

**T.C.
MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN
PURE AND APPLIED SCIENCES**

**COMPARISON OF CONTACT FORCE CONTROL
STRATEGIES ON DIFFERENT ROBOT ARM TYPES**

**Hüseyin YALTIRIK
(Mechanical Engineering)**

**THESIS
FOR THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING PROGRAMME**

**SUPERVISOR
Prof. Dr. Abdülkerim KAR**

İSTANBUL 2007

ACKNOWLEDGMENT

I would like to thank Prof. Dr. A.Kerim KAR for giving me the opportunity to work on this interesting project and for the help and guidance.

I wish thank to Ass. Prof. Dr. Haluk KÜÇÜK for helping me with my doubts and questions about learning and using Parameter Optimization Toolbox of MATLAB and also Ass. Prof. Dr. Bülent EKİCİ for his timely suggestions and encouragement. Working together with them has not only been a good learning experience, but also a great pleasure. In addition I would like to thank Nebahat ARSLAN AVCI for moral encouragement.

TABLE OF CONTENTS

	PAGES
ÖZET	v
ABSTRACT	vi
LIST OF SYMBOLS	vii
ABBREVIATIONS	ix
LIST OF FIGURES	x
LIST OF TABLES	xii
PART I. INTRODUCTION AND OBJECTIVES	1
I.1. INTRODUCTION	1
I.2. OBJECTIVES AND SCOPE OF THE THESIS	2
PART II. GENERAL BACKGROUND	3
II.1. CONTROL OF ROBOTIC MANIPULATORS	3
II.2. FORCE CONTROL	5

II.2.1. Compliance (Stiffness) Motion Control	7
II.2.1.1. Passive Compliance	8
II.2.1.2. Active Compliance	9
II.2.2. Hybrid Position/Force Control	11
II.2.3. Impedance Control	14
II.2.3.1. Instantaneous Mode Impedance Control	17
II.2.3.2. Adaptive and Robust Impedance Control	17
II.2.3.3. Sliding Mode Impedance Control	18
II.2.3.4. Hybrid Impedance Control	18
II.2.3.5. Extended Impedance Control	19
II.2.3.6. Spatial Impedance Control	19
II.2.3.7. Neural Networks Control Based Impedance Control	19
II.2.4. Resolved Acceleration Control	20
II.2.5. Computed Torque Control	21
II.2.6. Parallel Force/Position Control	21
PART III. MODELING, SIMULATION AND CONTROL	23
III.1. MATHEMATICAL MODEL OF ROBOT ARMS.....	23
III.1.1. Modeling Of The 3 DOF Spherical Robotic Manipulator	23
III.1.2. Modeling Of The 3 DOF Cartesian Robotic Manipulator	30
III.2. SIMULATION OF ROBOTIC MANIPULATORS	32
III.2.1. Simulink Models of The Force Control Algorithms	33
III.2.2. Force Control Models For Cartesian Type Of Manipulator	34
III.2.2.1. Hybrid Control	35
III.2.2.2. Resolved Acceleration Control (RAFC)	35
III.2.2.3. Parallel Force/Position Control	36
III.2.3. Force Control Models For Revolute Type Of Manipulator	36
III.2.3.1. Hybrid Control	37
III.2.3.2. Resolved Acceleration Control (RAFC)	37
III.2.3.3. Parallel Force/Position Control	38
PART IV. PART IV. RESULTS AND EVALUATIONS	39
IV.1. STEP INPUT RESPONSE FOR POSITION CONTROL	39
IV.1.1. Position and Velocity Gain of Hybrid Control of Cartesian Type of Robot	40
IV.1.2. Position and Velocity Gain of RAFC of Cartesian Type of Robot	40
IV.1.3. Position and Velocity Gain of Parallel Force/Position Control of Cartesian Type of Robot.....	41
IV.1.4. Position and Velocity Gain of Hybrid Control of Revolute Type of Robot	42
IV.1.5. Position and Velocity Gain of RAFC of Revolute Type of Robot	42
IV.1.6. Position and Velocity Gain of Parallel Force/Position Control of Revolute Type of Robot	43
IV.2. REDUCED TAGUCHI L9 (3⁴) TABLES	44

IV.2.1. Taguchi L9 (3 ⁴) Table of Hybrid Control of Cartesian Type Of Robot Arm	44
IV.2.2. Taguchi L9 (3 ⁴) Table of RAFC of Cartesian Type of Robot Arm	47
IV.2.3. Taguchi L9 (3 ⁴) Table of Parallel Force/Position Control of Cartesian Type of Robot Arm.....	49
IV.2.4. Taguchi L9 (3 ⁴) Table of Hybrid Control of Revolute Type Of Robot Arm	51
IV.2.5. Taguchi L9 (3 ⁴) Table of RAFC of Revolute Type of Robot Arm	53
IV.2.6. Taguchi L9 (3 ⁴) Table of Parallel Force/Position Control of Revolute Type of Robot Arm	55
IV.3. EVALUATION OF FINAL RESULTS	57
PART V. CONCLUSIONS	59
REFERENCES.....	61
APPENDICES	66
APPENDIX A. MATLAB FILES OF CARTESIAN TYPE OF ROBOT ARM	67
APPENDIX B. MATLAB FILES OF REVOLUTE TYPE OF ROBOT ARM	72

ÖZET

FARKLI ROBOT KOLLARI ÜZERİNDEKİ TEMAS KUVVETİ KONTROL STRATEJİLERİNİN KARŞILAŞTIRILMASI

Günümüzde robotlar çeşitli alanlarda kullanılmaktadırlar. Robot kollarının çevreyle ve ya bir nesneyle temasta olduğu bir çok önemli uygulamalar vardır. Statik veya dinamik hareketli, bir cismin kontrolü esnasında cisim veya robot kolunun zarar görmemesi gerekmektedir. Bu tür durumlarda etkileşim kuvveti önemlidir. Etkileşim kuvvetlerinin hassas kontrolüne bağlı olarak görev başarılabilir veya yapılamaz. Bu yüzden etkileşim kuvvetlerini kontrol etmek için bir kuvvet algoritması geliştirilmelidir. Bu tezde statige yakın temas durumu için bir kuvvet algoritması geliştirilecektir. Daha sonra bu durum dinamik durumlar için de genişletilecektir.

Bu tezin amacı robot kolu ile çevreye temas halinde istenen etkileşim kuvvetinin başarılı bir şekilde oluşturulmasını sağlayan bir kontrol stratejilerini karşılaştırmaktır.

Bu tezde, kuvvet kontrolü hakkında yazılmış yayınlar incelenmekte, robot kolu modelinin oluşturulması yapılmakta, temas anında oluşan kuvvetlerin değerlendirilmesi, uygun bir kuvvet kontrol algoritması oluşturulması yapılmaktadır ve bunların karşılaştırılması yapılacaktır.

Hüseyin YALTIRIK

ABSTRACT

COMPARISON OF CONTACT FORCE CONTROL STRATEGIES ON DIFFERENT ROBOT ARM TYPES

Nowadays robots are used in various areas. There are extremely important applications where the robot arm tip comes in contact with the environment or an object. During controlling an object, static or in motion, the object or the robot arm should not be damaged. The interaction forces are important in such conditions. Whether the task succeeds or fails depends on how accurate the interaction forces are controlled. The interaction forces are changed depending on the motion of the robot arm. Therefore, to control interaction forces a force control algorithm must be developed. In this research a force control algorithm will first be developed for the quasi-static contact tasks, then it will be extended to the dynamic cases.

The goal of this thesis is to compare force control strategies to achieve the desired interaction forces between the robot arm tip (end-effector) and the environment during contact tasks.

This thesis includes a literature survey about force control, modeling of a robot arm, evaluating measured forces during contact, constructing a suitable force control algorithm, and comparing these control strategies.

Hüseyin YALTIRIK

LIST OF SYMBOLS

B_d	: Desired damping in cartesian space (3x3)
$B(\theta)$: Coriolis and Centrifugal term in joint space (3x3)
C_e	: Compliance of environment (m/N)
I	: Unit matrix
f	: Force (N)
f_d	: Desired force (N)
f_e	: External force (N)
g	: Gravity constant (m/sec ²)
$G(\theta)$: Gravity term in joint space (m/sec ²)
J	: Jacobian matrix
K	: Kinetic energy
K_e	: Stiffness of environment (N/m)
K_v	: Velocity gain
K_p	: Position gain
l	: Length of a link
L	: Lagrangian term
m	: Mass of a link
M_d	: Desired mass in cartesian space (3x3)
$M(\theta)$: Inertia of links in joint space (3x3)
MxO	: Maximum overshoot
P	: Potential energy
P_i	: Initial position
P_l	: Last position
r	: radius

$\hat{\mathbf{R}}^{-1}$: Forward dynamics
S	: Selection matrix
T	: Time (sec)
T	: Torque (N.m)
θ	: Angular position (rad)
$\dot{\theta}$: Angular velocity (rad/sec)
$\ddot{\theta}$: Angular acceleration (rad/sec ²)
x	: Actual position of end effector (m)
x_0	: Initial position (m)
x_e	: Deformation of contact surface (m)
\dot{x}	: Velocity of end effector (m/sec)
\dot{x}_d	: Desired velocity (m/sec)
\ddot{x}	: Acceleration of end effector (m/sec ²)
\ddot{x}_d	: Desired acceleration (m/sec ²)

ABBREVIATIONS

C_i	: Cosine of an i^{th} link
C.T.	: Calculation Time
D-H	: Denavi Hartenberg
dir. kin.	: Direct Kinematics
DOF	: Degree of Freedom
Sqrt_{EMS}	: Square root of totals of ER^2 , MPR^2 and STR^2
$\text{Sqrt}_{\text{EMSC}}$: Square root of totals of ER^2 , MPR^2 , STR^2 and CTR^2
Error	: Steady State Error
ER	: Division Error by Maximum of Errors
Fwdyn	: Forward Dynamics
Hybrid	: Hybrid Position/Force
MPR	: Division %MxO by Maximum of %MxO's
MxO	: Maximum Overshoot
%MxO	: Percentages of difference wrt. Desired Force
PID	: Proportional, Integral and Derivation
CTR	: Division by Maximum of CT's
RAFC	: Resolved Acceleration Force Control
S_i	: Sine of an i^{th} link
S.T.	: Settling Time

LIST OF FIGURES

	<u>PAGES</u>
Figure II.1. Block diagram of a generic force-based, explicit force controller.....	6
Figure II.2. Block diagram of a generic position-based, explicit force controller ..	6
Figure II.3. Conceptual sketch of RCC hand	8
Figure II.4. Robot manipulator in contact with hard surface	10
Figure II.5. Hybrid position/force control	12
Figure II.6. Impedance control block diagram for equation (II.2.13)	16
Figure II.7. Block diagram of instantaneous model impedance control	17
Figure II.8. Resolved acceleration force control	20
Figure II.9. Parallel force/position control	21
Figure III.1. 3 DOF revolute type of robot arm	24
Figure III.2. 3 DOF Cartesian robot arm	31
Figure III.3. Force sensor simulink block diagram	34
Figure III.4. Selection matrix block diagram	34
Figure III.5. Simulink block diagram of the hybrid control of cartesian type of robotic manipulator	35
Figure III.6. Simulink block diagram of the RAFC of cartesian type of robotic manipulator	35
Figure III.7. The structure of the block of selection matrix of RAFC.....	36
Figure III.8. Simulink block diagram of the parallel force/position control of cartesian type of robotic manipulator	36
Figure III.9. Simulink block diagram of the hybrid control of revolute type of robotic manipulator	37
Figure III.10. Simulink block diagram of the RAFC of revolute type of robotic manipulator	38

Figure III.11.	Simulink block diagram of the parallel force/position control of revolute type of robotic manipulator	38
Figure IV.1	Step input response of position of hybrid control of cartesian type of robot	40
Figure IV.2	Step input response of position of RAFC of cartesian type of robot ..	41
Figure IV.3	Step input response of position of parallel force/position control of cartesian type of robot	41
Figure IV.4	Step input response of position of hybrid control of revolute type of robot	42
Figure IV.5	Step input response of position of RAFC of revolute type of robot ..	43
Figure IV.6	Step input response of position of parallel force/position control of revolute type of robot	43
Figure IV.7	The best value of the force and position graph of Hybrid Position/Force Control of cartesian type of robot	46
Figure IV.8	The worst value of the force and position graph of Hybrid Position/Force Control of cartesian type of robot	46
Figure IV.9	The best value of the force and position graph of RAFC of cartesian type of robot	48
Figure IV.10	The worst value of the force and position graph of RAFC of cartesian type of robot	48
Figure IV.11	The best value of the force and position graph of force/position control of cartesian type of robot	50
Figure IV.12	The worst value of the force and position graph of parallel force/position control of cartesian type of robot	50
Figure IV.13.	The best value of the force and position graph of Hybrid Position/Force Control of revolute type of robot	52
Figure IV.14.	The worst value of the force and position graph of Hybrid Position/Force Control of revolute type of robot	52
Figure IV.15	The best value of the Force and Position graph of RAFC of revolute type of robot	54
Figure IV.16	The worst value of the Force and Position graph of RAFC of revolute type of robot	54
Figure IV.17	The best value of the force and position graph of parallel force/position control of revolute type of robot	56
Figure IV.18	The worst value of the force and position graph of parallel force/position control of revolute type of robot	56

LIST OF TABLES

	<u>PAGES</u>
Table III.1. Robot arm D-H representation	24
Table III.2. Block list and their properties that are used in simulink models	33
Table IV.1. Taguchi arrays of hybrid control of cartesian type of robot.....	45
Table IV.2. Normalizing Taguchi arrays of hybrid control of cartesian type of robot	45
Table IV.3. Taguchi arrays of RAFC of cartesian type of robot	47
Table IV.4. Normalizing Taguchi arrays of RAFC of cartesian type of robot.....	47
Table IV.5. Taguchi arrays of parallel force/position control of cartesian type of robot	49
Table IV.6. Normalizing Taguchi arrays of parallel force/position control of cartesian type of robot	49
Table IV.7. Taguchi arrays of hybrid control of revolute type of robot.....	51
Table IV.8. Normalizing Taguchi arrays of hybrid control of revolute type of robot	51
Table IV.9. Taguchi arrays of RAFC of revolute type of robot	53
Table IV.10. Normalizing Taguchi arrays of RAFC of revolute type of robot.....	53
Table IV.11. Taguchi arrays of parallel force/position control of revolute type of robot	55
Table IV.12. Taguchi arrays of parallel force/position control of revolute type of robot	55
Table V.13. Row number of the best and the worst response	57
Table V.14. PID levels of the best and the worst response	57
Table V.15. MxO and ST of the best and the worst response.....	58

PART I.

INTRODUCTION AND DISCUSSION

I.1. INTRODUCTION

Research on robot force control has flourished in the past decades. Such a wide interest is motivated by the general desire of providing robotic systems with enhanced sensory capabilities. Robots using force, touch, distance, and visual feedback are expected to autonomously operate in structured environments other than the typical industrial shop floor.

It should be no surprise that managing the interaction of a robot with the environment by adopting a purely motion control strategy turns out to be inadequate. The unavoidable modeling errors and uncertainties may cause a rise of the contact forces ultimately to an unstable behavior during the interaction. On the other hand, since the early work on telemanipulation, the use of force feedback was conceived to assist the human operator in the remote handling of objects with a slave manipulator. More recently, cooperative robot systems have been developed where two or more manipulators are to be controlled so as to limit the exchange force and avoid squeezing of commonly held object.

The force control problem is traced as the natural evaluation of the motion control problem, where feedback of the contact force as provided by a force/torque sensor is used to manage the interaction of a robot manipulator with a scarcely structured environment. In this respect, those control strategies, e.g. hybrid

position/force control, devised for interaction with an accurately modeled environment are not treated.

I.2. OBJECTIVES AND SCOPE OF THE THESIS

Nowadays robots are used in various areas. There are extremely important applications where the robot arm tip comes in contact with the environment or an object. During controlling an object, static or in motion, the object or the robot arm should not be damaged. The interaction forces are important in such conditions. Whether the task succeeds or fails depends on how accurate the interaction forces are controlled. The interaction forces are changed depending on the motion of the robot arm. Therefore, to control interaction forces a force control algorithm must be developed. In this research a force control algorithm will first be developed for the quasi-static contact tasks, then it will be extended to the dynamic cases.

The goal of this thesis to compare force control strategies to achieve the desired interaction forces between the robot arm tip (end-effector) and the environment during contact tasks. Energy methods will be utilized to derive equations of motions. MATLAB package will be used for simulation and control studies.

TAGUCHI Methods will be utilized to find the best combination of control parameters for each different robot arm. Then results will be compared among robot arms to find the best combination.

After an introduction in Part I, this thesis includes a literature survey about force control in Part II, modeling of a robot arm, evaluating measured forces during contact and constructing a suitable force control algorithm in Part III. Results of the Control Strategies and Taguchi Method applications are presented in Part IV. Conclusions reached in this work are listed in the Part V. Referred published and nonpublished references are included in the References chapter. MATLAB files of robot arms are included in the Apendieces.

PART II.

GENERAL BACKGROUND

In this chapter, a literature survey and force control types of robot manipulators will be presented about to force control of robot manipulators. In the subsection II.1. Control of Robotic Manipulators the definition of a robot, the necessity of force control algorithms and a brief history of studies were done on force control of robot manipulators. Then, in the subsection II.2. Force control main force control algorithms are presented.

II.1. CONTROL OF ROBOTIC MANIPULATORS

Robotic manipulators are dynamic systems and interact with other dynamic systems such as mechanical objects, tools, other robots and humans or environment [1].

The majority of contemporary robotic manipulators are utilized for tasks such as material transfer, arc welding, spot welding, spray painting, pick-and-place, and laser cutting to name a few, in which the manipulators do not come into touch (non-contact tasks) [2], with the objects in the workspace (environment). Pure position control (motion control or free motion control) is sufficient enough for the above tasks. However, many tasks, such as assembly, drilling, grinding, deburring, polishing, driving a screw, opening and closing a door, turning a crank, handling fragile objects, etc., which involve extensive contact with the environment, are

require the control of the interaction (contact tasks) forces arisen during the contact (constrained motion) [2-6]. In other words, for execution of such tasks, traditional pure position control strategies are obviously unsuitable since the environmental dynamics will affect the stability and the performance of the robot's position control loop [2, 7-10].

The problems of controlling the mechanical interaction between a manipulator and its environment have been addressed by many researchers. The inadequacies of conventional position control are widely recognized and the alternatives are typically referred to as force control, compliance, compliant motion control, fine motion control, impedance control [7, 11-13]. The pure force control along the normal direction of the dynamic constraint is unsuitable since the contact discontinuity will disable the force feedback loop and will in turn cause the robot to be driven by the commanded force alone and crash to the environment. The traditional pure force control is also unsuitable since the environmental dynamics will affect the stability and the performance of the robot's position control loop [7].

The main force control types that they base many force control algorithms are direct force control, compliance (stiffness) control –passive and active-, impedance control, hybrid position/force control, resolved acceleration force control.

Research on compliant motion control of robotic manipulators actually begins in the early 1970s. Since these early effects, a lot of approaches have been developed to ensure compliant motion. Salisbury and Craig have developed a so-called stiffness control scheme, in which the position trajectories are adjusted in terms of the force feedback, as the name would imply. Whitney, Paul and Shimano have suggested a method called damping control, in which the velocity of the end-effector is modified by means of feedback. Later, Hogan has proposed an impedance control scheme as a generalization of the two previous control schemes. Hogan used impedance control in experiments involving contact transitions. His implementation achieved stability against a stiff environment and avoided controller transitions and inverse kinematic computations.

Vossoughi and Donath also employed impedance methods in tests with varying environment stiffness.

Furthermore, Raibert, Craig and Mason have developed a control scheme, termed hybrid position/force control, which controls the position of the end-effector and the force exerted by the end-effector explicitly. Besides, there have existed the

so-called explicit force control and hybrid impedance control, a comprehensive survey of compliant motion control has been given by Whitney. Nowadays, hybrid position/force control (hybrid control) and impedance control have gained considerable popularity over the other existing compliant motion schemes, thereby becoming the major approaches for ensuring the compliant motion [12].

Actively compliant fingertips containing electro-rheological fluids were explored by Akella, Siegwart, and Cutkosky, who managed to control the dumping characteristics of fluid-filled fingertip to minimize for oscillations following the onset of contact [14].

Yocef-Toumi and Gutz developed a dimensionless representation of impact behavior and used integral force compensation with velocity feedback to improve impact response [14].

Khatib and Burdick proposed a method for dissipating impact oscillations that involved increasing the velocity gains of a proportional-derivative force feedback controller for a limited time following impact. By disabling the high velocity feedback gain after the impact oscillations decayed, response times to subsequent force commands were decreased [14].

More recently, Qian and de Schutter presented an active nonlinear damping approach. This method examined the force signal derivative, and effectively added a coulomb friction term to the output force command when this derivative exceeded a threshold [14].

II.2. FORCE CONTROL

Force feedback applications can be divided in two types:

Applications, whereby the contact force is an essential part of the process itself (grinding, polishing, deburring, etc.). Both the position of the end effector and the force exerted by the end effector on the environment have to be controlled simultaneously.

Applications, whereby the contact forces can be used as a source of information on the actual position of the endpoint (gripper) of the robot relative to the environment. Most operations during the assembly process belong to this category. Force feedback offers an important means to enlarge the allowed region of uncertainty, thus avoiding more severe requirements on the positioning accuracy of

the robot and the peripheral equipment. Compliance control and impedance control are examples of this type of force control [15].

A pure force control with proportional gains (K_p) and (K_v) can be formulized in the eq. (II.1).

$$\tau = -K_v f + K_p (f - f_d) \quad (II.1)$$

Force control techniques can be classified by getting control signals from a force sensor in direct way or from position and velocity in indirect way. The first is called explicit force control and, the second is implicit force sensor

Explicit force control also has two types that are force based and position based explicit force control.

Force based explicit force control describes a force controller that compares the reference and measured force signals, processes them, and provides an actuation signal directly to the plant. General block diagram is;

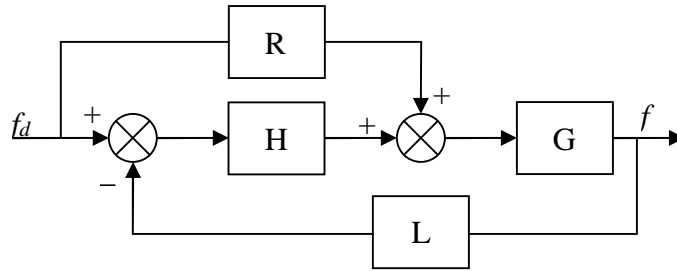


Figure II.1. Block diagram of a generic force-based, explicit force controller

Position based explicit force control: A second class of explicit force controllers consists of those based on inner position loop see Figure II.2. These controllers were probably implemented first for practical reasons – most commercial manipulators have built in position controllers and do not allow direct access to actuator torques. The commanded force is transformed into a commanded position through and admittance, which is described as the inverse of second order impedance. Joint torques are obtained through the transpose of the Jacobian, and gravity compensator is employed [16].

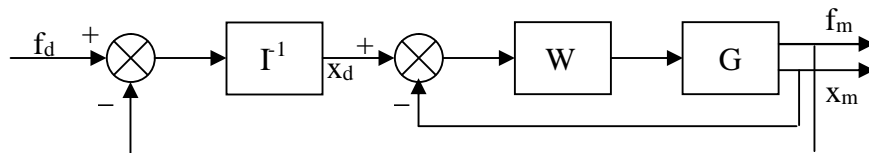


Figure II.2. Block diagram of a generic position-based, explicit force controller

Implicit force control requires position or velocity feedback while the feedback gains are adjusted to simulate environment stiffness; no force sensor is required [17].

Previous conventional force control algorithms could not control the robot effectively with considering the variation of working environment, and sensed values cannot be assumed to be precise unless considering sensing delay time of the sensor, mechanical vibration due to sensor structure. This method is very sensitive to inaccuracies in the dynamic model [2, 18].

II.2.1 Compliance (Stiffness) Motion Control

Compliance motion control some where is called as stiffness control, concerned with the control of manipulator in contact with its environment, represents one of the most challenging research areas of today's robotics. Compliant motion of a manipulator occurs when the manipulator position is constrained by the task geometry.

