];
  
```


end
end

v) **q_i and q_{id} are not defined in the text**

It is by mistake that I forget to define it.

q_i is actual measured leg length for the i^{th} leg and q_{id} is the desired leg length of the i^{th} leg.

vi) **What are ΔM , ΔC and ΔG ? Similarly what are G_m , C_m and M_m ?**

It is by mistake.

The parameters ΔM , ΔC and ΔG are the uncertainties in manipulator inertia matrix, Coriolis/centrifugal torque and gravitational torque respectively. It is assumed that the maximum values of these uncertainties are known. And hence, the parameters G_m , C_m and M_m are the maximum values of the uncertainties for Gravitational torque, Coriolis/Centrifugal torque and the manipulators Inertia matrix respectively.

vii) **The proof for the guaranteed reaching of the sliding surface in a finite time is not clear. Explain more clearly.**

The proof was not clear because some terms were not defined, as given in Vi above and some steps have also been omitted assuming that they were implied. The complete proof is given below.

Consider the dynamic equation of a Stewart platform manipulator

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= M^{-1} \left(J_p^{-T} \tau - C(x_1, x_2) X_2 - G(x_1) \right) \quad (5.1)\end{aligned}$$

Where x_1 is (6x1) state vector of Cartesian space positions and orientations and x_2 is (6x1) state vector of the Cartesian space velocities. Moreover, M is 6x6 manipulator inertia matrix, C is also 6x6 coriolis and centrifugal torque/force, G is 6x1 gravitational torque/force, J_p is 6x6 manipulator Jacobian, τ is the actuator torque

Let x_d be (6x1) vector of desired task space trajectories. Then, the task space tracking error and its rate vector are given as

$$e = x_d - x \quad (5.2)$$

$$\dot{e} = \dot{x}_d - \dot{x} \quad (5.3)$$

The following assumptions are taken

Assumption 1: Manipulator inertia matrix is non singular

Assumption 2: Manipulator Jacobian matrix is nonsingular throughout the workspace

In addition to the above two assumptions, in the design of robust controllers, there are some assumptions about the uncertainties.

Assumption 3: The uncertainties in the dynamic parameters are additive and can be expressed as nominal and deviation as shown below

$$M = M_N + \Delta M$$

$$C = C_N + \Delta C$$

$$G = G_N + \Delta G$$

Where M_N , C_N and G_N are nominal values of inertia matrix, coriolis/centrifugal torque and gravitational torques respectively and ΔM , ΔC and ΔG are the corresponding uncertainties. It is also assumed that the maximum values of the uncertainties are bounded as

$$\|\Delta M\| \leq M_m$$

$$\|\Delta C\| \leq C_m$$

$$\|\Delta G\| \leq G_m$$

Taking these uncertainties and the assumptions above and considering also external disturbance torque and actuator friction, the dynamic equation can be rewritten as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= (M_N + \Delta M)^{-1} (J^{-1}\tau - (C_N + \Delta C)x_2 - (G_N + \Delta G)) + d \end{aligned} \quad (5.4)$$

Where d is the sum of actuator friction and external disturbance torque

The above equation can be rewritten as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= (M_N)^{-1} (J^{-1}\tau - C_N x_2 - G_N) + \tilde{h} \end{aligned} \quad (5.5)$$

Where

$$\tilde{h} = -(M_N + \Delta M)^{-1} (\Delta C x_2 + \Delta G) + d + \text{res}$$

res is the residual error due to the approximations made in the inverse of inertia matrix.

