Indian Journal of Science and Technology, Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/102274, December 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Comparison of Techniques for Disturbance-Tolerant Position Control of the Manipulator of *PUMA* Robot using PID

Seema Mittal[1*], M.P. Dave[2] and Anil Kumar[3]

[1]Ajay Kumar Garg Engineering College, Ghaziabad – 201009, Uttar Pradesh, India; simmiraj06@gmail.com
[2]Department Electrical Engineering, Shiv Nader University, NCR Delhi – 201308, India; davemp2003@yahoo.com
[3]Al Falah University, Dhauj, Faridabad –121004, Haryana, India; vcafuanil@gmail.com,

Abstract

Objectives: The control of the robotic manipulator arm under a variety of faults has been studied and the performance is compared using PID and other technique. **Methods/Statistical Analysis:** In a highly nonlinear environment such as manipulator of a robot, employing more than one control techniques yields desirable results. Here, a combination of PID along with pole-placement control of linear model has been designed. The feedback control gains have been obtained offline using equivalent linearization of the nonlinear coupled robot dynamic system. The input torque has been obtained from PID. The combined torque has been applied to the joints. This scheme has been implemented online in a standard PUMA manipulator with the payload. **Findings:** It has been observed that PID as compared to modified pole placement method is more efficient to control a robotic arm. **Application/Improvement:** The proposed hybrid control approach involving offline designs and their online implementation on six degrees of freedom robot has been found to be efficient and capable of accommodating common types of faults represented as an exponential or sine or a constant function but sudden or abrupt in nature.

Keywords: Fault-Tolerant Control, Hybrid Control, Linearization, PID, Pole Placement Control, *PUMA* Robot, Robotic Toolbox.

1. Introduction

Robots are ubiquities whether in industry or in health or other services of mankind. An important aspect of a robot manipulator is the precise placement of the target in space and can be achieved through appropriate control of the manipulator's arm. There are uncertainties associated with the model parameters of the robot such as PUMA 560. Further, the electro-mechanical components of the robot may develop a snag leading to a fault in any of the component thus affecting the normal functioning of the robot system. Therefore, an appropriate control technique is an integral part of the functioning of robots[1].

Robot manipulation is a complex activity as there is coupling among different terms in equation of motion i.e. rotation of one joint affects motion of other joints also. Therefore, under these conditions more than one fault diagnosis and control tool may be used. One approach is by employing duplicate sensors and actuators or in terms of provision of kinematic redundancy[2]. They demonstrated the efficacy of such scheme for maximizing post failure work space in a single locked joint situation. Using kinematic redundancy, the fault tolerance measure and its gradient were measured computationally and were correlated with the optimal reconfiguration of the robot. Another approach may be to use optimal or robust controller or a combination of them[3]. They have used two types of

*Author for correspondence

Artificial Neural Network (ANN) classifiers in the controller. The first neural network chosen was Multi-Layer Perception (MLP) with back propagation or Radial Basis Function (RBF). The performance was compared based on results of detection of localization. Another classifier used therein was a combination of RBF as well as MLP to arrive at the detection of the fault. However, for greater reliability for fault tolerance redundancy in the robotic system, a combination of control techniques called hybrid approaches have been proposed[1,4]. Other hybrid control technique involving Proportional-Integral-Derivative (PID) and fuzzy gain scheduling has been proposed[5] for control of industrial robot manipulator. Computed Torque Control (CTC) along with the Fuzzy Control (FC) was applied in their study for the trajectory control of robot manipulator. This method[6] does not require precise dynamical model and was able to control even under unstructured uncertainty. A hybrid neuro-fuzzy controller has been proposed[7] for controlling two arms of a robot by simulating the movements of a claw. The PID tuning has also been reported where its coefficients were computed based on optimization of certain cost function[8]. The neural network factors were optimized using Particle Swarm Optimization (PSO). However, this method involves large computational efforts and may not be desirable in case of large d.o.f. systems having constraints such as in PUMA robot. PID controller having single or two degree of freedom (d.o.f.) which was tuned using Genetic Algorithm (GA) has been employed in a reheat thermal system[9]. They observed that two d.o.f. PID controller provided improved transient responses.