The idea behind the compliance control can be described as force and position relation of force exerted on a spring. This relation is formulated as,

$$f=K_e(x-x_0) \quad (II.2)$$

where K_e is stiffness, x is actual position x_0 initial position and f is force exerted on the spring.

If we know the displacement then we can find the force exerted to the spring. It should be given attention to the equation. When the force can be controlled then position can be controlled indirectly. It is true for vise. To control interaction force position control is required. This is preferable than the use of pure force control technique that is affected by the chattering and unstable of the force sensor signaling.

Environment that the robot manipulator is in contact has a spring property and the interaction force can be simulated by eq. (II.2). Compliance is the inverse of the stiffness and derived as

$$C_e=K_e^{-1} \quad (II.3)$$

C_e and K_e are compliance and stiffness of the environment respectively.

In the many literature, compliance and stiffness controls have the same meaning.

Compliant motion may be produced either by a passive mechanical compliance built in to the manipulator, by an active compliance implemented in the control servo loop [19].

II.2.1.1. Passive Compliance

Compliant motion may be produced either by a passive compliance built into the manipulator structure. A passive compliance, well known as RCC (Remote Center Compliance), can be realized by a mechanical device composed of springs and dampers for the purpose of reducing the endpoint stiffness [5]. Using such a device, as illustrated in Figure II.3, certain tasks such as inserting a peg in a hole can be accomplished under pure position control. A passive compliance is simple structure and therefore inexpensive, for the feedback of the contact forces and additional calculations are not required. However, a disadvantage of these devices is that their applications are necessarily limited to very specific tasks.

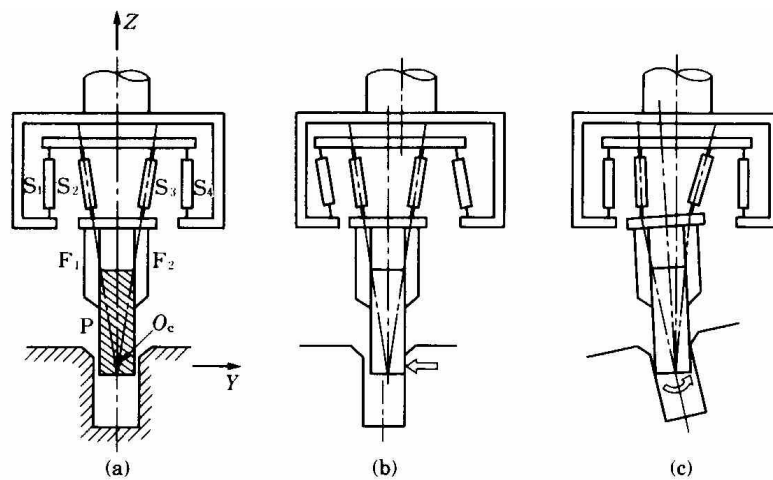


Figure II.3. Conceptual sketch of RCC hand. (a) Structure, (b) Compliance to horizontal force, (c) Compliance to moment [9]

For instance, passive compliance can not be applied for different peg-in-hole insertion tasks. In order to achieve a wider application range, the active compliance is more preferable. In contrast to the passive compliance, the active compliance is produced by a control loop where the feedback of contact forces is generally indispensable. Since an advantage of the active compliance is its reprogrammability to suit various tasks, which is considered to be one prerequisite for advanced manipulators, to realize the active compliance becomes the main current of the compliant motion control, despite the necessity of complicated force sensors and advanced manipulators, to realize the active compliance becomes the main current of the compliant motion control, despite the necessity of complicated force sensors and advanced control techniques [9, 12, 15, 19, 20].

II.2.1.2. Active Compliance

For example, consider a drawing compliant motion task in which an intelligent robot with vision sensors is required to draw a person's resemblance on a sketchpad with charcoal pencil based on the vision sensing information. In such a case, pure position control is apparently unacceptable, since slight errors along the normal direction of the sketchpad may result in losing contact with the sketchpad or on the contrary pressing too strongly to break the charcoal pencil. In order not to break the charcoal pencil or paper, the interaction force between the charcoal pencil and the sketchpad must be controlled in the normal direction, whereas in order to make the sketch appear similar enough to the person, position control is required along the plane of the sketchpad. It is clear that both position control and force control are needed for compliant tasks [12].

Compliant manipulation fundamentally requires the controlled robot not only to follow the input trajectory exactly in free space, but also manipulate adaptively on dynamic environments while making compliant contact with the environment's dynamic constrained space [8].

Mason first formalized the compliance coordinate frame for various contact tasks. According to his definition of "Compliant Motion", the contact forces modify the manipulator's trajectory during motion when its position is constrained by the environment geometry (kinematic constraint) [8].

For most contact tasks, we not only require the robot to move (both free motion and kinematically constrained motion) but also impose mechanical manipulations on the dynamic environments and/or its operated environment. The dynamics of the environment, in turn, will affect the performance of the robot's end-effector. Therefore, there is also a problem of how to be compliant with the dynamic environment when performing the required manipulation on it. Waibel and Kazerooni studied the robot's compliant motion control when considering the dynamic environment from the viewpoint of an unstructured model [8].

Assumed that the robot manipulator is described as

$$M(\theta)\ddot{\theta} + B(\theta, \dot{\theta}) + G(\theta) = \tau + J^T f_e \quad (II.4)$$

The joint torque is computed in

$$\tau = J^T f_e \quad (II.5)$$

where f_e is the external force occurs during contact end effector with environment. And it is produced when the surface is deformed during contact task. This relation is described in mathematical expression in eq. (II.6)

$$f_e = K_e x_e \quad (II.6)$$

$$x_e = x - x_d \quad (II.7)$$

where K_e is environment stiffness, x_e Cartesian displacement, x is the actual position and x_d is desired or commanded position.

The joint torque now is

$$\tau = J^T K_e x_e \quad (II.8)$$

The derivative transformation of joint velocity and linear velocity that is called Jacobian is derived as

$$J = \delta x / \delta \theta \quad (II.9)$$

If δx and $\delta \theta$ very small then we get the relation

$$\delta x = J \delta \theta \quad (II.10)$$

$$x_e = \delta x = J \delta \theta \quad (II.11)$$

Substituting eq. (II.9) into eq. (II.11)

$$\tau = J^T K_e J \delta \theta \quad (II.12)$$

$$K_p = J^T K_e J \quad (II.13)$$

$$\tau = K_p \delta \theta \quad (II.14)$$

This relation was derived by Salisbury [21] and was called active stiffness [22-24].

For example, suppose the tool tip comes into contact with a hard surface as shown in Figure II.4.

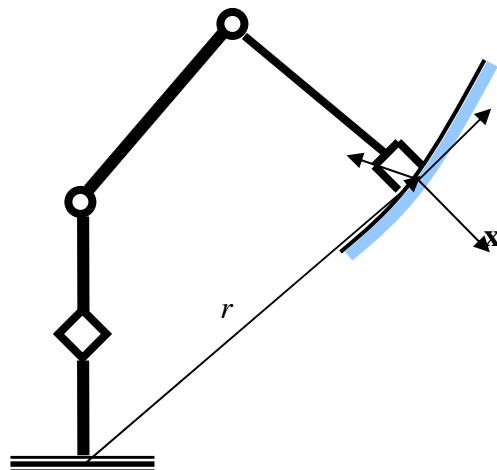


Figure II.4. Robot manipulator in contact with hard surface

Here the environment imposes a natural position constraint, which prevents the tool tip from penetrating the surface. When the tool is in contact with surface, it no

longer makes sense to attempt to control position in the direction x orthogonal to surface. However, we can control tool-tip force along x , for example, to maintain contact with the surface. Furthermore, we can simultaneously control the tool-tip position along directions tangent to surface. In this way the tool can be made to slide along the surface [11].

This is called a compliant along the surface, since the trajectory is made to comply with the natural constraints imposed by surface [11].

Compliance control based algorithms have been developed such as compliance based fuzzy control [25,26], multi point compliance control [27], combined compliance control [28] and compliance control using self-controlled stiffness function [29] etc.

II.2.2 Hybrid Position/Force Control

Hybrid control, initially proposed by Raibert and Craig, is based on an orthogonal decomposition of the task space. Because the position of the end-effector and the contact force with the environment along one degree of freedom can not be controlled independently, the task space can be divided into two orthogonal subspaces, namely, the position controlled subspace and the force-controlled subspace.

A correction to the position formulation of the hybrid position/force control scheme was needed because of mistake in the equation. In the original position formulation, the derivation that uses the inverse of the Jacobian matrix to map from Cartesian space to joint space was shown to be algebraically incorrect by [28-30]. This incorrect derivation causes the kinematic instability attributed to the hybrid position/force control scheme. With the correct position formulation, the hybrid position/force control scheme demonstrates kinematically stable behavior and was applied to an example.

A block diagram of hybrid control is shown in Figure II.5. and, the control law of this control system is given below.

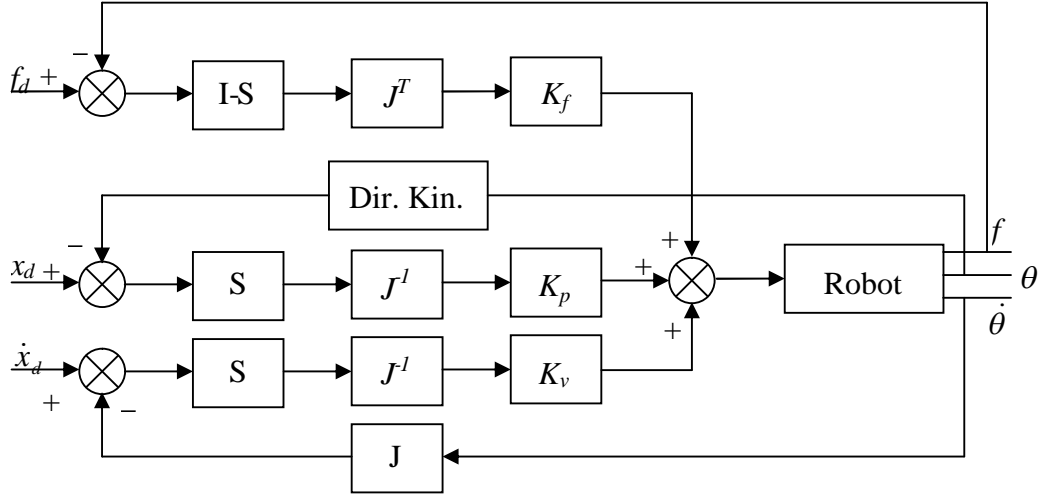


Figure II.5. Hybrid position/force control

$$\tau = K_p J^{-1} S (x_d - x) + K_v J^{-1} S (\dot{x}_d - \dot{x}) + K_f J^T (I - S) (f_d - f) \quad (II.15)$$

As the name implies, positions are controlled to track the desired trajectories along the other. A selection matrix, S , is utilized to decide which degrees of freedom are positions controlled and which degrees of freedom are forces controlled. In the original version of the hybrid control, the dynamics of the manipulator is not taken into account. Since then, the importance of the dynamic model in the hybrid control has been recognized. Shin and Lee, Khatib and Yoshikawa have presented the dynamic hybrid control by the consideration of the manipulator dynamics based on the computed torque method, the widely used control scheme for position control. Besides, Yabuta and Hollberbach have discussed the stability issues related to hybrid control. It has been shown by experiments that using the dynamic model of manipulators leads to more accurate and stable force control [12].

Hybrid control on the basis of formulation is highly intuitive approach to compliant motion. Recently, the equivalence between the hybrid control and the constrained motion control is recognized. Constrained motion refers to the motion of manipulators, during which the manipulator comes into contact with a rigid environment. The basic theory for the dynamics of constrained mechanical systems is well established. Nevertheless, it is only recently that the significance of the dynamics of constrained system for the model and control of compliant motion tasks has been recognized. In Kankaanranta and Koivo, a method for reducing the dimension of the model of constrained motion is presented, and a control architecture

which leads to exact decoupling of position and force controlled directions is suggested. Similar studies are given in McClamroch and Wang, and the recent work of Mills. These studies give hybrid position/force control a theoretical basis. From the constrained motion perspective, the global asymptotic stability of the hybrid control can be easily established. This is an improvement over the prior intuitive and heuristic formulation [12].

Within the framework of the hybrid control, a prerequisite to the successful execution of a compliant motion task is the availability of the data that describe the task geometry or constraints imposed on the end-effector. Defining the task geometry is a very complex work and requires complex programming, even when the task geometry is known. Nonetheless, it is desirable to define the task geometry automatically even though the task geometry is unknown. During recent years, there have been several studies related to the automatic estimation of the task geometry. Blauer and Belanger have proposed a method to estimate some unknown parameters of the task geometry by using the extended Kalman filter. Merlet, Kazanzides, Yoshikawa and Sudou have suggested a similar approach to determine the normal and tangent vectors of the task geometry by the use of the force measurements end-effector velocities [12].

In order to achieve robust compliant motion control, Seraji has presented an adaptive hybrid control scheme [33], which is robust for unpredictable changes in the manipulator parameters, since in the presented adaptive scheme the dynamic model and parameter values of the manipulator are not required. Chen and Pandey have proposed a robust control scheme for hybrid control to cope with the bounded uncertainties of the manipulator parameters. Likewise, Kuo and Wang have given a similar approach, called nonlinear robust hybrid control. In impedance control, by introducing the reference trajectory of target impedance, which characterizes the desired dynamic relations of the end-effector with the environment, Lu and Meng have shown that the adaptive mechanism of Slotine and Li for motion control can be applied to impedance control, and they have also demonstrated that performance degradation due to parameter uncertainties in an impedance controlled robot system can be prevented by utilizing the proposed adaptive impedance control algorithm. Kazerooni have also considered the robust impedance control in the presence of parameter uncertainties and high frequency unmodelled dynamics by using a frequency domain design method [12].

Research on hybrid position/force control continues by many researchers and more of the results were printed. This control type applied to dual-arm robot manipulators that interact each other [34, 35] and to robust control of industrial robots [36], for redundant manipulators extended hybrid position/force control developed [9, 12, 44]. [37] Studied on the stable force controller design based on frequency response identification.

After several new researches hybrid position/force control based new force control types have been developed. These are fuzzy rule based position/force control [38, 39], adaptive hybrid position/force controls [39-43] and extended hybrid position/force controls [12, 44], etc.

II.2.3 Impedance Control

As opposed to hybrid control, impedance control does not try to track position and force trajectories directly, but rather to regulate the dynamic relationship between the contact forces and manipulator positions, namely the mechanical impedance. Impedance control focused on the design of a robot's dynamic behavior (mechanical impedance) as seen from the environment. In this control strategy, no hardware or software, switch is needed in the robot's control system when the robot travels from the free motion space to the constrained space. The force feedback loop closes naturally as soon as the robot interacts with the environment, which changes the robot's impedance as seen from the environment [7, 8, 10, 45, 46].

By controlling the manipulator positions, and regulating their relationship to the contact forces, the manipulator can be controlled to maintain appropriate contact forces thereby accomplishing the compliant motion task. As mentioned before, impedance control is introduced by Hogan as an extension of the stiffness and damping control. Impedance control has an advantage; this control scheme is suitable for both motion control and compliant motion control and compliant motion control without mode switching or changing control parameters [12].

The choice of the appropriate impedance can be based on extremely sparse information about the actual environment.

1. Almost no information about the environment (the four-bar linkage) need be known to pick the manipulator impedance.

2. Commands need not be accurate. If the commanded position (the lowest point in the internal energy field) could not be reached because of the constraint, the system simply finds the point of closest approach and stops.

3. Very little supervision is required.

4. The required instrumentation is extremely simple.

An impedance-controlled manipulator is multi-mode, parameter adaptive device. Its mechanical behavior can be adjusted to suit the immediate task requirements. As a result, the more conventional state control in which the impedance has been fixed. Thus Impedance Control is quite general [18].

The dynamic interaction between manipulator and environment may then be modulated, regulated, and controlled by that impedance. Thus, impedance control is especially beneficial in contact tasks, co-operating robots, and assembly tasks [7,18].

Seen from the environment along any degree of freedom, physical systems come in only two types; admittances, which accept effort (e.g., force) inputs and yield flow (e.g., motion) outputs; and impedances, which accept flow inputs and yield effort outputs.

The most important consequence of dynamic interaction between two physical systems is that one must physically complement the other: Along any degree of freedom, if one is impedance, the other must be admittance and vice versa. While a constrained inertial object can always be pushed on, it cannot always be moved; these systems are properly described as admittances. Seen from the manipulator, the world is admittance [7].

When a manipulator is mechanically coupled to its environment, to ensure physical compatibility with the environment admittance, the manipulator should assume the behavior of admittance. Because the mechanical interaction with the environment will change with different tasks, or even in the course of a single task-the manipulator may be coupled to the environment in one phase and decoupled from it in another-the behavior of the manipulator should be adaptable. Thus, the controller should be capable of modulating the impedance of the manipulator as appropriate for a particular phase of a task [7].

Impedance method, on the other hand, realizes the desired mechanical impedance by driving joint actuators using feedback control based on measurements of end-effector position, velocity, contact force, and so on [47]. The desired

impedance of the end-effector expressed by the equation and represented as block diagram in Figure II.6.

$$M_d(\ddot{x} - \ddot{x}_d) + B_d(\dot{x} - \dot{x}_d) + K_d(x - x_d) = -f_e \quad (II.16)$$

where M_d , B_d and K_d representing the desired inertia, damping and stiffness matrices of the robot manipulator [58].

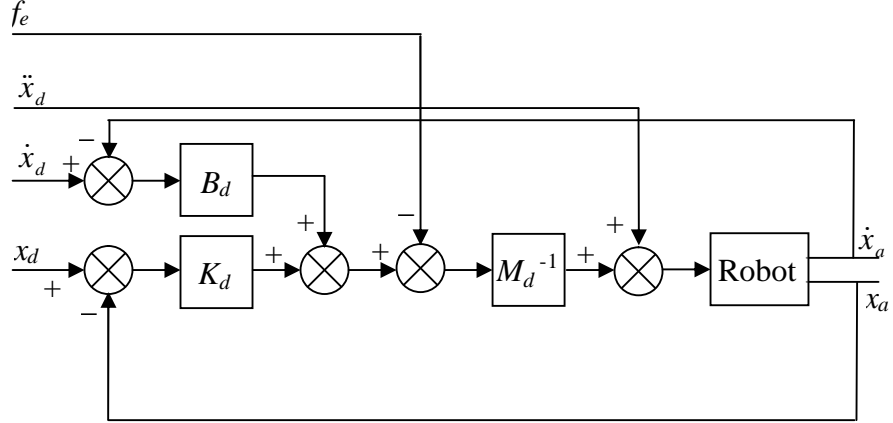


Figure II.6. Impedance control block diagram for (II.16) [58]

An important issue in robot action planning and control is to design a controller to achieve a stable phase transition and output force regulation with minimum impact forces and bouncing. Impedance control provides a stable and unified control structure for the phase transition. The problem with the impedance control is that, after contact has been established, the output force cannot be regulated unless the exact environmental model was known and was integrated into the motion plan. In contrast to impedance control, several discontinuous control schemes have been proposed. The major result is that the output force can be regulated after impact. But this may not be true if bouncing occurs during the impact. On the other hand, most of the existing phase transition control design depends on the parameters of the environment such as the stiffness and the surface location. Sometimes these requirements are not realistic for different material handling. In addition, the difficulty of the phase transition control is largely due to the existence of impact between a robot and the environment [48].

However, despite the important benefits of impedance control to robotic application this strategy is not widely used. That is mainly due to difficulties in controlling the desired impedance. These difficulties may be attributed to lack of accurate dynamic models and deficiencies in the proposed compensation mechanisms [18].

However, it introduces impedance errors whenever the actual position of the robot differs from that of the model [18].

Mostly, impedance controllers were implemented in a computed torque way, which means that an accurate model of the robot is needed. Hence, the controller is likely to be very sensitive to model uncertainties. Perhaps this is the reason that in industry impedance control is not yet widely used [47].

To overcome these problems new impedance control algorithms are developed and have been investigated. There are many force control types that have been presented.

II.2.3.1. Instantaneous Mode Impedance Control

The proposed instantaneous impedance control is designed to overcome the sensitivity of the dynamic based method to model uncertainties, and to improve the impedance tracking of the position based method. Two generic strategies for implementing impedance control have been described in the literature: the dynamic based approach [7] and the position based approach [15]. The second method has been introduced to eliminate the inherent sensitivity of the first method to the accuracy of the dynamical model. However, it introduces impedance errors whenever the actual position of the robot differs from that of the model. A proposed a new method for implementing impedance control, which is designed to take advantage of the error-correction capabilities of position controllers, while maintaining good impedance tracking performance by [18]. Instantaneous model impedance control reinitializes the impedance model to the current position of the robot so the model does not accumulate position errors. Its block diagram is given in Figure II.7.

II.2.3.2. Adaptive and Robust Impedance Control

Parameter mismatching always exists in practice and further adjustments of some parameters would be required in case of variable payloads. In such cases, it is possible to resort to an adaptation mechanism, which provides an on-line update of

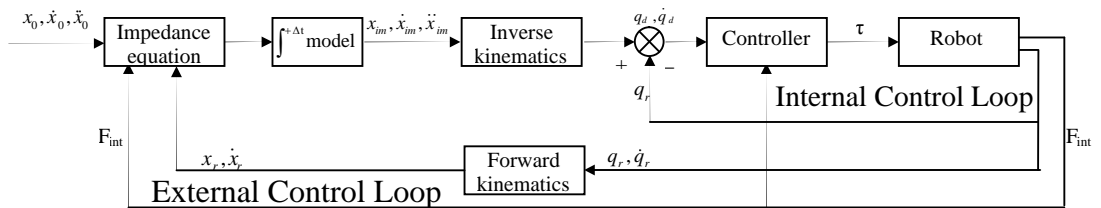


Figure II.7. Block diagram of instantaneous model impedance

the estimates of the dynamic parameters as a function of the relevant errors. To this purpose, adaptive versions of the force and position control schemes are developed [23, 24, 49-53].

II.2.3.3. Sliding Mode Impedance Control

Sliding mode control is distinct for its robustness property. When the system is in the sliding mode, its trajectories are reaching and are maintained on the sliding surface. The dynamics of the system, which is determined by the sliding surface, is unchanged under disturbances and model uncertainties. The objective is to combine the properties of sliding mode control and impedance control to make a robust impedance controller. Sliding mode impedance control consists of two parts, a nominal part and a robustness part. The nominal part is based on a nominal model of the robot arm that partially compensates for the nonlinearities of the robot. The robustness part compensates for the effect of the difference between the nominal model and the true robot on impedance implementation. Based on the objective of impedance control, an impedance error is defined as target impedance – apparent impedance. Thus, when the impedance error is zero, the apparent impedance is equal to the target one. To achieve zero error a sliding surface is defined as such that when the system is in the sliding mode the state trajectory remains on, or very close to, the sliding surface. The controller is formulated to guarantee the existence of reaching and sliding mode in the presence of model uncertainties. The robustness part can be switched, saturated proportional, or saturated proportional and integral differing in chattering steady-state error and complexity [2, 54, 55].

Object impedance control for flexible objects: Object impedance control was developed as an object-based control policy for multiple robotic arms manipulating a rigid object. One of the benefits of flexible- object impedance control is the ability to perform both free-space motions and contact tasks without switching control modes. The stability of the flexible-object impedance controller when coupled to an arbitrary passive environment is examined in [47, 56]. Dual robot operations [51, 52, 65], which are used for handling an object or objects.

II.2.3.4. Hybrid Impedance Control

The hybrid impedance control scheme presented in this paper combines impedance control and an I-type force feedback into one strategy. Used for handling flexible beam. The hybrid impedance controller uses an inner control loop for

manipulator dynamics linearization and decoupling. Impedance control substitutes for the position control used on hybrid controllers position subspace. This way, a desired manipulator dynamic is imposed [57-58].

II.2.3.5. Extended Impedance Control

Two control schemes, called extended hybrid control and extended impedance control respectively are developed for assuring the compliant motion of redundant manipulators. A method of compliant and collision avoidance control is presented. The compliant control method, which is compatible with present computer's facility, for redundant manipulators is proposed. As a feasible algorithm for the implementation of compliant motion and obstacle avoidance of redundant manipulators, we proposed an impedance control method that reduces the delay time in sensing and control circuit [12, 41, 59-61].