The sliding manifold is given by

$$S = \Lambda e + \dot{e} \quad (5.6)$$

Where e and \dot{e} are as defined in (5.2) and (5.3)

Taking a Lyapunov function given by

$$V = \frac{1}{2} S^T S \quad (5.7)$$

Hence its derivative is

$$\dot{V} = S^T \dot{S} \quad (5.8)$$

$$= S^T (\lambda \dot{e} + \ddot{e}) \quad (5.9)$$

Differentiating (5.6) and using it in (5.9)

$$= S^T (\lambda \dot{x}_d - \lambda \dot{x} + \ddot{x}_d - \ddot{x}) \quad (5.10)$$

Using the dynamic equation (5.5) and the fact that $\dot{x} = x_1$ and $x_2 = \ddot{x}$

$$= S^T (\lambda \dot{x}_d - \lambda x_1 + \ddot{x}_d - M_N^{-1} (J^{-T} \tau - C_N - G_N) + \tilde{h}) \quad (5.11)$$

The control torque is given as

$$\tau = \tau_{eq} + \tau_s \quad (5.12)$$

Where

$$\tau_s = J^T (Kf_f(S)) \quad (5.13)$$

and

$$\tau_{eq} = J^T \{M_N (\lambda (\dot{x}_d - x_2) + \ddot{x}_d) + C_N x_2 + G_N\} \quad (5.14)$$

Hence using (5.12)-(5.14) into (5.11)

$$= S^T \left(\lambda \dot{x}_d - \lambda x_2 + \ddot{x}_d - M_N^{-1} \left(J^{-T} \left(J^T \{M_N (\lambda (\dot{x}_d - x_2) + \ddot{x}_d) + C_N x_2 + G_N\} + J^T (-Kf_s(S)) \right) - C_N - G_N \right) + \tilde{h} \right) \quad (5.15)$$

Rearranging and collecting like terms and simplifying

$$= S^T (M_N^{-1} (-Kf_s(S)) + \tilde{h}) \quad (5.16)$$

Substituting for \tilde{h} from (5.5)

$$= S^T M_N^{-1} (-Kf_s(S) - M_N ((M_N + \Delta M)^{-1} (\Delta C x_2 + \Delta G) + d + res)) \quad (5.17)$$

$$= -S^T M_N^{-1} (Kf_s(S) + \Delta C x_2 + \Delta G + d + res) \quad (5.18)$$

Hence for guaranteed reaching to the sliding surface, the following condition should hold true:

$$K \geq \|(\Delta Cx_2 + \Delta G) + d + \text{res}\|$$

Which means,

$$K \geq \|\Delta Cx_2\| + \|\Delta G\| + \|d + \text{res}\|$$

$$K \geq C_m \|x_2\| + G_m + \|d + \text{res}\|$$

- viii) What is the new sliding surface proposed as mentioned in section 5.3
Previous methods use sliding surface formulated from two parameters, the leg length error and its derivative. Hence they were not able to drive the synchronization error to zero. In our new method the new sliding surface that we proposed has three parameters: leg length error, its derivative and the cross coupling or synchronization error. Hence by forcing the system to move to the sliding surface and keeping it there, we are trying to achieve zero error in leg length of each leg and we are also making the synchronization error zero. And this helps to achieve a better tracking and improves the system's safe operation.
- ix) **Page 105, please check the statement that ‘ the transpose of the output matrix as gain of the sliding surface’**
It is not the author of the thesis who states that the use of the transpose of the output matrix as gain of the sliding surface, rather it is reference no. [42], which is a paper in an IEEE transaction. We have also commented on that stating, such a method needs a system with constant output matrix.

Report B queries and replies

I. Concerning main results and unresolved issues

Chapter 3

- The idea of using neural networks for estimation and particularly the application of neural networks for estimation of forward kinematics of Stewart platform is not new. However, neural networks have not been used in any controller implementation though the work on controllers itself is limited. The main reason for this is the issue of reliability. And hence, the main objective in this research is to strengthen the results obtained by other authors and to evaluate their performances when they are used in a closed loop control implementation. Accordingly, the research has summarized some of the previous works and we have used neural networks in closed loop control implementation and have shown their performances.
- The Stewart platform manipulator used in the research is completely different from the one existing in MATLAB toolbox. The one used in MATLAB is a very big one where the leg mass is very small compared to platform mass and can be neglected, both of the joints connecting the legs to the base and platform are universal joints and the actuated joint is cylindrical. In our research work, we used a medium sized Stewart platform where the mass of each leg is $\frac{1}{4}$ of the mass of the platform and it cannot be neglected in controller design. Moreover, the joint structure in our case is UPS (universal, prismatic and spherical). These being the main differences, we have tried to show the list of differences in Table 1. In addition to the mechanical structural differences, as our research was aimed at designing model based controllers, a complete inverse dynamic model of the system was necessary and we have done this completely by ourselves by taking mechanical parameters from papers cited as references. Therefore, the MATLAB SimMechanics toolbox has been used to implement the forward dynamic model of the Stewart platform manipulator which we analyzed and we have not used the existing one. At this point we would like to stress that it is not either lack of interest or inability to implement practically which prevented the authors from doing so. It is only lack of funds. The author has tried his best to find ways of practically implementing the control algorithms proposed. Some of these were, searching for internships in other universities, trying to test the performance of the controllers on single DOF robots such as inverted pendulum.

- Comparison of neural networks was made with Newton Raphson numerical method because it is the method which is used in all of the task space control algorithms available. Newton Raphson method is widely used because it is easier to implement and is faster than other numerical algorithms. It also needs small number of iteration if initial guess is properly selected. Some of the opponents of the use of neural network for forward kinematics accept the use of Newton Raphson method and therefore, in our research we compared neural network estimation with Newton Raphson method considering it as a bench mark algorithm. The idea of using RBF neural network has been explored but it is totally unacceptable. It is clear that the number of neurons in RBF networks proportionally increases with the number of samples used. For good approximation, especially for the precision work which we have been investigating, a large number of training samples have to be used. In our case, we used 6000 training samples. With this huge size of training samples, the number of RBF neurons was extremely big and hence it is not suitable for the application at hand.