II.2.3.6. Spatial Impedance Control

Spatial parameters, e.g., principal directions of translational stiffness, have well-defined, intuitive, spatial transformation properties and can be selected depending on the configuration of objects with which the robot must interact. Spatial compliance control is a Euclidean-geometrical version of compliance control that is parameterized in an intuitive way [1,61-63].

II.2.3.7. Neural Networks Control Based Impedance Control

The purpose of the [53,65] is to improve the controller robustness by applying the neural network technique to compensate for the uncertainties in the robot model.

The impedance control is used to control haptic devices. A haptic, or force-reflecting, interface is a device that can sense a human operator's hand motion, and actuate the operator with forces or torques, without significantly impeding the hand motion. These devices have been used for teleoperation of a remote robot, the characterization of human motor performance, conducting psychophysical experiments on haptics, and the interaction of a human with a virtual environment [66, 67].

Variable impedance control in [68], Enhanced impedance control for spring-like impedance modulation in human arm models were presented in [69]. Impedance modulation in Human Arm Models in [70], Impedance control of generalized contact force and position. In [71]. Several methods for controlling the transition from free motion to constrained motion, with an emphasis on minimizing fingertip load

oscillations during the transition were examined in [71]. [14] Study on selection of the robot's target impedance. Six-DOF impedance control based on angle/axis represented in [72].

Dual robot operations [51, 52, 65], which are used for handling an object or objects.

II.2.4 Resolved Acceleration Force Control

Resolved acceleration force control is a simple extension to resolved acceleration position control by adding a force loop

Position control part

$$\ddot{x}^* = \ddot{x}_d + K_p(x_d - x) + K_v(\dot{x}_d - \dot{x}) \quad (II.16)$$

$$\dot{\theta}^* = J^{-1}(\ddot{x}^* - \dot{J}\dot{\theta}) \quad (II.17)$$

$$\tau = \hat{R}^{-1}(\theta, \dot{\theta}, \ddot{\theta}^*) \quad (II.18)$$

where \hat{R}^{-1} represents the forward dynamics equation of the robot manipulator

Force control part

$$\tau_f = J^T(I-S)K_f(f_d - f) \quad (II.19)$$

Resolved acceleration part

$$\tau = \hat{R}^{-1}(\theta, \dot{\theta}, \ddot{\theta}^*) + \tau_f \quad (II.20)$$

$$\tau = \hat{R}^{-1}(\theta, \dot{\theta}, \ddot{\theta}^*) + J^T(I-S)K_f(f_d - f) \quad (II.21)$$

A force error $f_d - f$ is multiplied first by a force gain K_f and then by a selection matrix I-S to ensure the correct partitioning between position variables and force variables. The transformation to joint torques is then accomplished with the

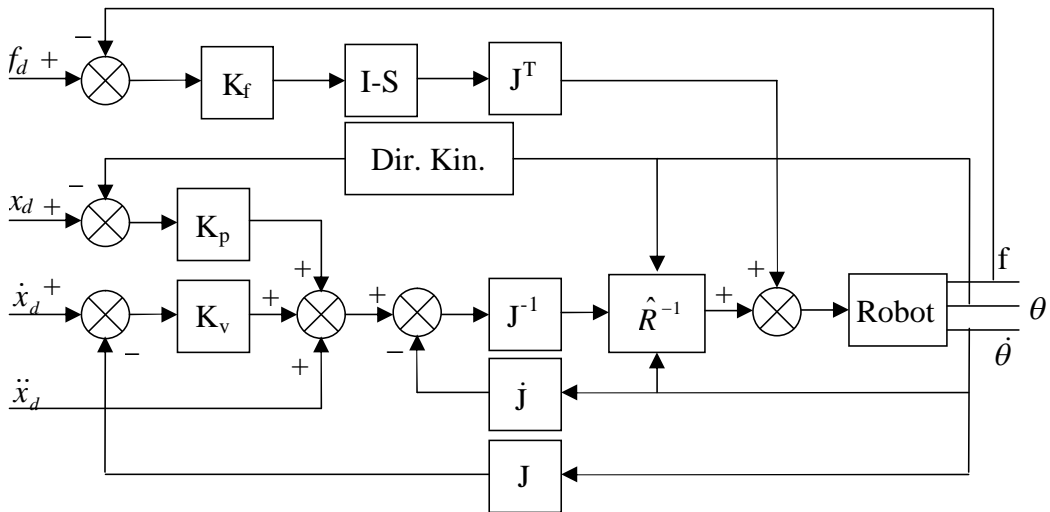


Figure II.8. Resolved acceleration force control

constrained task space directions. The control input is chosen as the sum of a position control action and a force control action [76]. The driving torques is calculated using eq. (II.22). The block diagram of this type of control is given in the Figure II.9.

$$\tau = J^T(\theta) \left(k_p(x_d - x) + \dot{f}_d + k_f(f_d - f) + k_i \int_0^t (f_d - f) \partial t \right) - k_v \dot{\theta} + g(\theta) \quad (II.22)$$

PART III.

MODELING, SIMULATION AND CONTROL

III.1.MATHEMATICAL MODEL OF ROBOT ARMS

In this chapter, dynamic models of robotic manipulators will be derived, simulated under various conditions and force control of them will be performed.

III.1.1 Modeling of The 3 DOF Spherical Robotic Manipulator

The robot arm used in this chapter is shown in Figure III.1 in terms of its degree of freedoms. Its kinematic and dynamic equations will be derived below.

This anthropomorphic manipulator is capable of a waist rotation θ_1 about axis z_1 , a shoulder rotation θ_2 about axis z_2 , an elbow θ_3 about axis z_3 , as shown in the Figure III.1. The tool origin position depicted in the figure might correspond to an immovable tool tip, such as that associated with a welding gun or paint sprayer, with the tool itself pointing in the approach direction. Other lengths are indicated on the figure.

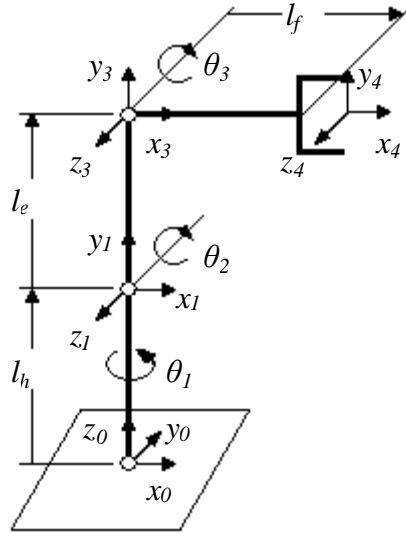


Figure III.1 3 DOF revolute type of robot arm

The appropriate D-H coordinate transformations in order to obtain the forward kinematic equations associated with the first 3 DOF of the robot arm are given in Table III.1.

Table III.1. Robot arm D-H representation

i_i	α_i	a_i	θ_i	d_i
1	0	0	θ_1	l_h
2	90	0	θ_2	0
3	0	l_e	θ_3	0
4	0	l_f	0	0

Direct substitution of the individual row elements into eq. (III.1)

$$T_{i-1}^i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_i \\ C\alpha_i S\theta_i & C\alpha_i C\theta_i & -S\alpha_i & -d_i S\alpha_i \\ S\alpha_i S\theta_i & S\alpha_i C\theta_i & C\alpha_i & d_i C\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.1)$$

The transformation matrices are given in eq. (III.2-5).

$$T_0^1 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & l_h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.2)$$

$$T_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.3)$$

$$\mathbf{T}_2^3 = \begin{bmatrix} C_3 & -S_3 & 0 & l_e \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.4)$$

$$\mathbf{T}_3^4 = \begin{bmatrix} 1 & 0 & 0 & l_f \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.5)$$

From these the orientation coordinate transformation of each link with respect to base are obtained in eq. (III.6-III.8)

$$\mathbf{T}_0^2 = \begin{bmatrix} C_1 C_2 & -C_1 S_2 & S_1 & 0 \\ S_1 C_2 & -S_1 S_2 & -C_1 & 0 \\ S_2 & C_2 & 0 & l_h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.6)$$

$$\mathbf{T}_0^3 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1 C_2 l_e \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1 C_2 l_e \\ S_{23} & C_{23} & 0 & S_2 l_e + l_h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.7)$$

$$\mathbf{T}_0^4 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1 C_2 l_e + C_1 C_{23} l_f \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1 C_2 l_e + S_1 C_{23} l_f \\ S_{23} & C_{23} & 0 & S_{23} l_f + S_2 l_e + l_h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.8)$$

Inertia matrices for dynamic equations of motion for waist, shoulder and elbow are given in eqns. (III.9-11). It should be noted that the waist is defined as a cylinder, shoulder as a cone and elbow as a cylindrical rod.

$$\mathbf{I}_1 = \begin{bmatrix} \frac{1}{2} m_1 r_1^2 + \frac{1}{3} m_1 l_h^2 & 0 & 0 \\ 0 & \frac{1}{2} m_1 r_1^2 + \frac{1}{3} m_1 l_h^2 & 0 \\ 0 & 0 & m_1 r_1^2 \end{bmatrix} \quad (III.9)$$

$$I_2 = \begin{bmatrix} \frac{1}{2}m_2r_2^2 & 0 & 0 \\ 0 & \frac{1}{4}m_2r_2^2 + \frac{1}{6}m_2l_e^2 & 0 \\ 0 & 0 & \frac{1}{4}m_2r_2^2 + \frac{1}{6}m_2l_e^2 \end{bmatrix} \quad (III.10)$$

$$I_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{2}m_3l_f^2 & 0 \\ 0 & 0 & \frac{1}{2}m_3l_f^2 \end{bmatrix} \quad (III.11)$$

Kinetic energy equations for the links are given in eqns. (III.12-14).

$$K_1 = \frac{1}{2}\dot{\theta}_1^T I_1 \dot{\theta}_1 \quad (III.12)$$

$$K_2 = \frac{1}{2}\dot{\theta}_2^T I_2 \dot{\theta}_2 \quad (III.13)$$

$$K_3 = \frac{1}{2}m_3v_3^T v_3 + \frac{1}{2}\dot{\theta}_3^T I_3 \dot{\theta}_3 \quad (III.14)$$

Angular velocity of the link 1 (waist) with respect to the base frame is given in eq. (III.15), link 2 (shoulder) in eq. (III.16) and link 3 (elbow) in eq. (III.17).

$$\dot{\theta}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (III.15)$$

$$\dot{\theta}_2 = \begin{bmatrix} S_2 \dot{\theta}_1 \\ C_2 \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \quad (III.16)$$

$$\dot{\theta}_3 = \begin{bmatrix} S_{23} \dot{\theta}_1 \\ C_{23} \dot{\theta}_1 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} \quad (III.17)$$

Linear velocity of link 3 (elbow) with respect to the base frame is

$$v_3 = \begin{bmatrix} -\frac{l_f}{2}S_1C_{23}\dot{\theta}_1 - \frac{l_f}{2}C_1S_{23}(\dot{\theta}_2 + \dot{\theta}_3) - l_eS_1C_2\dot{\theta}_1 - l_eC_1S_2\dot{\theta}_2 \\ \frac{l_f}{2}C_1C_{23}\dot{\theta}_1 - \frac{l_f}{2}S_1S_{23}(\dot{\theta}_2 + \dot{\theta}_3) + l_eC_1C_2\dot{\theta}_1 - l_eS_1S_2\dot{\theta}_2 \\ \frac{l_f}{2}C_{23}(\dot{\theta}_2 + \dot{\theta}_3) + l_eC_2\dot{\theta}_2 \end{bmatrix} \quad (III.18)$$

Potential energies of link 1, link 2 and link 3 are

$$P_1 = 0 \quad (III.19)$$

$$P_2 = m_2 g \frac{l_e}{3} S_2 + m_3 g \left[l_e S_2 + \frac{l_f}{2} S_{23} \right] \quad (III.20)$$

$$P_3 = m_3 g \frac{l_f}{2} S_{23} \quad (III.21)$$

Total kinetic and potential energies of the links are obtained as

$$K = K_1 + K_2 + K_3 \quad (III.22)$$

$$P = P_1 + P_2 + P_3 \quad (III.23)$$

Defining the Lagrangian L as

$$L = K - P \quad (III.24)$$

We can get the equations of motion of the robot links without any external effects

$$\tau_1 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} \quad (III.25)$$

$$\begin{aligned} \tau_1 = & \left(m_1 r_1^2 + \frac{m_2 r_2^2}{2} - \left[\frac{m_2 r_2^2}{4} - \left(\frac{m_2}{6} + m_3 \right) l_e^2 \right] C_2^2 + \frac{m_3 l_f^2 C_{23}^2}{3} + m_3 l_e l_f C_2 C_{23} \right) \ddot{\theta}_1 \\ & + \left(\left(\frac{m_2 r_2^2}{2} - \left(\frac{m_2}{3} + 2m_3 \right) l_e^2 \right) S_2 C_2 - \frac{2m_3 l_f^2 S_{23} C_{23}}{3} - m_3 l_e l_f (C_2 S_{23} + S_2 C_{23}) \right) \dot{\theta}_1 \dot{\theta}_2 \\ & + \left(-m_3 l_e l_f C_2 S_{23} - \frac{2m_3 l_f^2 S_{23} C_{23}}{3} \right) \dot{\theta}_1 \dot{\theta}_3 \end{aligned} \quad (III.26)$$

$$\tau_2 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} \quad (III.27)$$

$$\begin{aligned} \tau_2 = & \left(\frac{m_2 r_2^2}{4} + \left(\frac{m_2}{6} + m_3 \right) l_e^2 + \frac{m_3 l_f^2}{3} + m_3 l_e l_f C_3 \right) \ddot{\theta}_2 + \left(\frac{m_3 l_f^2}{3} + \frac{m_3 l_e l_f C_3}{2} \right) \ddot{\theta}_3 \\ & + \left(\left(-m_2 \frac{r_2^2}{4} + \left(\frac{m_2}{6} + m_3 \right) l_e^2 \right) S_2 C_2 + \frac{m_3 l_f^2 S_{23} C_{23}}{3} + \frac{m_3 l_e l_f (C_2 S_{23} + S_2 C_{23})}{2} \right) \dot{\theta}_1^2 \\ & - \frac{m_3 l_e l_f S_3}{2} \dot{\theta}_3^2 - m_3 l_e l_f S_3 \dot{\theta}_2 \dot{\theta}_3 - g \left(\frac{m_2 l_e C_2}{3} + m_3 \left(l_e C_2 + \frac{l_f C_{23}}{2} \right) \right) \end{aligned} \quad (III.28)$$

$$\tau_3 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_3} \right) - \frac{\partial L}{\partial \theta_3} \quad (III.29)$$

$$\begin{aligned} \tau_3 = & \left(\frac{m_3 l_f^2}{3} + \frac{m_3 l_e l_f C_3}{2} \right) \ddot{\theta}_2 + \frac{m_3 l_f^2}{3} \ddot{\theta}_3 + \left(\frac{m_3 l_e l_f C_2 S_{23}}{2} + \frac{m_3 l_f^2 S_{23} C_{23}}{3} \right) \dot{\theta}_1^2 \\ & + \frac{m_3 l_e l_f S_3}{2} \dot{\theta}_2^2 + \frac{g m_3 l_f C_{23}}{2} \end{aligned} \quad (III.30)$$

These equations are rearranged to obtain the general form of the robot arm with external force applied to the end effector. The general form is given in the eq. (III.31)

$$M(\theta)\ddot{\theta} + B(\theta, \dot{\theta}) + G(\theta) - J(\theta)^T f_e = \tau \quad (III.31)$$

where

$M(\theta)$: Mass and inertia term matrix, (3x3)

$B(\theta, \dot{\theta})$: Coriolis and centrifugal term matrix, (3x3)

$G(\theta)$: Gravity term matrix, (3x3)

f_e : External force term matrix, (3x3)

τ : Input torque term matrix (3x3)

The matrix forms of these terms are,

$$M(\theta) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \quad (III.32)$$

where

$$M_{11} = m_1 r_1^2 + \frac{m_2 r_2^2}{2} - \left(\frac{m_2 r_2^2}{4} - \left(\frac{m_2}{6} + m_3 \right) l_e^2 \right) C_2^2 + \frac{m_3 l_f^2 C_{23}^2}{3} + m_3 l_e l_f C_2 C_{23} \quad (III.33)$$

$$M_{22} = \frac{m_2 r_2^2}{4} + \left(\frac{m_2}{6} + m_3 \right) l_e^2 + \frac{m_3 l_f^2}{3} + m_3 l_e l_f C_3 \quad (III.34)$$

$$M_{23} = M_{32} = \frac{m_3 l_f^2}{3} + \frac{m_3 l_e l_f C_3}{2} \quad (III.35)$$

$$M_{33} = \frac{m_3 l_f^2}{3} \quad (III.36)$$

$$M_{12} = M_{13} = M_{21} = M_{31} = 0 \quad (III.37)$$

$$B(\theta, \dot{\theta}) = \begin{bmatrix} C_{s11} & C_{s12} & C_{s13} \\ C_{s21} & C_{s22} & C_{s23} \\ C_{s31} & C_{s32} & C_{s33} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \end{bmatrix} + \begin{bmatrix} C_{p11} & C_{p12} & C_{p13} \\ C_{p21} & C_{p22} & C_{p23} \\ C_{p31} & C_{p32} & C_{p33} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \dot{\theta}_2 \\ \dot{\theta}_1 \dot{\theta}_3 \\ \dot{\theta}_2 \dot{\theta}_3 \end{bmatrix} \quad (III.38)$$

where

$$Cs_{11} = Cs_{12} = Cs_{13} = Cs_{22} = Cs_{33} = 0 \quad (III.39)$$

$$Cs_{21} = (-m_2 \frac{r_2^2}{4} + (\frac{m_2}{6} + m_3) l_e^2) S_2 C_2 + \frac{m_3 l_f^2 S_{23} C_{23}}{3} + \frac{m_3 l_e l_f S_{223}}{2} \quad (III.40)$$

$$Cs_{23} = -Cs_{32} = -\frac{m_3 l_e l_f S_3}{2} \quad (III.41)$$

$$Cs_{31} = \frac{m_3 l_e l_f C_2 S_{23}}{2} + \frac{m_3 l_f^2 S_{23} C_{23}}{3} \quad (III.42)$$

$$Cp_{11} = (\frac{m_2 r_2^2}{2} - (\frac{m_2}{3} + 2m_3) l_e^2) S_2 C_2 - \frac{2m_3 l_f^2 S_{23} C_{23}}{3} - m_3 l_e l_f S_{223} \quad (III.43)$$

$$Cp_{12} = -m_3 l_e l_f C_2 S_{23} - \frac{2m_3 l_f^2 S_{23} C_{23}}{3} \quad (III.44)$$

$$Cp_{23} = -m_3 l_e l_f S_3 \quad (III.45)$$

$$Cp_{13} = Cp_{21} = Cp_{22} = Cp_{31} = Cp_{32} = Cp_{33} = 0 \quad (III.46)$$

$$G(\theta) = \begin{bmatrix} 0 \\ g(\frac{m_2 l_e C_2}{3} + m_3 (l_e C_2 + \frac{l_f C_{23}}{2})) \\ \frac{gm_3 l_f C_{23}}{2} \end{bmatrix} \quad (III.47)$$

Jacobian matrix for the system is given below

$$J = \begin{bmatrix} -S_1 (l_e C_2 + l_f C_{23}) & -C_1 (l_e S_2 + l_f S_{23}) & -l_f C_1 S_{23} \\ C_1 (l_e C_2 + l_f C_{23}) & -S_1 (l_e S_2 + l_f S_{23}) & -l_f S_1 S_{23} \\ 0 & l_e C_2 + l_f C_{23} & l_f C_{23} \end{bmatrix} \quad (III.48)$$

Kinematic equations for the system is

$$P_x = l_e C_1 C_2 + l_f C_1 C_{23} \quad (III.49)$$

$$P_y = l_e S_1 C_2 + l_f S_1 C_{23} \quad (III.50)$$

$$P_z = l_h + l_e S_2 + l_f S_{23} \quad (III.51)$$

Inverse kinematics of the spherical model

θ_1 is dual valued and explicitly given by

$$\theta_1 = a \tan 2 \left[\frac{P_y}{P_x} \right] \text{ and} \quad (III.52)$$

$$\theta_1 = a \tan 2 \left[-\frac{P_y}{P_x} \right] \text{ or } \theta_1 = \pm 180^\circ \quad (III.53)$$

$$\theta_3 = a \tan 2 \left[\frac{\pm \sqrt{4le^2 lf^2 - [(P_z - lh)^2 + P_x^2 + P_y^2 - le^2 - lf^2]^2}}{(P_z - lh)^2 + P_x^2 + P_y^2 - le^2 - lf^2} \right] \quad (III.54)$$

i.e. both θ_3 and $-\theta_3$ solve eq. (III.54). It might be noted that this particular expression for θ_3 is dependent only on the known initial values associated with the problem namely, P_x, P_y, P_z, l_e, l_f and l_h . Therefore, θ_3 can be determined prior to and independent of θ_1 if desired, although it is not really necessary to do so: i.e. eq. (III.54) can also be directly used to determine θ_3 , via (B1), once θ_1 is determined.

Once θ_1 is determined, θ_2 can be obtained as

If θ_1 is given by eq. (III.52) then,

$$\theta_2 = a \tan 2 \left[\frac{(P_z - lh)(le + lfC_3) - lfS_3 \sqrt{P_x^2 + P_y^2}}{(P_z - lh)lfS_3 + (le + lfC_3) \sqrt{P_x^2 + P_y^2}} \right] \quad (III.55)$$

On the other hand if θ_1 is given by eq. (III.53) then,

$$\theta_2 = a \tan 2 \left[\frac{(P_z - lh)(le + lfC_3) + lfS_3 \sqrt{P_x^2 + P_y^2}}{(P_z - lh)lfS_3 - (le + lfC_3) \sqrt{P_x^2 + P_y^2}} \right] \quad (III.56)$$

Note that both of those expression for θ_2 are dual-valued because of θ_3 , as given by eq. (III.54), dual-valued.

These equations will be utilized in the next section for simulation studies.

III.1.2 Modeling of The 3 DOF Cartesian Robotic Manipulator

The second type of robot manipulator is 3DOF Cartesian type manipulator. The calculation of this manipulator is quite simple then the revolute type. All drivers are work independent each other and are not affect their motion like revolute type. The structure of this robot is drawn in the Figure III.2. l_1, l_2 and l_3 represent link length of the x, y and z axis respectively.

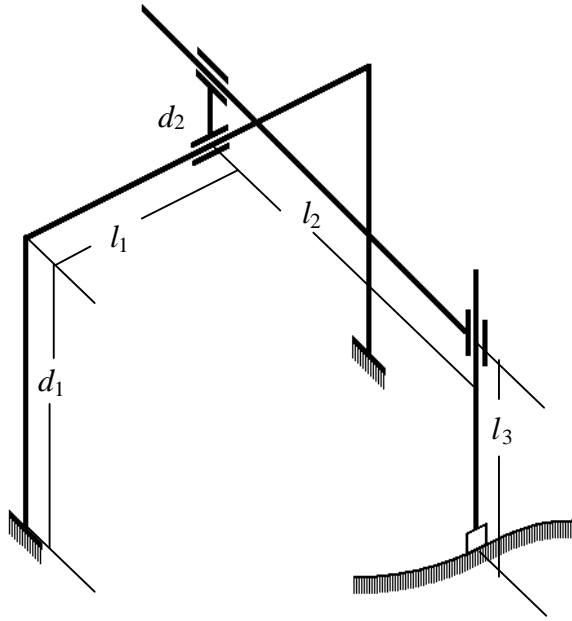


Figure III.2. 3 DOF Cartesian robot arm

Transformation matrices for this robot are given below

$$T_0^1 = \begin{bmatrix} 0 & 0 & 1 & l_1 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.57)$$

$$T_1^2 = \begin{bmatrix} 0 & -1 & 0 & d_2 \\ 0 & 0 & -1 & l_2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.58)$$

$$T_2^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.59)$$

$$T_3^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.60)$$

$$T_0^4 = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & l_2 \\ 0 & 0 & 1 & d_1 + d_2 - l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.61)$$

Kinetic energy equations for the links are given in eqns. (III.62-64).

$$K_1 = \frac{1}{2} m_1 v_1^2 \quad (III.62)$$

$$K_2 = \frac{1}{2} m_2 v_2^2 \quad (III.63)$$

$$K_3 = \frac{1}{2} m_3 v_3^2 \quad (III.64)$$

Potential energies of link 1, link 2 and link 3 are

$$P1=0 \quad (III.65)$$

$$P2=0 \quad (III.66)$$

$$P3=m_3g \quad (III.67)$$

The force equations of the link 1 to link3 are solved using energy equation. The results are given in eqns. (III.68-70)

$$F_1=(m_1+m_2+m_3)a_1 \quad (III.68)$$

$$F_2=(m_2+m_3)a_2 \quad (III.69)$$

$$F_3=m_3(a_3+g) \quad (III.70)$$

III.2.SIMULATION OF ROBOTIC MANIPULATORS

Above equations are modeled and solved using MATLAB SIMULINK. M files that are used for Simulink files are given in Appendix A and Appendix B. Simulink modeling diagrams are shown in Figure III.2. to Figure III.11. Simulink is a very versatile program to simulate dynamic systems [77]. It solves the resulting differential equations of the robot manipulator using functions of MATLAB , but it has a graphical interface.















The desired forces are set as $f_x=10$ N, $f_y=0$ N, $f_z=0$ N and the desired position is set $x=3$ m, $y=0$ m, $z=3$ m. Initial position before starting motion is for $x=2$ m, $y=0$ m, $z=3$ m. The contact position, x_c , of the end effector with the environment is equal to 2.8 m in x direction. The motion is restricted only through x direction. End effector move freely by the directions of y and z coordinates.

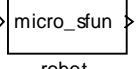
Desired force value is stored in the Fd block

III.2.1 Simulink Models of The Force Control Algorithms

Three kind force controls were tested in the two types of robots. The blocks that are used in the models are listed in the Table III.2. In the second raw of the table the properties of the blocks are given.

Table III.2. Block list and their properties that are used in simulink models

Blocks	Property
	Desired force in cartesian space.
	Desired position in cartesian space.
	Desired velocity in cartesian space.
	Position gain matrix.
	Velocity gain matrix.
	Force sensor model.
	Selection matrix for
	Forward dynamics of the cartsian type robot arm. Inputs are combined array of x, \dot{x}, \ddot{x} . Uses crtfdyn.m (Appendix A)
	Forward dynamics of the revolute type robot arm. Inputs are combined array of $\theta, \dot{\theta}, \ddot{\theta}$.
	Derivation of any input.
	Calculate the inverse, transpose and derivation of Jacobian matrix for revolute type of robot. Uses microjacobian_sfun.m. See Appendix B
	Direct kinematics of robot arms. Uses microdirkin_sfun.m See Appendix B
	Gravity compensator to eliminate the effect of the gravity. Uses crtgravity.m (Appendix A) for Cartesian and microgravity.m (Appendix B) for revolute robot
	Robot model of the cartesian type of the robot arm. Uses crt.m file see Appendix A

 >micro_sfuns< robot	Robot model of the revolute type of the robot arm. Uses micro_sfuns.m file. See Appendix B
---	--

Force sensor block represents the contact force in the models. Force depends on the stiffness of the environment and the deformation of the contact surface. It is simply formulated as

$$F_s = K_e * (x - x_c) \quad (III.71)$$

Contact position and environment stiffness are assumed 2.80 m (in x direction) and 100.000 N/m respectively. The force sensor is built using eq. (III.71) and is shown in Figure III.3.

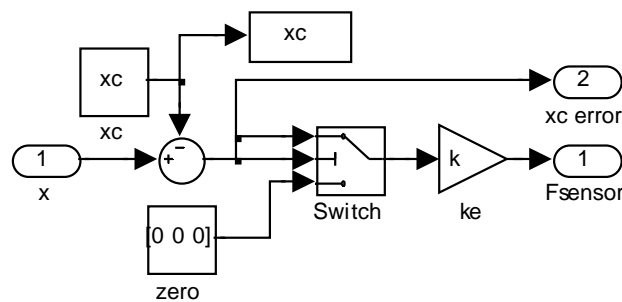


Figure III.3. Force sensor simulink block diagram

The selection matrix is used as a switch to activate or deactivate the force and the position control algorithms when the end effector contact with the environment. Only one of them are activated at the same by the selection matrix. The structure of this block is seen in the Figure III.4.

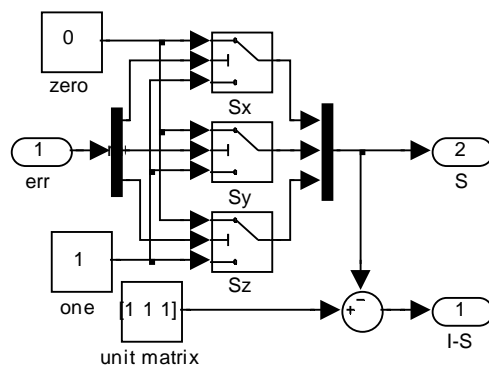


Figure III.4. Selection matrix block diagram

III.2.2 Force Control Models For Cartesian Type of Robot Arm

Simulink block diagrams of Hybrid, RAF and Parallel Force/Position controls are depend on the related control equations. The control equation of each model is referred before the block figures.

III.2.2.1. Hybrid Position/Force Control

The block diagram of hybrid control in simulink model, as it seen in the Figure III.5., is built by adapting eq. (II.15) from joint space to the Cartesian space. The model is similar to the Figure II.5.

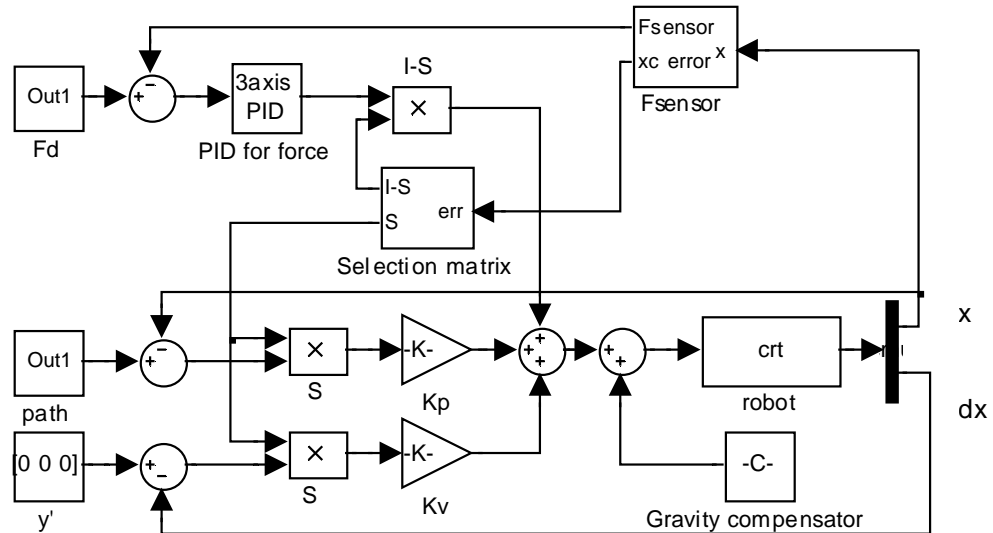


Figure III.5. Simulink block diagram of the hybrid control of cartesian type of robotic manipulator

III.2.2.2. Resolved Acceleration Force Control (RAFC)

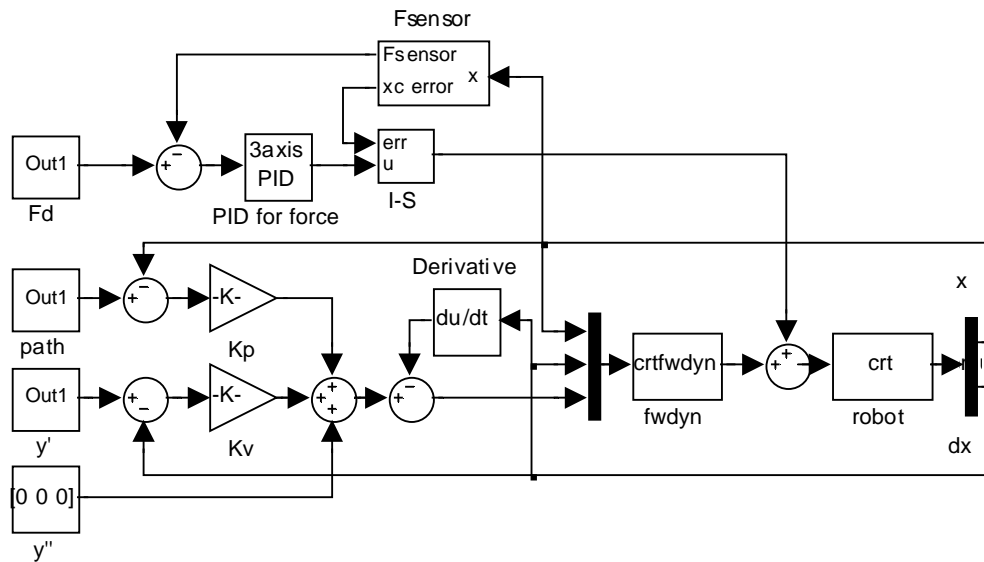


Figure III.6. Simulink block diagram of the RAFC of cartesian type of robotic manipulator

RAFC looks like hybrid control. However, selection matrix switch only force control part. When force control activated it added to position control as seen in the Figure III. 6. Acceleration obtained by derivation of the velocity.

The structure of the block of selection matrix is used in this control scheme is slightly different from of the hybrid control. This block enables or disables the force control section without affecting the position control as in hybrid control.

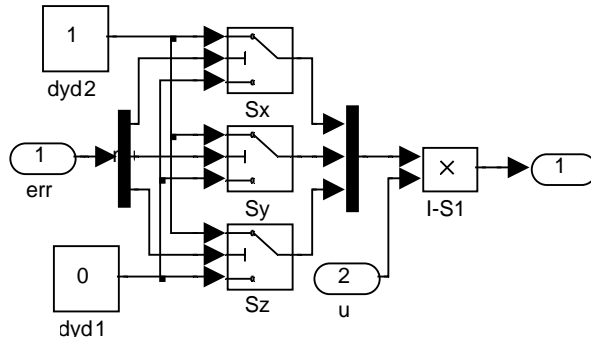


Figure III.7. The structure of the block of selection matrix of RAFC

III.2.2.3. Parallel Force/Position Control

This control is added force and position control scheme like RAFC. No need to switching to activate the force control section. Velocity gain is multiplied by velocity without error detection. The Simulink structure of this control scheme is in Figure III.8

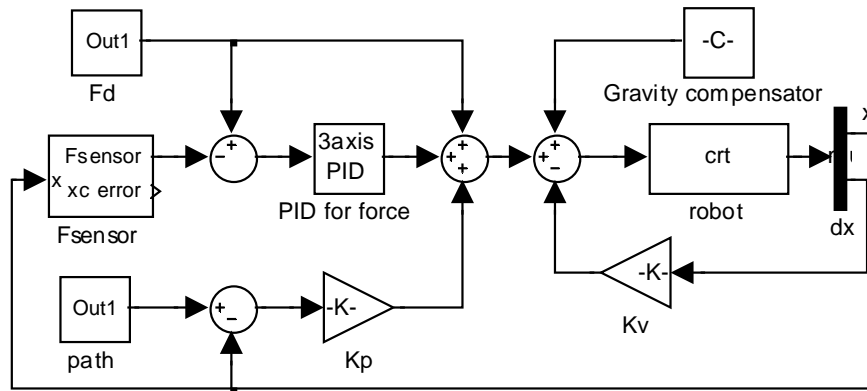


Figure III.8. Simulink block diagram of the parallel force/position control of cartesian type of robotic manipulator

III.2.3 Force control models for revolute type of robot arm

Force control model of the revolute type of robot arm is more complicated than Cartesian type. Link torques are dependent on all links motion and end effector propagated by each link. All links motion are rotational type and that's why Jacobian matrix must be used to transfer Cartesian space force and position values to joint space. Jacobian matrix calculations are made and are called from one function

that is called microjacobian.m. The calculations are transpose, inverse and derivation of the Jacobian matrix. In the models J' represents transpose, $\text{inv}(J)$ represents inverse and dJ represents derivation of Jacobian matrix.

III.2.3.1. Hybrid Position/Force Control

The Simulink control model of revolute type of robot little complex than the cartesian type. This complexity comes from Jacobian matrix. For position control and force sensing a conversion is needed from joint space to cartesian space. This conversion made by microdirkin_sfun.m matlab file. The model is shown in the Figure III.9.

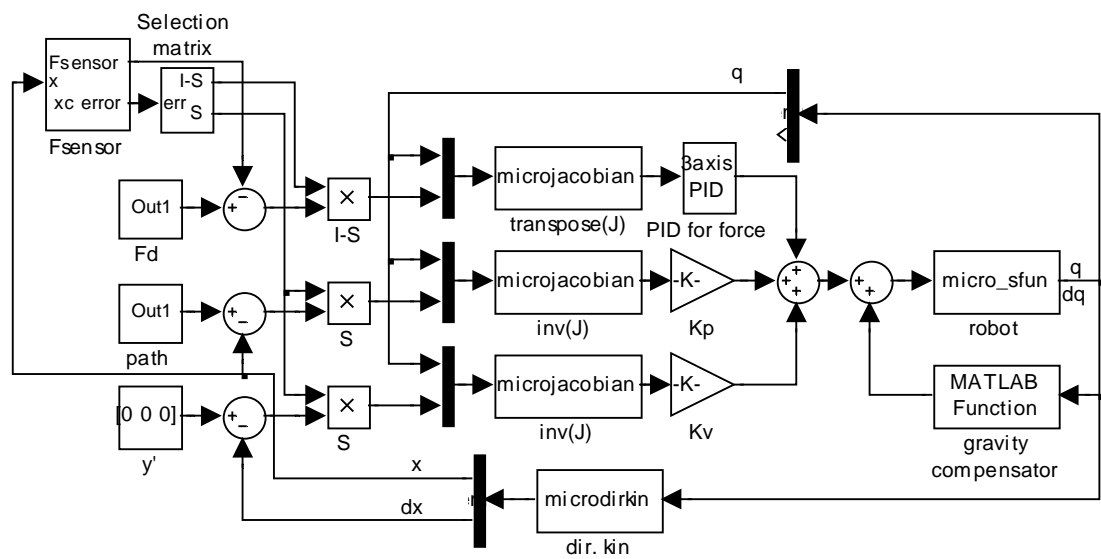


Figure III.9. Simulink block diagram of the hybrid control of revolute type of robotic manipulator

III.2.3.2. Resolved Acceleration Force Control (RAFC)

The Simulink model of RAFC is graphical representation of eq. (II.21) by block diagram. Figure III.10 shows the block diagram of the RAFC for revolute type of robot.

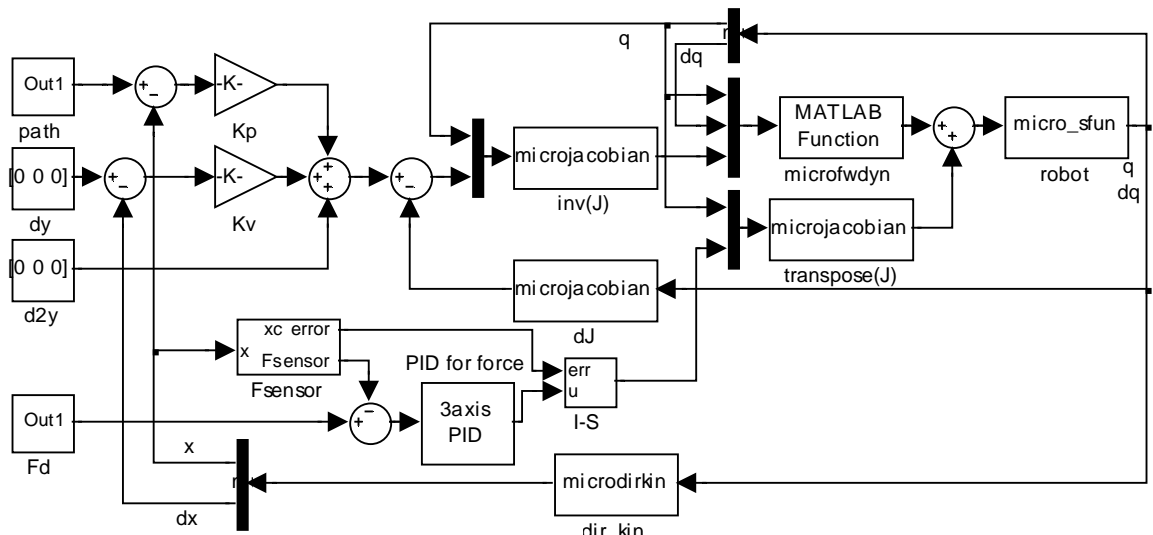


Figure III.10. Simulink block diagram of the RAFC of revolute type of robotic manipulator

III.2.3.3. Parallel Force/Position Control

This model looks like in the cartesian type of robot. Only transpose of the Jacobian matrix and direct kinematic blocks are added to the model. In the Figure III.11 can be seen that this control scheme is the simplest of these three models. This model depends on the eq. (II.22)

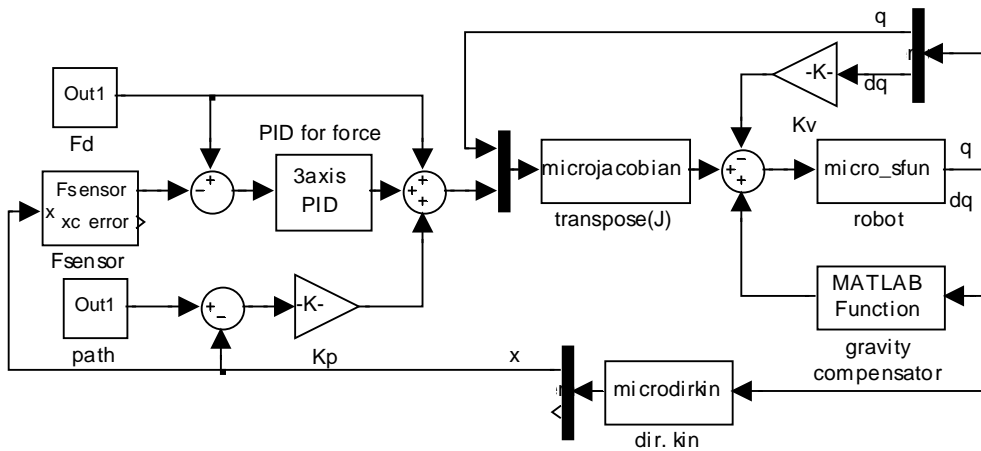


Figure III.11. Simulink block diagram of the parallel force/position control of revolute type of robotic manipulator

PART IV.

RESULTS ND EVALUATIONS

IV.1.STEP INPUT RESPONSE FOR POSITION CONTROL

For both robot arms same target and step input response has been set. Position and velocity gains, K_p and K_v , value are found under the predefined response behavior using MATLAB Parameter Optimization Toolbox. It is desired that the motion behavior of each robot at each control algorithm must be the same. That's why upper and lower limit of the position response is the same for each control schemes. Maximum overshoots are defined for x, y and z positions are $M_{xOx}=3.13$ m, $M_{xOy}=1.13$ m and $M_{xOz}= 4.18$ m respectively. Settling time and allowable tolerances, for all axes are the same, 1 sec. and %2 respectively. Initial position of end effector is $P_i(2,0,3)$ and the desired position after step input is $P_f(3,1,4)$. Position and velocity gains and step input responses are given related subsections from IV.1.1 to IV.1.3.

IV.1.1 Position and Velocity Gain of Hybrid Position/Force Control of Cartesian Type of Robot

The responses of the Hybrid Position/Force Control are in desired boundaries for x,y and z axes as seen in the Figure IV.1. The position and velocity gains are

$K_p = [1058.60 \ 623.20 \ 129.49]$, $K_v = [0193.88 \ 116.80 \ 27.03]$ respectively

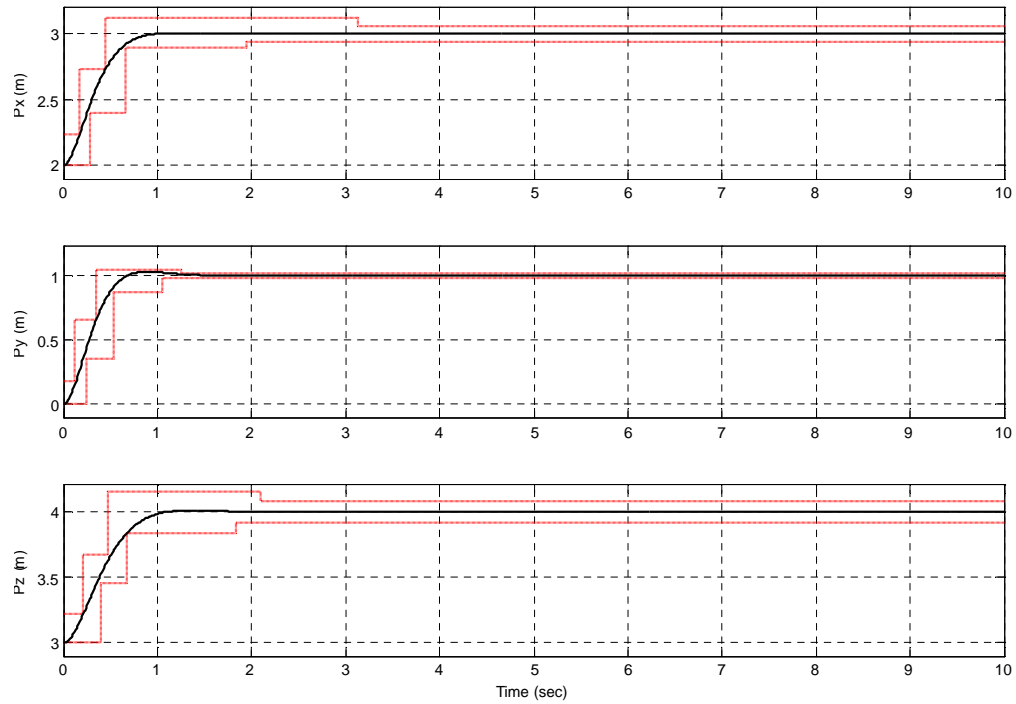


Figure IV.1 Step input response of position of hybrid control of cartesian type of robot

IV.1.2 Position and Velocity Gain of RAFC of Cartesian Type of Robot

The position and velocity gain values are small than the hybrid control. The gain values are $K_p = [144.21 \ 118.27 \ 066.46]$, $K_v = [027.82 \ 020.79 \ 013.56]$. Response graphs are in the limits Figure IV.2.

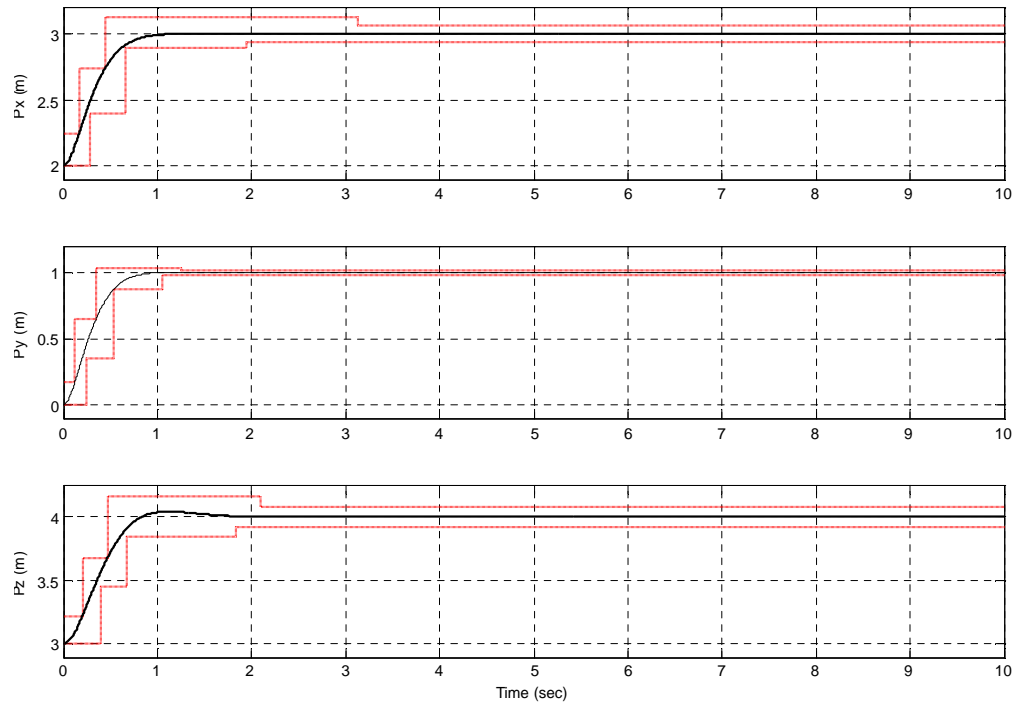


Figure IV.2 Step input response of position of RAFC of cartesian type of robot

IV.1.3 Position and Velocity Gain of Parallel Force/Position Control of Cartesian Type of Robot

Position and velocity gains are close to the values of the hybrid control. The response of all axes are in the desired range Figure IV.3. The gain values for this graph are, $K_p = [1029.80 \ 0511.46 \ 0132.31]$ and $K_v = [0190.11 \ 0097.49 \ 0027.76]$.