Table 1 Mechanical parameters of Stewart platform used in our research and its comparison with the one in MATLAB toolbox

No.	Items or parameters	Stewart platform in MATLAB	Stewart platform used in our research
1	Base radius	3m	0.8m
2	Platform radius	1m	0.5m
3	Height	2m	1.5m
4	Mass of upper leg	143.9229Kg	4Kg
5	Mass of lower leg	92.1107Kg	92.1107Kg
6	Inertia of lower leg	Diag(43.02 43.02 0.15) KgM ²	Diag(0.03 0.03 0.002)KgM ²
7	Inertia of upper leg	Diag(67.19 67.19 0.18) KgM ²	Diag(0.75 0.75 0.018) KgM ²
8	Mass of platform	10952.3414Kg	32Kg
9	Inertia of platform	Diag(304.5 304.5 608.5) KgM ²	Diag(2 2 4) KgM ²
10	Top thickness	0.05m	0.02m

Chapter 4

- In our research we followed two main directions in applying soft computing techniques for the control of Stewart platform manipulator. The first one is to improve existing methods. And hence the industrially accepted and widely used controller is independent leg PID controller. Therefore our fuzzy logic based tuning was aimed at improving the performance of independent leg PID controllers. Though the idea of tuning PID controllers using fuzzy logic is not a new idea, its application for Stewart platform control is new. Moreover, unlike fuzzy tuned PID controllers designed for other systems and which are available in the literature, the fuzzy logic tuned PID controller has one new feature which is used to consider the coupling effect of legs. A coefficient matrix C is used to multiply the error from each leg and the result is used as an input to the fuzzy logic. This structure is used to solve the problem of synchronization which is another drawback of independent leg PID.
- The fuzzy logic system used is Mamdani type and for each leg a separate fuzzy logic controller is used. Hence the parallel structure of the complete control system has been maintained. For each fuzzy logic block, two inputs and one output were used. The inputs were weighted sum of errors and its rate, the output is a value used for tuning the PID parameters. Each of the inputs and the output were divided into five linguistic classes and triangular membership functions were used for all of them. Three and seven classes have been tested but they have drawbacks of less performance and decrease in speed of operation respectively. The membership functions have been tuned manually for optimal performance and slightly skewed membership functions resulted in better performance. The triangular membership functions were selected to speed up the algorithm. The rule base used is as shown in Table 2. The rule base is derived from step response analysis though the control objective is trajectory tracking.

Table 2 Rule base of the fuzzy logic PID tuner used for each leg

		Error_rate				
		NB	NS	Z	PS	PB
Error	NB	NB	NB	NS	NS	Z
	NS	NB	NS	NS	Z	PS
	Z	NS	NS	Z	PS	PS
	PS	NS	Z	PS	PS	PS
	PB	Z	PS	PS	PB	PB

The tuning algorithm used in the research is as follows. The algorithm is a modification of some of the algorithms reported in the literature for other systems. The proportional gain is tuned based on the present error while the derivative constant is tuned based on the error rate. This is derived from the basic objective of the two parameters. That is, proportional constant takes care of present effects while derivative constant takes care of future predictions. The integrator is tuned proportional to the product of the error and error rate. This three algorithms are separately written as s-functions in MATLAB

$$K_p(k+1) = K_p(k) + \alpha_r(k)e(k)K_p(0)$$

$$K_D(k+1) = K_D(k) + \alpha_r(k)\dot{e}(k)K_D(0)$$

$$K_I(k+1) = K_I(k) + \alpha_r(k)e(k)\dot{e}(k)K_I(0)$$

Initially, the proportional, derivative and integral parameters are set to nominal values. The variation of the parameters with respect to the fuzzy logic rule base can be analyzed from the rule base and the algorithm. For example, if both error and error rate are zero, the output of the fuzzy logic system will be zero and the values of the parameters remain the same. If error increases to positive small and error rate is negative, then the proportional parameter increases and the derivative and integral parameters are decreased. The simple PID controller used for comparison has the same parameters with the nominal parameters used in the tuning. This nominal parameter values are obtained through manual tuning.

- The stability issue raised is a real and important concern. We share and agree that stability analysis has to be done. But it was not overlooked in our research also. Though formal mathematical proof was not presented, the algorithm has carefully considered in its implementation. This is done as follows. The block which implements the tuning algorithm is implemented in MATLAB using s-function and the values of the parameters, K_p , K_D , and K_I are increased or decreased between an upper limit and lower limit respectively. And the stability of the manipulator with worst case conditions has been checked separately. Hence it is guaranteed that the manipulator will remain stable in spite of the parameter variations.