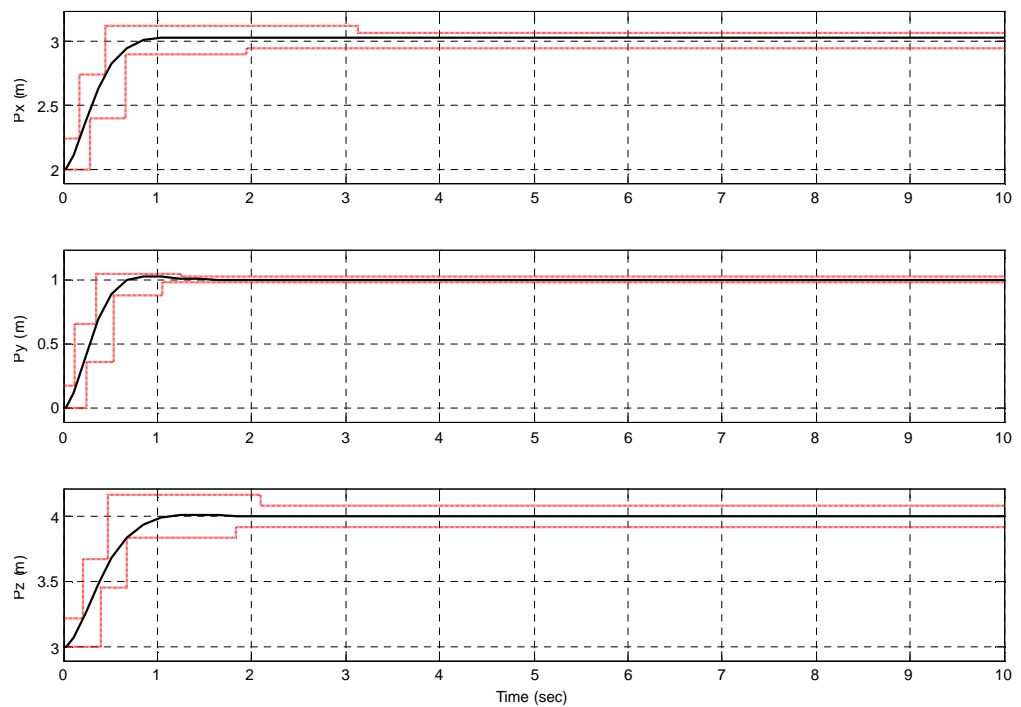


Figure IV.3 Step input response of position of parallel force/position control of cartesian type of robot

IV.1.4 Position and Velocity Gain of Hybrid Position/Force Control of Revolute Type of Robot

The response of control algorithm in y and z axes are in defined range. But in x axis overshoot is not in the range besides whole x axis graph Figure IV.4.. It is tried to bring the overshoot value in the range. But the response of the other axes are changed and overshoot values are in the defined range. The response in x is accepted as it seen in the Figure IV.4. The related gain values are

$$K_p = [544.23 \ 007.61 \ 068.59], K_v = [655.71 \ 000.00 \ 045.67]$$

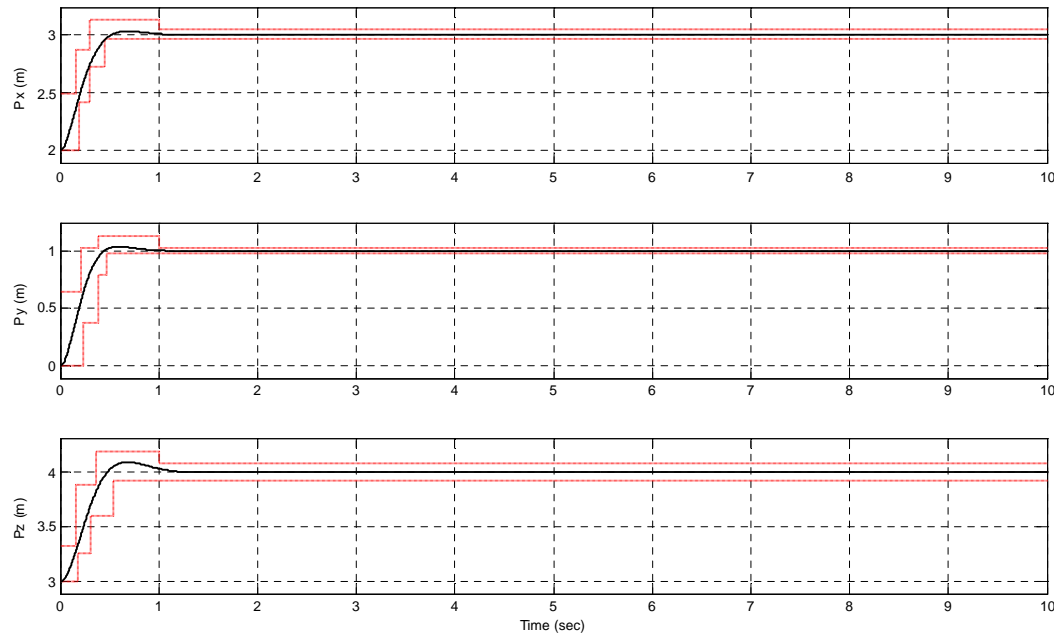


Figure IV.4 Step input response of position of hybrid control of revolute type of robot

IV.1.5 Position and Velocity Gain of RAFC of Revolute Type of Robot

The good graph of response of the revolute robot arm is gotten from this control model. The graphs of all axes are stayed in the desired limits Figure IV.5. Gain values for this response are

$$K_p = [104.75 \ 110.36 \ 063.43], K_v = [019.10 \ 022.30 \ 014.74]$$

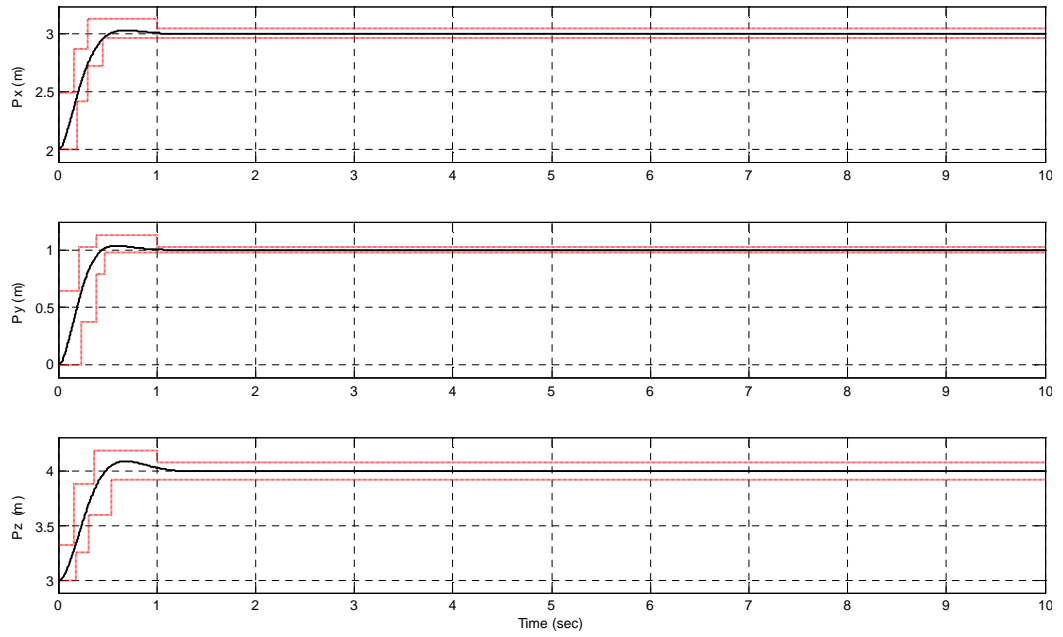


Figure IV.5 Step input response of position of RAFC of revolute type of robot

IV.1.6 Position and Velocity Gain of Parallel Force/Position Control of Revolute Type of Robot

The response of the control algorithm is acceptable. There is a small steady state error in the x -axis. This error is not taken in care of because of its small value.

The gain values that affect the result are

$$K_p = [077.80 \ 730.67 \ 017.80], \quad K_v = [999.77 \ 014.68 \ 054.60]$$

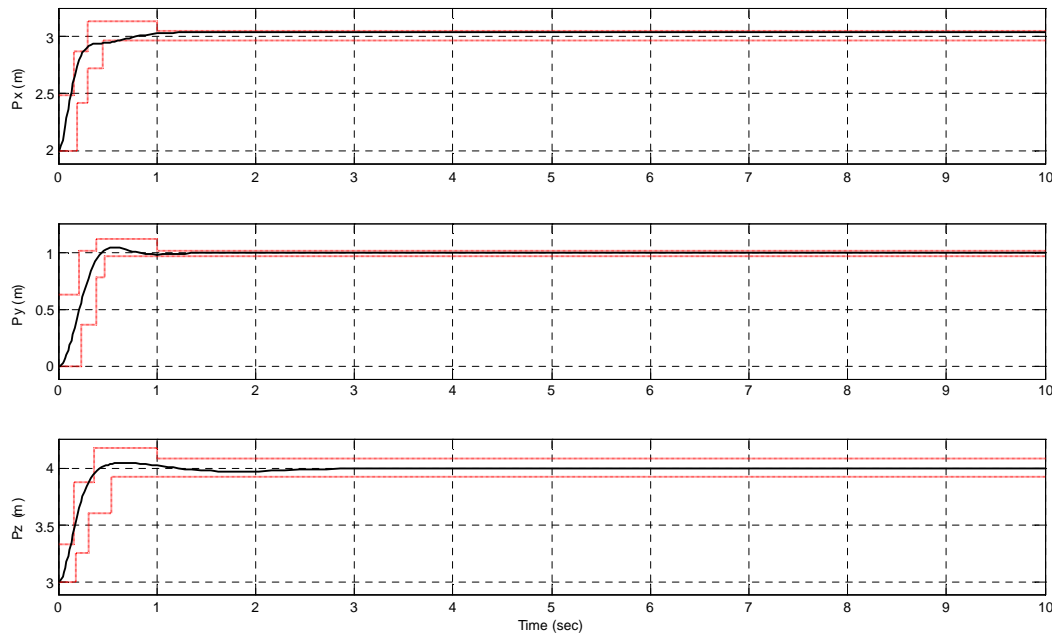


Figure IV.6 Step input response of position of parallel force/position control of revolute type of robot

IV.2. REDUCED TAGUCHI L9 (3⁴) TABLES

In this section, PID parameters of Force control algorithms are examined and compared using Taguchi arrays [78]. Taguchi method is suitable for reducing experiment number for large parameter with more than two levels. The response of force control algorithms are adjusted by PID parameters. There are three levels of PID parameters; that are the value found by Simulink Parameter Optimization Method itself, which is placed in the second row, %50 over and %50 under this value. The first row includes %50 small and the third row includes %50 big values of PID parameter. The table is organized using Taguchi L9 (3⁴) arrays by three level and three parameters. PID parameters in the nine rows are applied to force control algorithms respectively. Steady state error (Error), percentage difference of maximum overshoot (MxO) w.r.t. 10 N (desired force) and settling time (S.T.) are results that are selected for comparison. Computing time (C.T.) is used when results are near to each other for selection criteria.

Results are listed in the part (a) of the table. The values of the results are divided by their maximum value by column. The results of this division are listed in the part (b) of the table. At the part (c), the values are calculated using eq. (IV.1) and listed at column $Sqtr_{EMS}$. Results of eq. (IV.2) are also listed in the $Sqtr_{EMSC}$ column at the part (c).

$$Sqtr_{EMS} = \sqrt{Error^2 + MPR^2 + STR^2} \quad (IV.1)$$

$$Sqtr_{EMSC} = \sqrt{Error^2 + MPR^2 + STR^2 + CTR^2} \quad (IV.2)$$

Minimum of the $Sqtr_{EMS}$ is the best value of them. Maximum value $Sqtr_{EMS}$ is the worst of them. Graph of the best and the worst are given after tables.

IV.2.1 Taguchi L9 (3⁴) Table of Hybrid Position/Force Control of Cartesian Type of Robot Arm

Hybrid Position/Force Control type of force control includes two types of controls, position and force control, works independent from each other and only one of them works at the same axis. If robot arm tip moves freely position control works otherwise during restricted motion or contact to an environment force control part works. In the control equation as it seen force control part contain only proportional gain. Integration and derivation gain are added to the force control.

Table IV.1 Taguchi arrays of hybrid control of cartesian type of robot (a)

	P	I	D	Error	MxO	%MxO	S.T.	C.T.
1	0,85	0,25	0,83	0,00	38,03	280,30	11,50	52,66
2	1,70	0,50	1,65	0,00	19,20	92,00	9,50	103,10
3	2,55	0,75	2,48	0,00	13,35	33,50	5,00	153,10
4	0,85	0,50	2,48	0,00	13,30	33,00	16,20	153,80
5	1,70	0,75	0,83	0,00	37,50	275,00	7,00	51,78
6	2,55	0,25	1,65	0,00	19,30	93,00	7,00	103,90
7	0,85	0,75	1,65	0,00	19,20	92,00	15,00	104,50
8	1,70	0,25	2,48	0,00	13,30	33,00	7,50	152,50
9	2,55	0,50	0,83	0,00	37,50	275,00	9,60	52,72
Max=				1,00		280,30	16,20	153,80

Table IV.2 Normalizing Taguchi arrays of hybrid control of cartesian type of robot (b,c)

	Divided by maximum (b)				(c)	
	ER	MPR	STR	CTR	Sqrt _{EMSC}	Sqrt _{EMS}
1	0,000	1,0000	0,7099	0,3424	1,2732	1,2263
2	0,000	0,3282	0,5864	0,6704	0,9492	0,6720
3	0,000	0,1195	0,3086	0,9954	1,0490	0,3310
4	0,000	0,1177	1,0000	1,0000	1,4191	1,0069
5	0,000	0,9811	0,4321	0,3367	1,1237	1,0720
6	0,000	0,3318	0,4321	0,6756	0,8678	0,5448
7	0,000	0,3282	0,9259	0,6795	1,1945	0,9824
8	0,000	0,1177	0,4630	0,9915	1,1006	0,4777
9	0,000	0,9811	0,5926	0,3428	1,1963	1,1462

The minimum value is in the 3. row of the Sqrt_{EMS} column of the Table IV.2. Maximum values of PID parameters are cause the minimum overshoot and settling time values. The worst results, on the other hand, are found at the 1. row of the table with maximum value of Sqrt_{EMS}. PID parameters at the 1. row are minimum. Increasing PID values together decrease the the overshoot and settling time. But its wise affect is increasing C.T. The response of the both position and force of end effector for the best and worst values are seen at the Figure IV.7 and Figure IV.8 respectively.

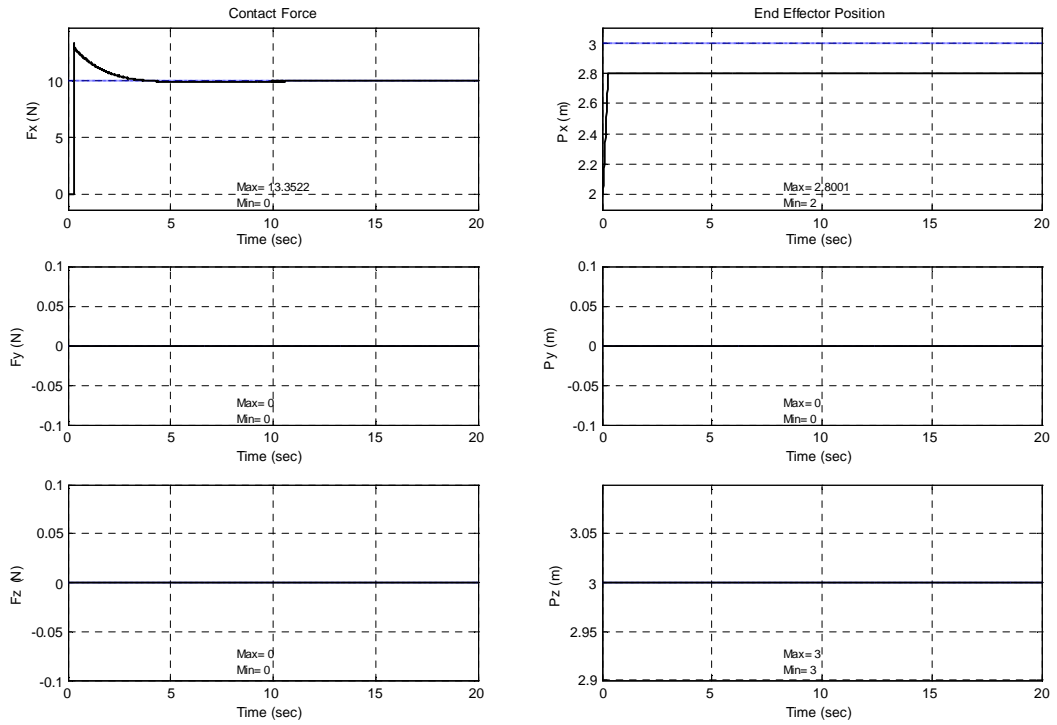


Figure IV.7 The best value of the force and position graph of Hybrid Position/Force Control of cartesian type of robot

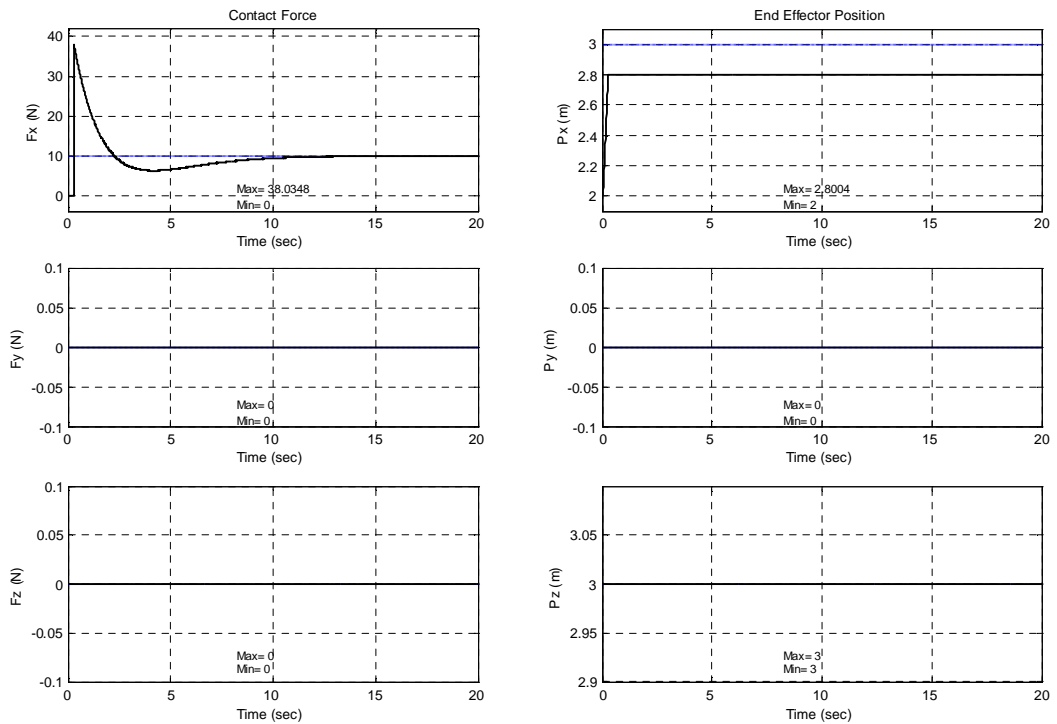


Figure IV.8 The worst value of the force and position graph of Hybrid Position/Force Control of cartesian type of robot

IV.2.2 Taguchi L9 (3⁴) Table of RAFC of Cartesian Type of Robot Arm

The best and the worst values for this type are in the 2. and 8. column of Table IV.1 and Table IV.2 respectively. The best value shows that mean PID value cause good value of MxO and ST as is seen in the Figure IV.9. At the worst value, at Figure IV.10 it can be seen that the contact force is not stable and its settling time is took long time. There is steady state error and can not reach the desired force.

Table IV.3 Taguchi arrays of RAFC of cartesian type of robot (a)

(a)	P	I	D	Error	MxO	%MxO	S.T.	C.T.
1	125,00	7,50	1,50	5,00	531,00	5210,00	100,00	169,10
2	250,00	15,00	3,00	0,00	21,50	115,00	25,00	343,30
3	375,00	22,50	4,50	5,00	54,00	440,00	100,00	364,60
4	125,00	15,00	4,50	5,00	351,50	3415,00	100,00	458,50
5	250,00	22,50	1,50	5,00	91,00	810,00	60,00	173,10
6	375,00	7,50	3,00	0,00	21,00	110,00	52,00	356,80
7	125,00	22,50	3,00	5,00	124,00	1140,00	100,00	298,50
8	250,00	7,50	4,50	10,00	302,00	2920,00	100,00	387,70
9	375,00	15,00	1,50	1,00	39,10	291,00	0,64	213,60
Max=				10,00		5210,00	100,00	458,50

Table IV.4 Normalizing Taguchi arrays of RAFC of Cartesian type of robot (b, c)

	Divided by maximum (b)				(c)	
	ER	MPR	STR	CTR	Sqrt _{EMSC}	Sqrt _{EMS}
1	0,5000	1,0000	1,0000	0,3688	1,5447	1,5000
2	0,0000	0,0221	0,2500	0,7487	0,7897	0,2510
3	0,5000	0,0845	1,0000	0,7952	1,3746	1,1212
4	0,5000	0,6555	1,0000	1,0000	1,6370	1,2960
5	0,5000	0,1555	0,6000	0,3775	0,8813	0,7963
6	0,0000	0,0211	0,5200	0,7782	0,9362	0,5204
7	0,5000	0,2188	1,0000	0,6510	1,3121	1,1392
8	1,0000	0,5605	1,0000	0,8456	1,7404	1,5212
9	0,1000	0,0559	0,0064	0,4659	0,4798	0,1147

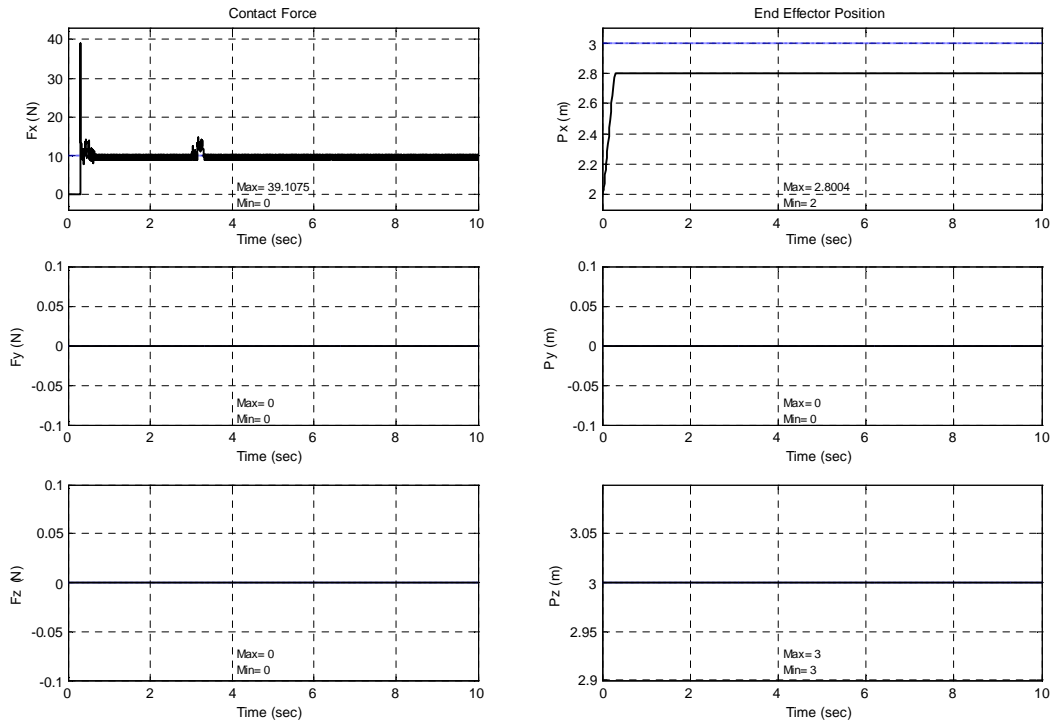


Figure IV.9 The best value of the Force and Position graph of RAFC of cartesian type of robot

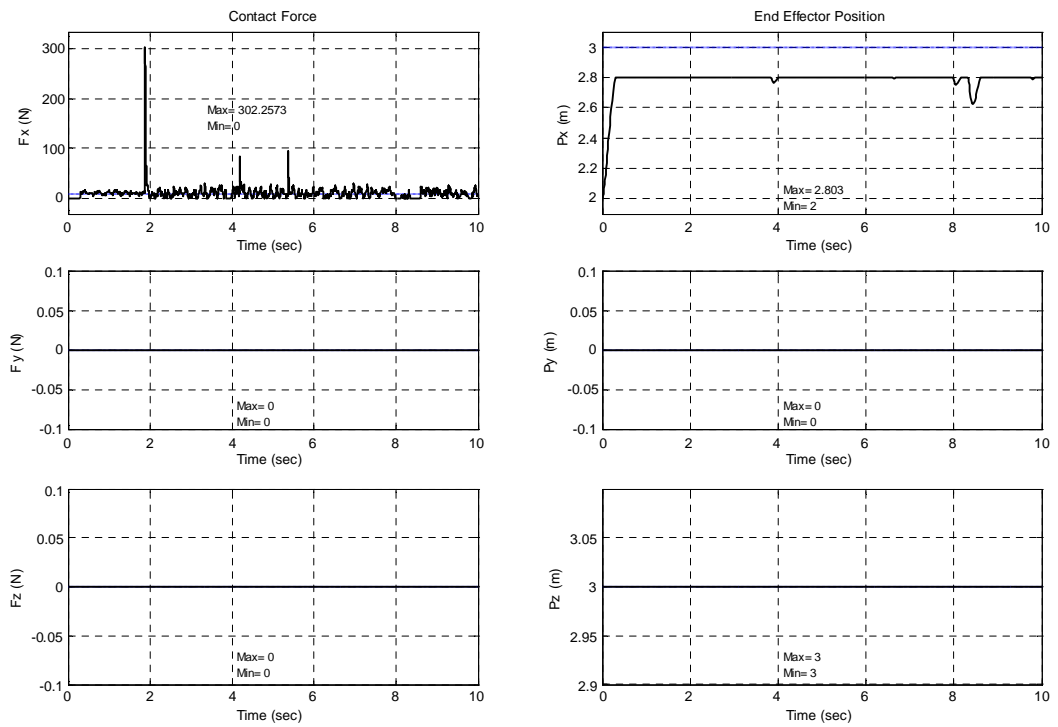


Figure IV.10 The worst value of the Force and Position graph of RAFC of cartesian type of robot

IV.2.3 Taguchi L9 (3⁴) Table of Parallel Force/Position Control of Cartesian Type of Robot Arm

At the 3. row, of Table IV.6, the value of PID is maximum, the value of $Sqrt_{EMS}$ is minimum. At the 1. row the value of $Sqrt_{EMS}$ is maximum with minimum PID value. Desired settling time is 10 sec. but non of them can reach this value. The nearest settling time is 13,6 sec at the 7. row Table IV.5. However, because of the biggest overshoot and big ST values, row 7 is the worst of them.

Table IV.5 Taguchi arrays of parallel force/position control of cartesian type of robot (a)

(a)	P	I	D	Error	MxO	%MxO	S.T.	C.T.
1	12,00	1,60	7,50	0,00	25,60	156,00	26,30	423,30
2	24,00	3,20	15,00	0,00	18,05	80,50	24,00	890,70
3	36,00	4,80	22,50	0,00	15,50	55,00	22,00	1324,00
4	12,00	3,20	22,50	0,00	22,50	125,00	20,00	1360,00
5	24,00	4,80	7,50	0,00	18,30	83,00	19,00	450,90
6	36,00	1,60	15,00	0,00	15,70	57,00	35,00	906,70
7	12,00	4,80	15,00	0,00	22,60	126,00	13,60	879,40
8	24,00	1,60	22,50	0,00	18,12	81,20	25,00	1323,00
9	36,00	3,20	7,50	0,00	15,80	58,00	24,00	428,90
Max=				1,00		156,00	35,00	1360,00

Table IV.6 Normalizing Taguchi arrays of parallel force/position control of cartesian type of robot (b, c)

	Divided by maximum (b)				(c)	
	ER	MPR	STR	CTR	$Sqrt_{EMSC}$	$Sqrt_{EMS}$
1	0,000	1,000	0,751	0,311	1,289	1,251
2	0,000	0,516	0,686	0,655	1,080	0,858
3	0,000	0,353	0,629	0,974	1,211	0,721
4	0,000	0,801	0,571	1,000	1,403	0,984
5	0,000	0,532	0,543	0,332	0,829	0,760
6	0,000	0,365	1,000	0,667	1,256	1,065
7	0,000	0,808	0,389	0,647	1,105	0,896
8	0,000	0,521	0,714	0,973	1,314	0,884
9	0,000	0,372	0,686	0,315	0,841	0,780

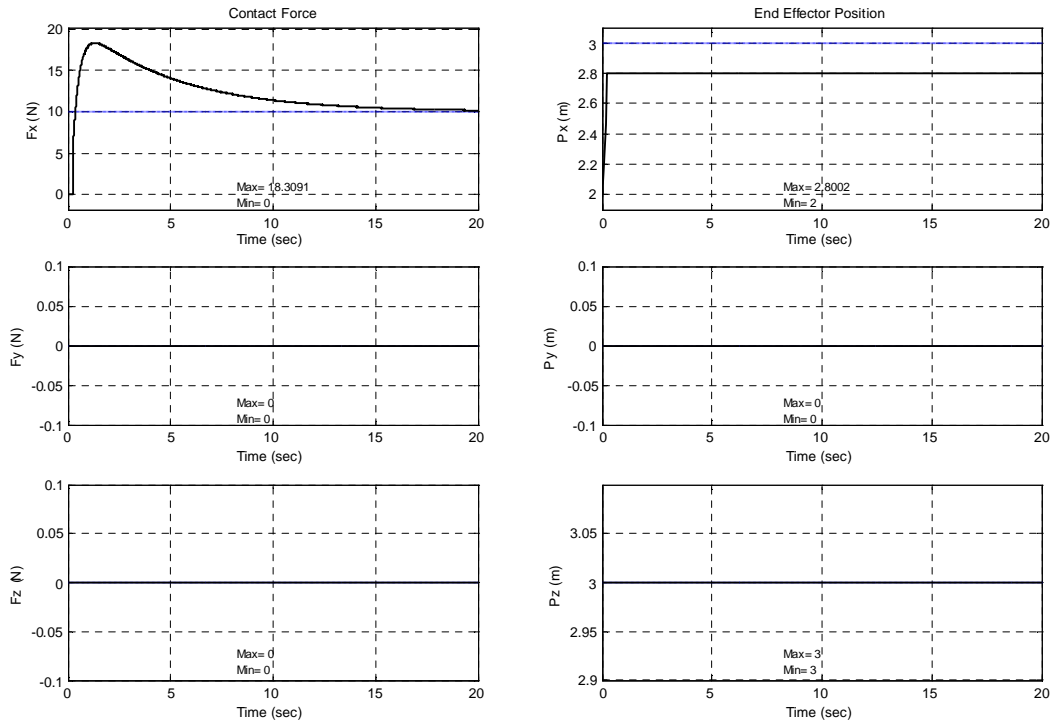


Figure IV.11 The best value of the force and position graph of parallel force/position control of cartesian type of robot

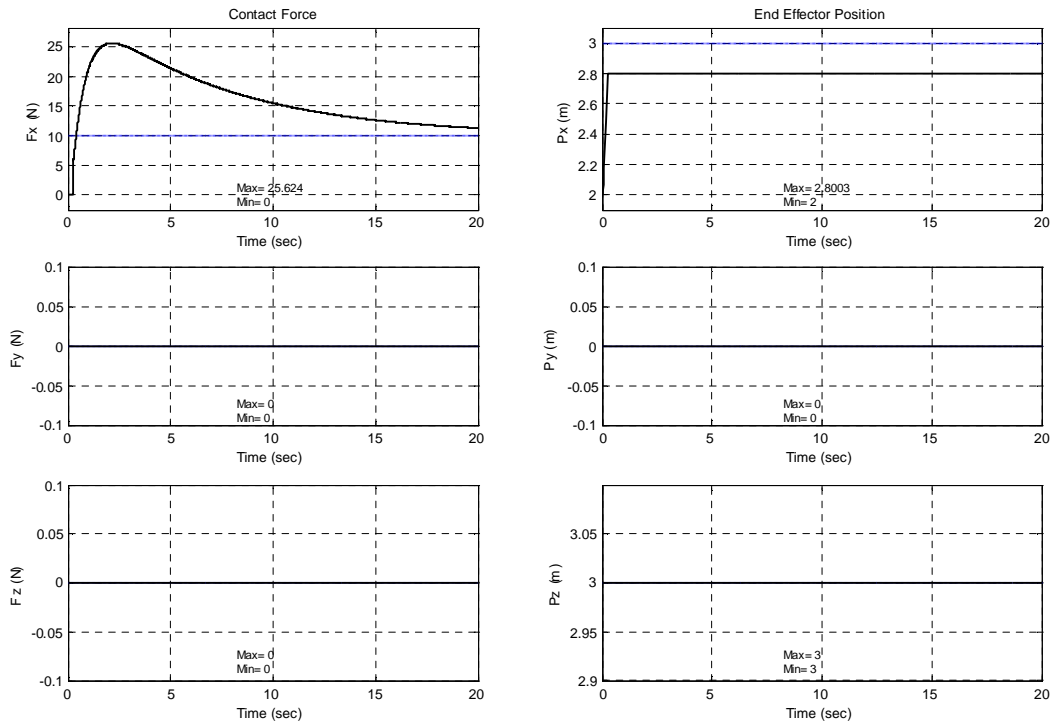


Figure IV.12 The worst value of the force and position graph of parallel force/position control of cartesian type of robot

IV.2.4 Taguchi L9 (3⁴) Table of Hybrid Position/Force Control of Revolute Type Of Robot Arm

When derivative parameter is less than 10⁻⁶ simulation works. For the derivative parameter that is greater than this value which the roots are not real, simulation is not run. That's why derivative parameter value is set to zero. Only PI parameters affect the results. But non of them settled during simulation. The contact force varies between 0 and 19 N. Because of chattering, the system cannot reach the desired force. The value of ST is written in the table as NaN its value is set zero for calculation. The row number of best and worst values is the same as hybrid control of cartesian type of robot. Minimum value of Sqrt_{EMS} is 0,962 at row 3 and maximum value is. 1.00 at row 1 Table IV.7, Table IV.8

Table IV.7 Taguchi arrays of Hybrid Position/Force Control of revolute type of robot (a)

(a)	P	I	D	Error	MxO	%MxO	S.T.	C.T.
1	175	25	0	0	22,46	124,6	NaN	170,4
2	350	50	0	0	20,65	106,5	NaN	166,1
3	525	75	0	0	20,87	108,7	NaN	210,3
4	175	50	0	0	22,45	124,5	NaN	144,7
5	350	75	0	0	20,66	106,6	NaN	143,3
6	525	25	0	0	20,88	108,8	NaN	263,5
7	175	75	0	0	22,43	124,3	NaN	126,4
8	350	25	0	0	20,41	104,1	NaN	204,5
9	525	50	0	0	20,99	109,9	NaN	231,0
Max=				1,00		124,6	1,00	263,5

Table IV.8 Normalizing Taguchi arrays of Hybrid Position/Force Control of revolute type of robot (b, c)

	Divided by maximum (b)				(c)	
	ER	MPR	STR	CTR	Sqrt _{EMSC}	Sqrt _{EMS}
1	0,0000	1,0000	0,0000	0,6467	1,1909	1,0000
2	0,0000	0,8547	0,0000	0,6304	1,0620	0,8547
3	0,0000	0,8724	0,0000	0,7981	1,1824	0,8724
4	0,0000	0,9992	0,0000	0,5491	1,1402	0,9992
5	0,0000	0,8555	0,0000	0,5438	1,0138	0,8555
6	0,0000	0,8732	0,0000	1,0000	1,3276	0,8732
7	0,0000	0,9976	0,0000	0,4797	1,1069	0,9976
8	0,0000	0,8355	0,0000	0,7761	1,1403	0,8355
9	0,0000	0,8820	0,0000	0,8767	1,2436	0,8820

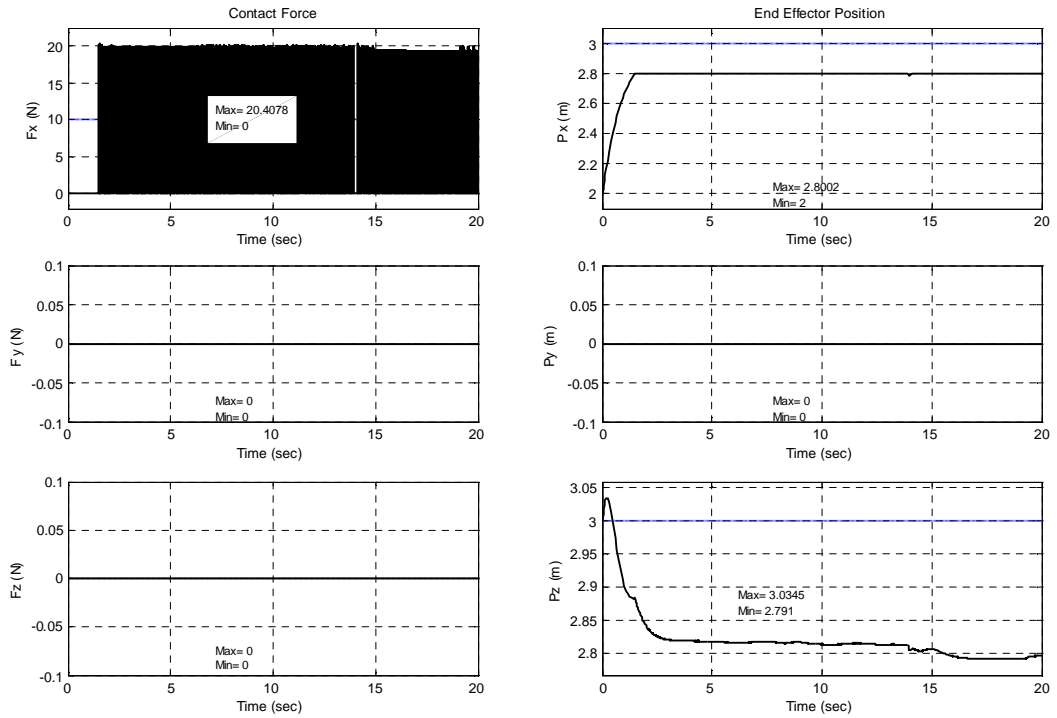


Figure IV.13 The best value of the force and position graph of Hybrid Position/Force Control of revolute type of robot

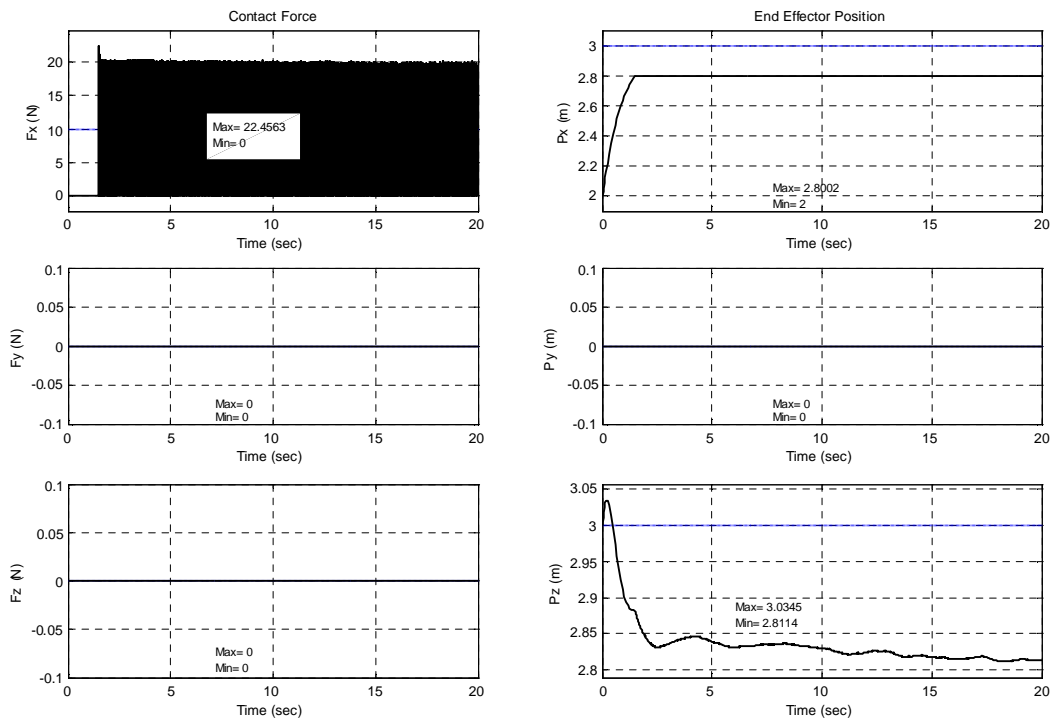


Figure IV.14 The worst value of the force and position graph of Hybrid Position/Force Control of revolute type of robot

IV.2.5 Taguchi L9 (3⁴) Table of RAFC of Revolute Type of Robot Arm

This force control algorithm has the maximum contact force after impact with small PI parameters that are given at the 1. row of Table IV.9. Like former force control algorithm small impact value is occurs for large PI parameters that are given at the 2. row of the table. There is a %2.86 positional error at the z-axis as seen in the Figure IV.15 and Figure IV.16. This error occurs at the all PI values. This difference may caused by gravitational effect. This problem always exists for RAFC of revolute type of robot arm.

Table IV.9 Taguchi arrays of RAFC of revolute type of robot (a)

(a)	P	I	D	Error	MxO	%MxO	S.T.	C.T.
1	2,50	1,25	0,03	0,00	36,11	261,10	7,59	34,83
2	5,00	2,50	0,05	0,00	20,07	100,70	6,92	59,36
3	7,50	3,75	0,08	0,00	17,89	78,90	6,64	82,13
4	2,50	2,50	0,08	0,00	27,18	171,80	4,66	72,86
5	5,00	3,75	0,03	0,00	35,26	252,60	5,32	41,83
6	7,50	1,25	0,05	0,00	19,00	90,00	14,08	64,19
7	2,50	3,75	0,05	0,00	32,35	223,50	3,64	53,81
8	5,00	1,25	0,08	0,00	16,48	64,80	11,37	78,08
9	7,50	2,50	0,03	0,00	32,24	222,40	8,63	46,59
Max=				1,00		261,10	14,08	82,13

Table IV.10 Normalizing Taguchi arrays of RAFC of revolute type of robot (b, c)

	Divided by maximum (b)				(c)	
	ER	MPR	STR	CTR	Sqrt _{EMSC}	Sqrt _{EMS}
1	0,000	1,000	0,539	0,424	1,213	1,136
2	0,000	0,386	0,491	0,723	0,955	0,625
3	0,000	0,302	0,472	1,000	1,146	0,560
4	0,000	0,658	0,331	0,887	1,153	0,737
5	0,000	0,967	0,378	0,509	1,157	1,039
6	0,000	0,345	1,000	0,782	1,315	1,058
7	0,000	0,856	0,259	0,655	1,109	0,894
8	0,000	0,248	0,808	0,951	1,272	0,845
9	0,000	0,852	0,613	0,567	1,193	1,049

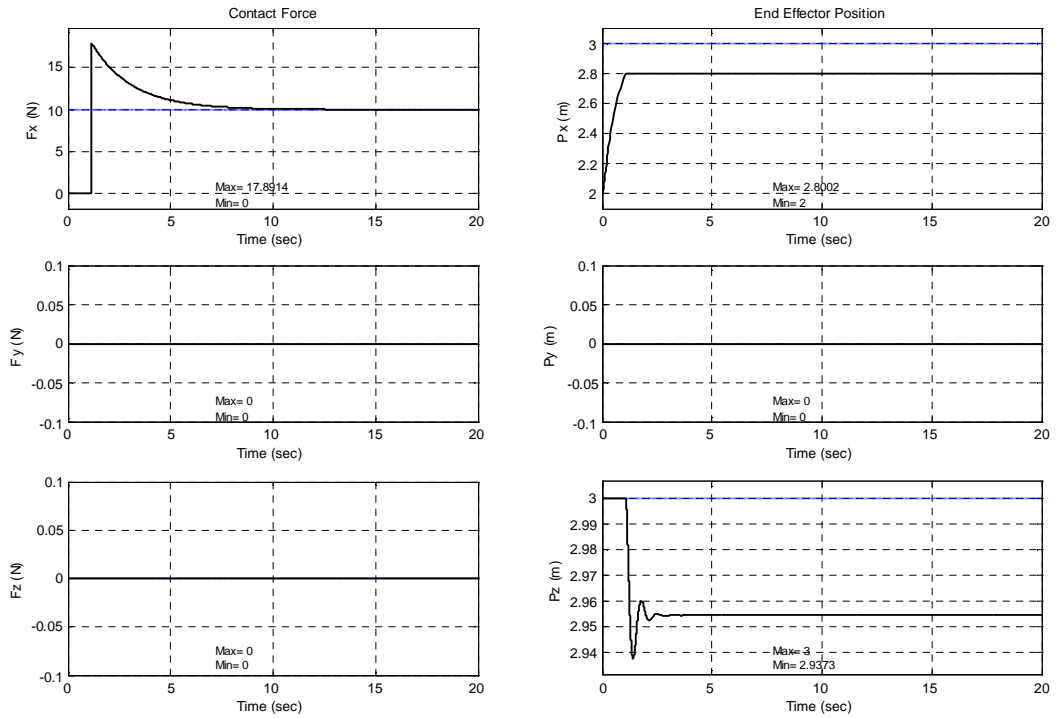


Figure IV.15 The best value of the Force and Position graph of RAFC of revolute type of robot

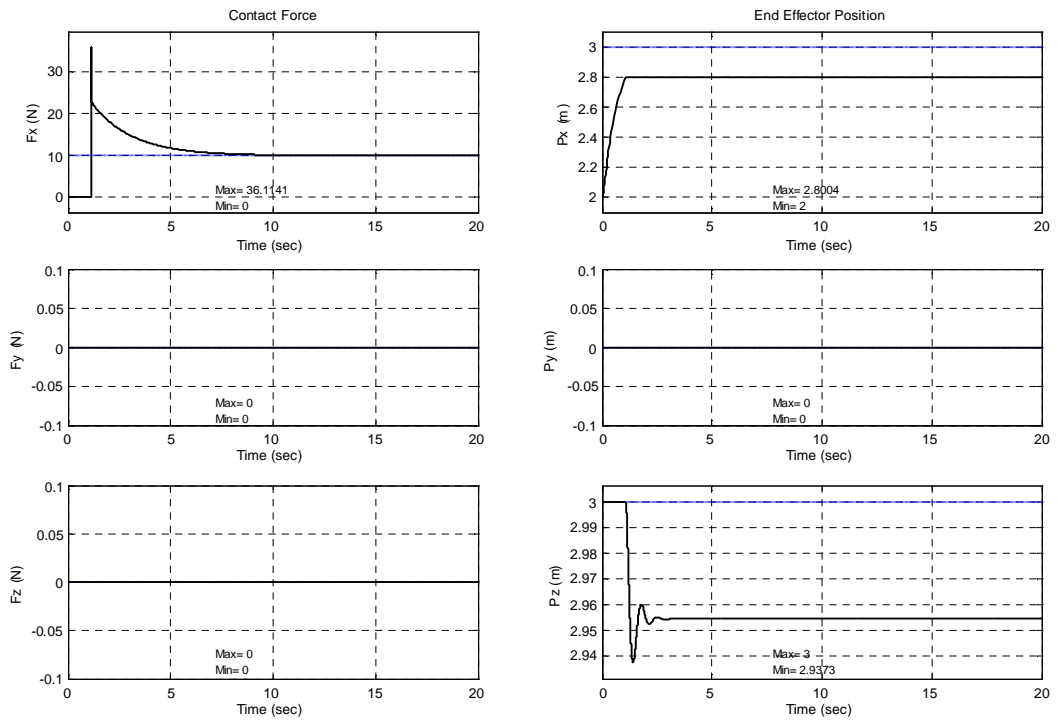


Figure IV.16 The worst value of the Force and Position graph of RAFC of revolute type of robot

IV.2.6 Taguchi L9 (3⁴) Table of Parallel Force/Position Control of Revolute Type of Robot Arm

The results of PI of this type of force control is different from other types of force control of both two types of robot arms. From the Table IV.12 it is seen that the best value is at the 7. row and the worst value is at the 6. row. Increasing “I” value cause decreasing ST. Increasing or decreasing of Integral parameter which is effect to the MxO is not gave any idea. On the other hand increasing P cause increasing MxO. Same thing is occurred for affect on the ST for changing P value Table IV.11.

Table IV.11 Taguchi arrays of parallel force/position control of revolute type of robot (a)

(a)	P	I	D	Error	MxO	%MxO	S.T.	C.T.
1	4,95	0,50	0,05	0,00	22,48	124,80	24,60	39,92
2	9,90	1,00	0,10	0,00	18,46	84,60	17,99	73,03
3	14,85	1,50	0,15	0,00	18,57	85,70	14,10	105,00
4	4,95	1,00	0,15	0,00	17,20	72,00	13,90	96,23
5	9,90	1,50	0,05	0,00	35,35	253,50	13,06	47,73
6	14,85	0,50	0,10	0,00	25,19	151,90	32,80	79,42
7	4,95	1,50	0,10	0,00	18,15	81,50	10,10	68,14
8	9,90	0,50	0,15	0,00	13,15	31,50	30,10	102,40
9	14,85	1,00	0,05	0,00	44,71	347,10	19,15	54,22
Max=				1,00		347,10	32,80	105,00

Table IV.12 Taguchi arrays of Parallel Force/Position Control of revolute type of robot (b, c)

	Devided by maximum (b)				(c)	
	ER	MPR	STR	CTR	Sqrt _{EMSC}	Sqrt _{EMS}
1	0,000	0,360	0,750	0,380	0,915	0,832
2	0,000	0,244	0,548	0,696	0,919	0,600
3	0,000	0,247	0,430	1,000	1,116	0,496
4	0,000	0,207	0,424	0,916	1,031	0,472
5	0,000	0,730	0,398	0,455	0,948	0,832
6	0,000	0,438	1,000	0,756	1,328	1,092
7	0,000	0,235	0,308	0,649	0,756	0,387
8	0,000	0,091	0,918	0,975	1,342	0,922
9	0,000	1,000	0,584	0,516	1,268	1,158

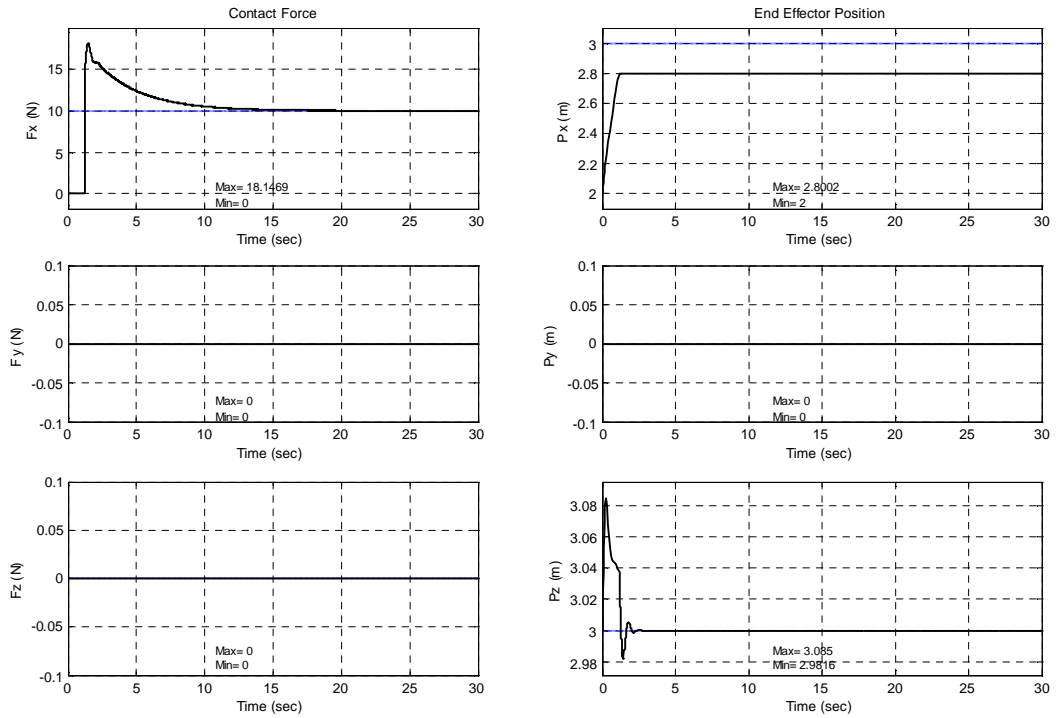


Figure IV.17 The best value of the force and position graph of parallel force/position control of revolute type of robot

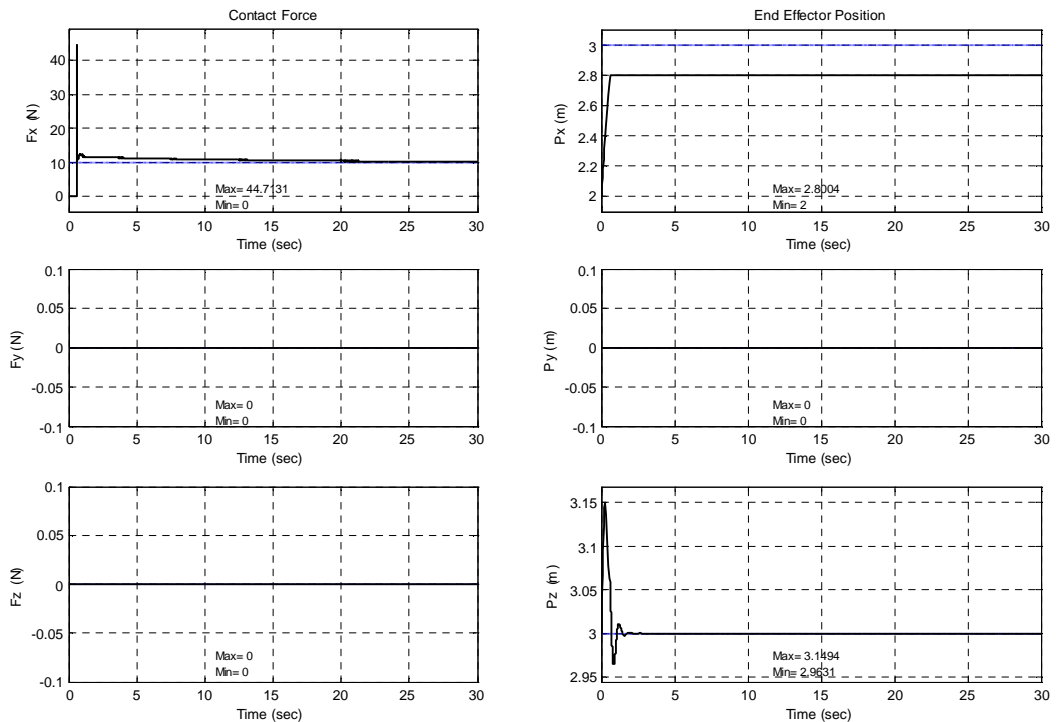


Figure IV.18 The worst value of the force and position graph of parallel force/position control of revolute type of robot

IV.3. EVALUATION OF FINAL RESULTS

Table V.13-15 present the final results are for evaluation of force control schemes.

Row number of the best and the worst values are given in the Table IV.13. In this table, it is understand the effect of the PID parameters of force control types on the robot arms.

For simplicity HP/FC abbreviated as Hybrid Position/Force Control, RAFC abbreviated as Resolved Acceleration Force Control and PF/PC abbreviated as Parallel Force/Position Control.

It can be seen that the response of HP/FC is good when PID parameters are big and bad when PID parameters are small for both two robot arms (Table V.15). The smallest values of MxO and ST of all experiments are get in HP/FC of Cartesian type of robot. On the other hand, HP/FC is not suitable for revolute type because of non-settling of contact force as it seen from Table V.15. MxO and ST of RAFC and PF/PC are nearest each other. But RAFC has small MxO and ST values than PF/PC.

Table V.13. Row number of the best and the worst response

	CARTESIAN		REVOLUTE	
	best	worst	best	Worst
HP/FC	3	1	8	1
RAFC	2	8	3	1
PF/PC	3	1	7	9

Levels of PID parameters of force control algorithms are listed in Table V.14. Levels are shown as min for Level 1, mean for Level 2 and max for Level 3.

Table V.14. PID levels of the best and the worst response

	CARTESIAN						REVOLUTE					
	best			Worst			best			worst		
	P	I	D	P	I	D	P	I	D	P	I	D
HP/FC	max	max	max	min	min	min	mean	min	-	min	min	-
RAFC	max	mean	min	mean	min	max	max	max	max	min	min	min
PF/PC	max	max	max	min	min	min	min	max	mean	max	max	min

At maximum value of proportional gain of all force control algorithms for Cartesian robot arm cause the best response.

The best and the worst results of MxO and ST that are taken from force control schemes for robot arms are given in the Table V.15.

Table V.15. MxO and ST of the best and the worst response

	CARTESIAN				REVOLUTE			
	best		worst		best		worst	
	MxO	ST	MxO	ST	MxO	ST	MxO	ST
HP/FC	13.35	5.00	38.03	11.50	20.41	NaN	20.46	NaN
RAFC	21.50	25.00	302.00	100.00	17.89	6.64	36.11	7.59
PF/PC	15.50	22.00	25.60	26.30	18.15	10.10	44.71	19.15

As a result the best force control algorithm is Hybrid Position/Force Control for cartesian and Resolved Acceleration Force Control for revolute type.

PART V.

CONCLUSIONS

In this thesis, three kinds of force control algorithms are used. This force control algorithms are hybrid force/position control, resolved acceleration force control and parallel force/position control. These control algorithms are applied to two different types of robot manipulator for comparison.

The first control type, hybrid force/position control, is one of the earlier force control type. It is an example of switching controls between force and position control. Only one of two control schemes is active by sensing contact force.

Hybrid Position/Force Control of Cartesian type robot is sensitive to PID parameters. Settling times are smaller than all other force control of two types of robots. MxO values vary between 13.30 and 38.03. This means that this type of force control can be used for Cartesian type robot if PID parameters adjusted carefully.

On the other hand, Hybrid Position/Force Control for revolute type is not suitable because of non-stable response. MxO values are very close to 20 N. This values is the allowable upper limit of the maximum over shoot value.

The second force control type, resolved acceleration control (RAFC), has similar force and position control algorithm with hybrid control. The difference between them is switching function. Force control activate when force sensed and

feedback value is summing with the position control value. An addition to position control type is feedback from acceleration. This is the extra in the control algorithm.

RAFC of both robots are not suitable for force control. MxO values are out of range for both robot types. Settling times for Cartesian type of robot are too long from other types. RAFC is not advised to use for force control.

The third force control type, Parallel Force/Position Control, quite different from two former force control type. Position control depends on only position error. Velocity feedback is directly added to control value. Force control part is also different from others. Desired force and force error are used at the same time with position control.

Parallel Force/Position Control of Cartesian type is looks stable to PID variation. However, settling time is long from Hybrid Position/Force Control. And simulation time is the longest of all the other runs.

Response of revolute type of robot is good at the parallel force/position control. MxO value is lower then the other two force control of this robot.

This thesis shows that simulations of robot arms using Simulink is helpful for researchers to adjust gains of control algorithms and test them virtually before using real robot manipulators. Testing different control algorithms with different situations in virtual world is useful to avoid robot manipulators from dangerous applications in real world. Building models are simple using blocks. Changes in model are quite easy to test at the different conditions and different values of parameters. Mathematical model of robot manipulator is important to simulate real world in virtual world. This is important case of simulation. Especially it is very useful for educational purpose to teach robot control.

REFERENCES

- [1.] Fasse, E. D.; Broenink, J. F.: "A Spatial Impedance Controller for Robotic Manipulation", IEEE International Conference on Robotics and Automation, Vol. XX (1999) 1-12.
- [2.] Broenink, J. F.; Visser, M; Tiernego, M. L. J.: "Simulation of Sliding Mode Impedance Control for a 6 DOF Anthropomorphic Robot", Proceeding of the 12th European Simulation Multiconference, Manchester UK, June 16-17 (1998) 561-565.
- [3.] Zuharas, N.; Hsia, T.C.: "Nonhomogenous Material Milling Using A Robot Manipulator With Force Controlled Velocity", IEEE International Conference on Robotics and Automation, (1995) 1461-1467.
- [4.] Jinno, M.; Ozaki, F.; Tatsuno, K.; Takahashi, M. et al: "Development of a Force Controlled Robot for Grinding, Chamfering and Polishing", IEEE International Conference on Robotics and Automation, (1995) 1455-1460.
- [5.] Jinno, M.; Yoshimi, T.; Abe, A.: "Force Controlled Grinding Robot System for Unstructured Tasks", Proc. Of the IEEE/RSJ International Conference on Intelligent Robots and Systems, July 7-10 (1992) 1117-1124.
- [6.] Nonaka, Y.; Sakaue, S.; Yanagihara, Y.; Yokoshima, K: "Development of an Impeller Grinding Robot System and a Gyro-moment Compensated Compliance Control", IEEE International Conference on Robotics and Automation, (1995) 2084-2089.
- [7.] Hogan, N.: "Impedance Control: An Approach o Manipulation: Part I-II-III", Journal of Dynamics Systems, Measurement and Control, Vol. 107 (1985) 1-24.
- [8.] Luo, Z.; Ito, M.: June 1993, "Control Design of Robot for Compliant Manipulation on Dynamic Environment", IEEE International Conference on Robotics and Automation, Vol. 9 (June 1993) 286-296.
- [9.] Yoshikawa, T.: Foundation of Robotics Analysis and Control, MIT Press, Cambridge, Massachusetts, (1990).
- [10.] Hogan, N.: "Impedance Control of Robotic Manipulator", Winter Annual Meeting of the ASME, (1981).
- [11.] Shilling, R. J.: "Fundamental of Robotics Analysis and Control", Prentice Hall, New York, (1990).
- [12.] Peng, Z.: "Compliant Motion Control of Manipulators", Thesis of Doctor of Philosophy, Niigata University, Niigata Japan, (1992).
- [13.] Whitney, D.E.: "Force Feedback Control of Manipulator Fine Motions", J. Dynamics Systems, Measurement and Control, June, (1977) 91-96.

- [14.] Hyde, J. M.; Cutkosky, M. R.: "Contact Transition Control: An Experimental Study", Center for Design Research, Stanford University, Stanford, California, USA (Personal communication), (2000).
- [15.] Simons, J.; Van Brussel, H.: "Force Control Schemes for Robot Assembly",
- [16.] Volpe, R.; Khosla, P.: "An Analysis of Manipulator Force Control Strategies Applied to an Experimentally Derived Model", Proc. Of the IEEE/RSJ International Conference on Intelligent Robots and Systems, July 7-10 (1992) 1989-1997.
- [17.] Borowiec, J.; Tzes, A.: "Frequency-Shaped Implicit Force Control of Flexible Link Manipulators", IEEE International Conference on Robotics and Automation, (1995) 913-918.
- [18.] Valency, T.; Zacksenhouse, M.: "Instantaneous Model Impedance Control for Robots", Sensory-Motor Integration Laboratory, Faculty of Mechanical Engineering, Technion- Israel Institute of Technology (Personal communication), (2000).
- [19.] Mason, M. T.: "Compliance and Force Control for Computer Controlled Manipulators", IEEE Transection on Systems, Man, and Cybernetics SMC-11, June 6 (1981) 418-432.
- [20.] Kazerooni, H.; Waibel, B. J.; Kim, S.: "On the Stability of Robot Compliant Motion Control: Theory and Experiments", Journal of Dynamic Systems, Measurement, and Control, Vol. 112 September (1990) 417-426.
- [21.] [21] Salisbury, J. K.: "Active Stiffness Control of a Manipulator in Cartesian Coordinates", Proceeding 19th IEEE Conference on Decision and Control, December (1980) 95-100.
- [22.] Craig, J. J.: "Introduction to Robotics Mechanics and Control", 2nd Ed., Addison-Wesley Pub. Comp., (1989)
- [23.] Sciavicco, L.; Siciliano, B.: "Modeling and Control of Robot Manipulators", The McGraw-Hill Companies Inc., (1996)
- [24.] Siciliano, B.; Villani, L.: "Robot Force Control", Kluwer Academic Publishers, Norwell, MA, (1999)
- [25.] Kim, S.; Park, M.: "Fuzzy Compliance Robot Control", IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 1628-1631.
- [26.] Kim, M. J.; Kwon, C.; Park, M. K.; Park, M.: "A Stiffness Control of a Manipulator Using a Fuzzy Model", IEEE Conference on Intelligent Robots and Systems – IROS '95, (1995) 322-327.
- [27.] Jazidie, A.; Tsuji, T.; Ito, K.: "Multi-Point Compliance Control of Dual-Arm Robots", Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, July 7-10 (1992) 65-70.
- [28.] Yokoi, K.; Maekawa, H.; Tanie, K.: "A Method of Compliance Control for a Redundant Manipulator", Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, July 7-10 (1992) 1927-1934.
- [29.] Oh, S. R.; Kim, H. C.; Suh, I. H.; You, B. J.; Lee, C. W.: "A Compliance Control Strategy for Robot manipulators Using a Self-Controlled Stiffness Function", IEEE Conference on Intelligent Robots and Systems – IROS '95, (1995) 179-184.
- [30.] Fisher, W. D.; Mujtaba, M. S.: "Hybrid Position/Force Control: A Correct Formulation", Measurement And Manufacturing Systems Laboratory, Hewlett-Packard Company, HPL-91-140 October (1991)

- [31.] Fisher, W. D.; Mujtaba, M. S.; Sinha, P.: "A 2D Study of Hybrid Position/Force Control", Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, July 7-10 (1992) 1980-1998
- [32.] Fisher, W. D.; Mujtaba, M. S.: "A Kinematically Stable Hybrid Position/Force Control Scheme", Measurement And Manufacturing Systems Laboratory, Hewlett-Packard Company, HPL-91-154 October (1991)
- [33.] Seraji, H.: "Adaptive Force Control in Compliant Motion", California Institute of Technology, Pasadena, CA, (Personal communication), (2000).
- [34.] Fraisse, P.; Delebarre, X.; Dauchez, P.; Pierrot, F.: "Towards Robust Hybrid Control for Two-Arm Robots", IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 331-336.
- [35.] Luca, A. D.; Mattone, R.: "Modeling and Control Alternatives for Robots in Dynamic Cooperation", IEEE International Conference on Robotics and Automation, (1995) 138-145.
- [36.] Natale, C.; Siciliano, B.; Villani, L.: "Robust Hybrid Force/Position control with Experiments on an Industrial Robot", <http://disna.dis.unina.it/prisma>.
- [37.] Oaki, J.: "Stable Force Controller Design Based on Frequency Response Identification", IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 1116-1121.
- [38.] Suh, I. H.; Hong, J. H.; Oh, S. R.; Kim, K. B.: "Fuzzy Rule Based Position/Force Control of Industrial Manipulator", IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 1617-1622.
- [39.] Hsu, F. Y.; Fu, L. C.: "A New Design of Adaptive Fuzzy Hybrid Force/Position Controller for Robot Manipulators", IEEE International Conference on Robotics and Automation, (1995) 863-868.
- [40.] Kim, H. G.: "A New Interpretation of Force/Servo Control of Robot Arms", IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 1623-1627.
- [41.] Hu, V.; Queiroz, M.; Burg, T.; Dawson, D.: "Adaptive Position/Force Control of Robot Manipulator Without Velocity Measurements", IEEE International Conference on Robotics and Automation, (1995) 887-892.
- [42.] Yang, J. H.; Lian, F. L.; Fu, L. C.: "Adaptive Hybrid Position/Force Control for Robotic Manipulators with Compliant Links", IEEE International Conference on Robotics and Automation, (1995) 603-608.
- [43.] Natale, C.; Villani, L.: "Adaptive Control of a Robot Manipulator in Contact with a Curved Compliant Surface", Proceeding of the 1999 American Control Conference, San Diego, CA, (1999) 288-292.
- [44.] Adachi, N.; Peng, Z. X.; Nakajima, S.: "Compliant Motion Control of Redundant Manipulators", IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 137-142.
- [45.] Caccavale, F.; Siciliano, B.; Villani, L.: "Robot Impedance Control with Nondiagonal Stiffness", IEEE Transection on Automatic Control, Vol. 44 (1999) 1943-1946.
- [46.] Chiaverini, S.: "A Survey of Robot Interaction Control Schemes with Experimental Comperison", IEEE/ASME Transection on Mechatronics, Vol. 4 (1999) 273-285.
- [47.] Meer, D. W.; Rock, S. M., "Experiments in Object Impedance Control for Flexible Objects", IEEE International Conference on Robotics and Automation, (1994).

- [48.] Wu, Y.; Tarn, T.; Xi, N.: “Force and Transition Control with Environmental Uncertainties”, IEEE International Conference on Robotics and Automation, (1995) 899-903 .
- [49.] Harada, T.; Nishida, Y.; Imamura, N.; Kimura, N.: “Robust Implementation of Impedance Control Using Impedance Error Feedback”, Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, July 7-10 (1992) 1998-2004.
- [50.] Tzafestas, C.; Guihard, M.; M’Sirdi, N. K.: “Two-Stage Adaptive Control Applied to a Legged Robot”, IEEE Conference on Intelligent Robots and Systems – IROS ’95, (1995) 173-178.
- [51.] Colbaugh, R.; Seraji, .; HGlass, K.: “Adaptive Compliant Motion Control for Dexterous Manipulators”, Robotica, Cambridge University Press, Vol. 14 No.3 (1995) 270-280.
- [52.] Colbaugh, R.; Glass, K.: “Decentralized Adaptive Compliance Control of Robot Manipulators”, Robotica, Cambridge University Press, Vol. 13 (1995) 485-498
- [53.] Jung, S.; Hsia, T. C.: “On Neural Network Application to Robust Impedance Control of Robot Manipulators”, IEEE International Conference on Robotics and Automation, (1995) 869-874.
- [54.] Lu, Z.; Kawamura, S.; Goldenberg, A. A.: “Sliding Mode Impedance Control and Its Application to Grinding Tasks”, IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 350-355.
- [55.] Visser, M.: “Sliding Mode Impedance Control”, MSc. Thesis, Electrical Engineering Department, University of Twente, Netherlands, December (1997).
- [56.] Meer, D. W.; Rock, S. M., “Coupled-System Stability of Flexible-Object Impedance Control”, IEEE International Conference on Robotics and Automation, (1995) 1839-1845.
- [57.] Sun, D.; Liu, Y.: “Modeling and Impedance Control of a Two-Manipulator System Handling a Flexible Beam”, Journal of Dynamic Systems, Measurement, and Control, Vol. 119 December (1997) 736-742.
- [58.] Almeida, F.; Lopes, A.; Abreu, P.: “Force-Impedance Control: a New Control Strategy of Robotic Manipulators”, 126-137.
- [59.] Inoue, Y.; Kitamura, S.; Kidawara, Y.: “Force Feedback Control and Collision Avoidance of Redundant Manipulator”, IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 149-152.
- [60.] Oh, Y.; Chung, W.; Youm, Y.: “Extended Impedance Control of Redundant Manipulators Based on Weighted Decomposition of Joint Space”, Journal of Robotic Systems, John Wiley & Sons. Inc, (1998) 232-258.
- [61.] Natale, C.; Siciliano, B.; Villani, L.: “Spatial Impedance Control of Redundant Manipulators”, Proceeding of the 1999 IEEE International Conference on Robotics and Automation, Detroit, MI, (1999) 1788-1793.
- [62.] Fasse, D. E.; Hogan, N.: “Control of Physical Contact and Dynamic Interaction”, Presented to the Seventh International Symposium of Robotics Research, October 21-24, Munchen, Germany, (1995).
- [63.] Fasse, E. D.: “On the Spatial Compliance of Robotic Manipulators”, J. Dynamics Systems, Measurement and Control, December (1997) 839-844.
- [64.] Cheah, C. C.; Wang, D.: “Learning Impedance Control for Robotic Manipulators”, IEEE International Conference on Robotics and Automation, (1995) 2150-2155.

- [65.] Caccavale, F.; Natale, C.; Siciliano, B.; Villani, L.: "Quaternion-Based Impedance Control for Dual-Robot Cooperation", 9th International Symposium of Robotics Research, Snowbird, UT, October (1999) 42-49.
- [66.] Ellis, R. E.; Sarkar, N.; Jenkins, M. A.: "Numerical Methods for the Force Reflection of Contact" Journal of Dynamic Systems, Measurement, and Control, Vol. 119 December (1997) 768-774.
- [67.] Carignan, C. R.; Cleary, K. R.: "Closed-Loop Force Control for Virtual Haptic Simulation of Virtual Environments", Haptics-e, www.haptics-e.org, Vol. 1, No. 2 (2000).
- [68.] Ikeura, R.; Inooka, H.: "Variable Impedance Control of a Robot Cooperation with a Human", IEEE International Conference on Robotics and Automation, (1995) 3097-3102.
- [69.] Love, L. J.; Book, W. J.: "Environment Estimation for Enhanced Impedance Control", IEEE International Conference on Robotics and Automation, (1995) 1854-1889.
- [70.] Yi, B. J.; Freeman, R. A.: "Feedforward Spring-like Modulation in Human Arm Models", IEEE International Conference on Robotics and Automation, (1995) 3121-3128.
- [71.] Mills, J. K.; Liu, G. J.: "Robotic Manipulator Impedance Control of Generalized Contact Force and Position", IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 1103-1108.
- [72.] Caccavale, F.; Natale, C.; Siciliano, B.; Villani, L.: "Six-DOF Impedance Control Based on Angle/Axis Representations", IEEE Transactions on Robotics and Automation, Vol. 15 (1999) 289-300.
- [73.] Chae, H. A.; Atkeson, C. G.; Hollerbach, J. M.: "Model Based Control of A Robot Manipulator", The MIT Press, Cambridge, MA, (1982).
- [74.] Rudas, I. J.; Mester, G.: "Industrial Robot Control in Case of Uncertain Dynamical Parameters", IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, November 3-5 (1991) 937-942.
- [75.] Li, Q.; Tso, S. K.; Poo, A. N.: "An Enhanced Computed-Torque Control Algorithm for Robot Manipulator", Proceeding Instn. Mechanical Engineers, Vol. 212 Part I (1998) 11-15.
- [76.] Chiaverini, S.; Siciliano, B.; Villani, L.: "A Survey of Robot Interaction Control Schemes with Experimental Comparison", Proc. IEEE/ASME Transactions on Mechatronics, Vol. 4, (1999) 273-285.
- [77.] SIMULINK User Manual, Mathworks, MA. (1999).
- [78.] Roy, R.: "A Primer On The Taguchi Method", Von Nostrand Reinhold, (1990)

APPENDICES

APPENDIX A

M FILES OF CARTESIAN TYPE OF ROBOT

CRT.M

```
function [sys,x0,str,ts] = crt(t,x,u,flag,px,py,pz,vx,vy,vz)
```

```
q=[px;py;pz];
```

```
qv=[vx;vy;vz];
```

```
switch flag,
```

```
% Initialization %
```

```
case 0,
```

```
  [sys,x0,str,ts]=mdlInitializeSizes(t,x,u,flag,q,qv);
```

```
% Derivatives %
```

```
case 1,
```

```
  sys=mdlDerivatives(t,x,u);
```

```
% Update %
```

```
case 2,
```

```
  sys=mdlUpdate(t,x,u);
```

```
% Outputs %
```

```
case 3,
```

```
  sys=mdlOutputs(t,x,u);
```

```
% GetTimeOfNextVarHit %
```

```
case 4,
```

```
  sys=mdlGetTimeOfNextVarHit(t,x,u);
```

```
% Terminate %
```

```
case 9,
```

```
  sys=mdlTerminate(t,x,u);
```

```
% Unexpected flags %
```

```
otherwise
```

```
  error(['Unhandled flag = ',num2str(flag)]);
```

```
end
```

```
% end sfuntmpl
```

```
%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
```

```

function [sys,x0,str,ts]=mdlInitializeSizes(t,x,u,flag,q,qv)

% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
% Note that in this example, the values are hard coded. This is not a
% recommended practice as the characteristics of the block are typically
% defined by the S-function parameters.
sizes = simsizes;

sizes.NumContStates = 6;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 6;
sizes.NumInputs = 3;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = [q;qv];
%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%
ts = [0 0];

% end mdlInitializeSizes

%
%=====
% mdlDerivatives
% Return the derivatives for the continuous states.
%=====
%
function sys=mdlDerivatives(t,x,u)

%4 space x1 x2 x3 x4 ==> q1 q2 dq1 dq2
[m,l,fc,pe]=crtparam;

%frictional term
b=[0;0;0];
g1=0;
g2=0;
g3=9.81*m(3);

m=[m(1)+m(2)+m(3) 0 0 ; 0 m(2)+m(3) 0; 0 0 m(3)];
gr=[g1; g2; g3];
to=[u(1); u(2); u(3)];
qa=(inv(m)*(to-gr-b));

sys = [x(4); x(5); x(6);qa ];

% end mdlDerivatives

```

```

%
%=====
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=====
%
function sys=mdlUpdate(t,x,u)

sys = [];

% end mdlUpdate

%
%=====
% mdlOutputs
% Return the block outputs.
%=====
%
function sys=mdlOutputs(t,x,u)

sys = [x(1);x(2);x(3);x(4);x(5);x(6)];
% end mdlOutputs

%
%=====
% mdlGetTimeOfNextVarHit
% Return the time of the next hit for this block. Note that the result is
% absolute time. Note that this function is only used when you specify a
% variable discrete-time sample time [-2 0] in the sample time array in
% mdlInitializeSizes.
%=====
%
function sys=mdlGetTimeOfNextVarHit(t,x,u)

sampleTime = 1; % Example, set the next hit to be one second later.
sys = t + sampleTime;

% end mdlGetTimeOfNextVarHit

%
%=====
% mdlTerminate
% Perform any end of simulation tasks.
%=====
%
function sys=mdlTerminate(t,x,u)

sys = [];

% end mdlTerminate

CRTGRAVITY.M
function g=crtgravity

m=crtparam;

g1=0;

```

```

g2=0;
g3=9.81*m(3);
g=[g1; g2; g3];

```

CRTPARAM.M

```

function [m,l,fc,pe]=crtparam
m1=8; m2=6; m3=4;
lz1=1.3; lx=0.1; lz2=0.05; ly=0.05; lz3=0.1;
m=[m1;m2;m3];
l=[lz1;lx;lz2;ly;lz3];
return

```

CRTFWDYN.M

```

function [sys,x0,str,ts] = crtfdyn(t,x,u,flag)

```

```

switch flag,
% Initialization %
case 0,
    [sys,x0,str,ts]=mdlInitializeSizes;
% Outputs %
case 3,
    sys=mdlOutputs(t,x,u,flag);
case { 1, 2, 4, 9 }
    sys = []; % Unused flags
% Unexpected flags %
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
% end sfuntmpl

%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
function [sys,x0,str,ts]=mdlInitializeSizes

sizes = simsizes;

sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 3;
sizes.NumInputs = 9;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = [];
%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times

```

```

%
ts = [-1 0];

%
%=====
% mdlOutputs
% Return the block outputs.
%=====
%
function sys=mdlOutputs(t,x,u,flag)

q=[u(1); u(2); u(3)];
qv=[u(4); u(5); u(6)];
qa=[u(7); u(8); u(9)];

m=crtinertia;
g=crtgravity;
b=[0; 0; 0];

sys=m*qa+g+b;

% end mdlOutputs

```

CRTINERTIA.M (Used in crtfdyn.m)

```

function m=crtinertia
m=crtparam;

m1=m(1); m2=m(2); m3=m(3);

%inertia matrix
m11=m1+m2+m3;
m12=0;
m13=0;
m21=0;
m22=m2+m3;
m23=0;
m31=0;
m32=0;
m33=m3;
m=[m11 m12 m13;m21 m22 m23;m31 m32 m33];

```

APPENDIX B

MATLAB FILES OF REVOLUTE TYPE OF ROBOT ARM

MICRO_SFUN.M

```
function [sys,x0,str,ts] = micro_sfun(t,x,u,flag,q1,q2,q3,qv1,qv2,qv3)% .qa1 .qa2.qa3)
q=[q1;q2;q3];
qv=[qv1;qv2;qv3];
%qa=[qa1;qa2;qa3];
switch flag,
% Initialization %
    case 0,
        [sys,x0,str,ts]=mdlInitializeSizes(t,x,u,flag,q,qv);% ,qa);
% Derivatives %
    case 1,
        sys=mdlDerivatives(t,x,u);
% Update %
    case 2,
        sys=mdlUpdate(t,x,u);
% Outputs %
    case 3,
        sys=mdlOutputs(t,x,u);
% GetTimeOfNextVarHit %
    case 4,
        sys=mdlGetTimeOfNextVarHit(t,x,u);
% Terminate %
    case 9,
        sys=mdlTerminate(t,x,u);
% Unexpected flags %
    otherwise
        error(['Unhandled flag = ',num2str(flag)]);
end
% end sfuntmpl
```

```

%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
function [sys,x0,str,ts]=mdlInitializeSizes(t,x,u,flag,q,qv);% ,qa

% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
% Note that in this example, the values are hard coded. This is not a
% recommended practice as the characteristics of the block are typically
% defined by the S-function parameters.
sizes = simsizes;

sizes.NumContStates = 6;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 6;
sizes.NumInputs = 3;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = [q;qv];
%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%
ts = [0 0];

% end mdlInitializeSizes

%
%=====
% mdlDerivatives
% Return the derivatives for the continuous states.
%=====
%
function sys=mdlDerivatives(t,x,u

q=[x(1); x(2); x(3)];
qv=[x(4); x(5); x(6)];
qs=[x(4)^2; x(5)^2; x(6)^2];
qp=[x(4)*x(5); x(4)*x(6); x(5)*x(6)];

ms=microstruct([q,qv],0);
m=ms.m;
cs=ms.cs;
cp=ms.cp;
g=ms.g;
b=(ms.b).*qv;% sign(qv);
invm=inv(m);
csq=cs*qs;

```

```

cpq=cp*qp;
to=[u(1); u(2); u(3)];
qa=(inv(m)*(to-g-cs*qs-cp*qp-b));
x(7)=qa(1);
x(8)=qa(2);
x(9)=qa(3);
sys = [x(4); x(5); x(6); x(7); x(8); x(9)];

% end mdlDerivatives

%
%=====
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=====
%
function sys=mdlUpdate(t,x,u)

sys = [];

% end mdlUpdate

%
%=====
% mdlOutputs
% Return the block outputs.
%=====
%
function sys=mdlOutputs(t,x,u)

sys = [x(1);x(2);x(3);x(4);x(5);x(6)];%x(7);x(8);x(9)];

% end mdlOutputs

%
%=====
% mdlGetTimeOfNextVarHit
% Return the time of the next hit for this block. Note that the result is
% absolute time. Note that this function is only used when you specify a
% variable discrete-time sample time [-2 0] in the sample time array in
% mdlInitializeSizes.
%=====
%
function sys=mdlGetTimeOfNextVarHit(t,x,u)

sampleTime = 1; % Example, set the next hit to be one second later.
sys = t + sampleTime;

% end mdlGetTimeOfNextVarHit

%
%=====
% mdlTerminate
% Perform any end of simulation tasks.
%=====
%
function sys=mdlTerminate(t,x,u)

sys = [];

```

```
% end mdlTerminate
```

MICRODIRKIN_SFUN.M

```
function [sys,x0,str,ts] = microdirkin_sfun(t,x,u,flag)
```

```
switch flag,  
% Initialization %  
    case 0,  
        [sys,x0,str,ts]=mdlInitializeSizes;  
% Outputs %  
    case 3,  
        sys=mdlOutputs(t,x,u);  
    case { 1, 2, 4, 9 }  
        sys = []; % Unused flags  
% Unexpected flags %  
    otherwise  
        error(['Unhandled flag = ',num2str(flag)]);  
end  
% end sfuntmpl
```

```
%=====
```

```
% mdlInitializeSizes  
% Return the sizes, initial conditions, and sample times for the S-function.  
%=====
```

```
function [sys,x0,str,ts]=mdlInitializeSizes
```

```
sizes = simsizes;  
  
sizes.NumContStates = 0;  
sizes.NumDiscStates = 0;  
sizes.NumOutputs = 6;  
sizes.NumInputs = 6;  
sizes.DirFeedthrough = 1;  
sizes.NumSampleTimes = 1; % at least one sample time is needed
```

```
sys = simsizes(sizes);
```

```
%  
% initialize the initial conditions  
%  
x0 = [];  
%  
% str is always an empty matrix  
%  
str = [];  
  
%  
% initialize the array of sample times  
%  
ts = [-1 0];
```

```
%  
%=====
```

```
% mdlOutputs  
% Return the block outputs.  
%=====
```

```
%  
function sys=mdlOutputs(t,x,u)
```

```

c1=cos(u(1));
s1=sin(u(1));
c2=cos(u(2));
s2=sin(u(2));
c3=cos(u(3));
s3=sin(u(3));
c23=cos(u(2)+u(3));
s23=sin(u(2)+u(3));

qv1=u(4);
qv2=u(5);
qv3=u(6);
qv23=qv2+qv3;

l=microstruct([0;0;0;0;0],7);
lh=l(1); le=l(2); lf=l(3);

Px=le*c1*c2+lf*c1*c23;
Py=le*s1*c2+lf*s1*c23;
Pz=lh+le*s2+lf*s23;

vx=-le*s1*c2*qv1-le*c1*s2*qv2-lf*s1*c23*qv1-lf*c1*s23*qv23;
vy= le*c1*c2*qv1-le*s1*s2*qv2+lf*c1*c23*qv1-lf*s1*s23*qv23;
vz= le*c2*qv2+lf*c23*qv23;

x=[Px;Py;Pz;vx;vy;vz];

sys = [x];

% end mdlOutputs

```

MICROFWDYN.M

```
function torque=microfwdyn(u)
```

```

qv=[u(4); u(5); u(6)];
qa=[u(7); u(8); u(9)];
qp=[qv(1)*qv(2); qv(1)*qv(3); qv(2)*qv(3)];

ms=microstruct(u,0);
m=ms.m;
cs=ms.cs;
cp=ms.cp;
g=ms.g;
b=(ms.b).*qv;

torque=m*qa+cs*qv.^2+cp*qp+g+b;

```

MICROJACOBIAN_SFUN.M

```

function [sys,x0,str,ts] = microjacobian_sfun(t,x,u,flag,select)
% Jp=translational Jacobian matrix, Jr=rotational Jacobian matrix
% Jselect=1 return Jp, Jselect=2 return inv(Jp), Jselect=3 return Jp', otherwise return Jf
switch flag,
% Initialization %
case 0,
[sys,x0,str,ts]=mdlInitializeSizes;
% Outputs %
case 3,
sys=mdlOutputs(t,x,u,flag,select);

```

```

    case { 1, 2, 4, 9 }
    sys = []; % Unused flags
% Unexpected flags %
    otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
% end sfuntmpl

%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
function [sys,x0,str,ts]=mdlInitializeSizes

sizes = simsizes;

sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 3;
sizes.NumInputs = 6;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = [];
%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%
ts = [-1 0];

%
%=====
% mdlOutputs
% Return the block outputs.
%=====
%
function sys=mdlOutputs(t,x,u,flag,select)

L=microstruct([0;0;0;0;0],7);
lh=L(1); le=L(2); lf=L(3);

qv1=u(4);
qv2=u(5);
qv3=u(6);
qv23=qv2+qv3;

inp=[u(4); u(5); u(6)];

c1=cos(u(1));
s1=sin(u(1));
c2=cos(u(2));

```

```

s2=sin(u(2));
c3=cos(u(3));
s3=sin(u(3));
c23=cos(u(2)+u(3));
s23=sin(u(2)+u(3));

switch select
case 1
J=[-s1*(le*c2+lf*c23) -c1*(le*s2+lf*s23) -lf*c1*s23;
  c1*(le*c2+lf*c23) -s1*(le*s2+lf*s23) -lf*s1*s23;
  0 le*c2+lf*c23 lf*c23];
sys=J*inp;
case 2 %J'
TJ=[-s1*(le*c2+lf*c23) c1*(le*c2+lf*c23) 0;
  -c1*(le*s2+lf*s23) -s1*(le*s2+lf*s23) le*c2+lf*c23;
  -lf*c1*s23 -lf*s1*s23 lf*c23];
sys=TJ*inp;
case 3 %inv(J)
iJ=[-s1/(s1^2+c1^2)/(le*c2+lf*c23) c1/(s1^2+c1^2)/(le*c2+lf*c23) 0;
  c1*c23/(s1^2+c1^2)/(-s2*c23+s23*c2)/le s1*c23/(s1^2+c1^2)/(-s2*c23+s23*c2)/le
s23/(-s2*c23+s23*c2)/le;
  -c1*(le*c2+lf*c23)/(s1^2+c1^2)/(-s2*c23+s23*c2)/lf/le -s1*(le*c2+lf*c23)/(s1^2+c1^2)/(-
s2*c23+s23*c2)/lf/le -(le*s2+lf*s23)/(-s2*c23+s23*c2)/le/lf];
sys=iJ*inp;
case 4 %inv(J')
iTJ=[-s1/(s1^2+c1^2)/(le*c2+lf*c23) c1*c23/(s1^2+c1^2)/(-c23*s2+s23*c2)/le -
c1*(le*c2+lf*c23)/(s1^2+c1^2)/(-c23*s2+s23*c2)/le/lf;
  c1/(s1^2+c1^2)/(le*c2+lf*c23) s1*c23/(s1^2+c1^2)/(-c23*s2+s23*c2)/le -
s1*(le*c2+lf*c23)/(s1^2+c1^2)/(-c23*s2+s23*c2)/le/lf;
  0 s23/(-c23*s2+s23*c2)/le -(le*s2+lf*s23)/(-c23*s2+s23*c2)/le/lf];
sys=iTJ*inp;
case 5 %dJ
dJ=[-qv1*c1*(le*c2+lf*c23)+qv2*le*s1*s2+qv23*lf*s1*s23 qv1*s1*(le*s2+lf*s23)-qv2*le*c1*c2-
qv23*lf*c1*c23 qv1*lf*s1*s23-qv23*lf*c1*c23;
  -qv1*s1*(le*c2+lf*c23)-qv2*le*c1*s2-qv23*lf*c1*s23 -qv1*c1*(le*s2+lf*s23)-qv2*le*s1*c2-
qv23*lf*s1*c23 -qv1*lf*c1*s23-qv23*lf*s1*c23;
  0 -qv2*le*s2-qv23*lf*s23 -qv23*lf*s23];
sys=dJ*inp;
end
% end mdlOutputs

```

MICROSTRUCT.M

```

function out = microstruct(u,output)
%microstruct(x,output)
%u is link angles in radian
%output is selection of output values
%
%output numbers defines:
%0 -> ms (ms.m=m; ms.cs=cs; ms.cp=cp; ms.g=g; ms.b=b;)
%1 -> m ([m11 m12 m13 ; m21 m22 m23; m31 m32 m33])
%2 -> cs ([cs11 cs12 cs13; cs21 cs22 cs23; cs31 cs32 cs33])
%3 -> cp ([cp112 cp113 cp123; cp212 cp213 cp223; cp312 cp313 cp323])
%4 -> g ([g1; g2; g3])
%5 -> b ([b1; b2; b3])
%6 -> lm ([m1; m2; m3])
%7 -> ll ([lh; le; lf; r])
%8 -> inv(m)

c1=cos(u(1));
s1=sin(u(1));

```

```

c2=cos(u(2));
s2=sin(u(2));
c3=cos(u(3));
s3=sin(u(3));
c23=cos(u(2)+u(3));
s23=sin(u(2)+u(3));
s223=sin(2*u(2)+u(3));

qs=[u(4)^2; u(5)^2; u(6)^2];
qp=[u(4)*u(5); u(4)*u(6); u(5)*u(6)];

%link mass and length
m1=20; m2=12; m3=6;
lh=3; le=2; lf=2; r=1;

%inertia matrix
m=[26+29*c2^2+8*c23^2+24*c2*c23 0 0; 0 43+24*c3 8+12*c3; 0 8+12*c3 8];

%diagonal terms of coriolis matrix
cs=[0 0 0; 29*s2*c2+8*s23*c23+12*s223 0 -12*s3; 12*c2*s23+8*s23*c23 12*s3 0];

%non-diagonal terms of coriolis matrix
cp=[-58*s2*c2-16*s23*c23-24*s223 -24*c2*s23-16*s23*c23 0; 0 0 -24*s3; 0 0 0];

%gravitational term
g=[0; 196.2*c2+58.86*c23; 58.86*c23];

%frictional term
b=[0; 0; 0];

if output==0
    ms.m=m;
    ms.cs=cs;
    ms.cp=cp;
    ms.g=g;
    ms.b=b;
    out=ms;
elseif output==1
    out=m;
elseif output==2
    out=cs*qs;
elseif output==3
    out=cp*qp;
elseif output==4
    out=g;
elseif output==5
    out=b;
elseif output==6
    out=[m1; m2; m3];
elseif output==7
    out=[lh; le; lf; r];
elseif output==8
    out=inv(m);
end

```

CV

Hüseyin YALTIRIK was born on September 24, 1971 in Silvan districts of Diyarbakır city in Turkey. He completed his primary and high school education in Silvan and received her B.Sc. degree from the department of Mechanical Engineering at GaziantepUniversity in Gaziantep, Turkey, in 1997. He has started working as research assistant at the Department of Mechanical Engineering at Marmara University in 1998. He is still working as research assistant in the same department since 1999. He is married with two children.

MARMARA UNIVERSITY
THE INSTITUTE FOR
GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

ACCEPTANCE AND APPROVAL DOCUMENT

**COMPARISON OF CONTACT FORCE CONTROL STRATEGIES ON
DIFFERENT ROBOT ARM TYPES**

Established committee listed below, on 18.09.2006 and 2006/23-16 by the ‘*INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES*’ Executive Committee, have accepted Mr./~~Mrs./Miss~~ Hüseyin YALTIRIK ‘s Master of Science / ~~Doctor of Philosophy~~ thesis, titled as “Comparison Of Contact Force Control Strategies On Different Robot Arm Types” in Mechanical Engineering

COMMITTEE

Advisor : Prof. Dr. Abdülkerim KAR (Marmara University)
Member : Asst. Prof. Dr. Bülent EKİCİ (Marmara University)
Member : Asst. Prof. Dr. Haluk KÜÇÜK (Marmara University)



Date of thesis’ / ~~dissertation~~’s defense before the committee: 24.07.2007

APPROVAL

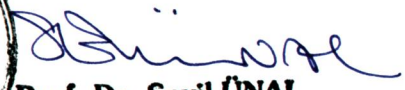
Mr. / Mrs. / ~~Miss~~ Hüseyin YALTIRIK has satisfactorily completed the requirements for the degree of ~~Doctor of Philosophy~~ / Master of Science in Mechanical Engineering at Marmara University.

Mr. / Mrs. / ~~Miss~~ Hüseyin YALTIRIK is eligible to have the degree awarded at our convocation on 30.07.2007, 2007/17-4.. Diploma and transcripts so noted will be available after that date.

Istanbul



DIRECTOR


Prof. Dr. Sevil ÜNAL
Müdür