Linearisation of a nonlinear dynamic system remains an interseting field of controls. A recent study[10] shows that the precise linearised model of a nonlinear system can be achieved by using a high dimensional latent variable space and using Bond-Graph model. The propsoed methodolgy has some resemblance with subspace methods for system identification but for nonlinear system. The methodology was demonstrated using a mass-spring-damper system. The authors reported that the methodology was already implemented in some biological systems, impedence control and wearbale robot control problems by this group.

In the present study, the control of the robotic arm has been implemented using PID and pole placement control methodology. A standard robot namely PUMA 560 having six d.o.f. has been studied. The constraints of angular rotations, velocity and accelerations due to the capacity of motors as applicable to this robot have been imposed in its numerical model. A combination of two control approaches has been employed. The torque applied to a joint consists of two components namely input and feedback control. The input control is assessed using PID. The feedback control parameters have been computed offline using linearized state space formulation. The control scheme is then implemented online on a robot. The proposed methodology helps achieve positioning the arm at the target more efficiently. The optimum position control has been achieved under a hybrid control and the methodology has been demonstrated on achieving target position of the manipulator using motion of all six joints of the PUMA560 robot arm when one of them develops some fault or disturbance.

2. Description of PUMA Robot

2.1 Engineering Parameters of Robot

PUMA 560 is a standard robot having six d.o.f. systems. It consists of six arms called links which are connected through six joints. The first three joints are called shoulder, elbow and wrist joints are considered as shown in Figure1 (adapted from[11]). The joints 4, 5 and 6 help achieving proper orientation of the end effect or to hold an object in a desired manner.

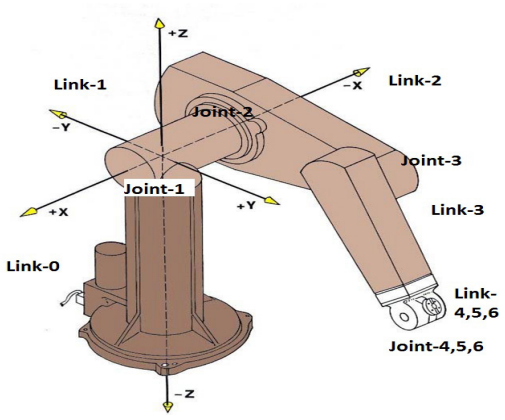

Figure 1. Schematic of Puma Robot[11].

The engineering parameters of the standard PUMA 560 robot have been considered[12]. The actuator's physical limits i.e. motor responses can be considered as the bounds imposed by capacity of the actuators and are provided in Table 1.

Table 1. Actuator physical limits.

Parameter for Motor	Type of Motor					
	PR090	PR090	PR090	PR070	PW0701	PW0702
Rotation (deg) [rad]	±320 [±5.585]	±250 [±4.363]	±270 [±4.712]	±300 [±5.236]	±200 [±3.491]	±532 [±9.285]
Velocity (deg/s), [rad/s]	149 [2.6]	149 [2.6]	149 [2.6]	149 [2.6]	248 [4.33]	320 [5.58]
Acceleration (deg/s²), [rad/s²]	596 [10.402]	596 [10.402]	596 [10.402]	596 [10.402]	992 [17.314]	1280 [22.340]
Current (Amp@24V)	30	30	30	15	15	15
Output torque (Nm)	206	206	206	73	54	28
Used in joint	1	2	3	4	5	6

2.2 Dynamic Modeling of Robot Manipulator

Considering the geometric and other parameters, the dynamic equation of motion of the robot can be expressed as in equation 1.

$$M\ddot{q} + Vm(q,\dot{q})\dot{q} + \chi(q) + F(\dot{q}) = \tau \quad (1)$$

Here, $q = [q_1, q_2, q_3, q_4, q_5, q_6] \in \mathcal{R}^n$ are the joint positions ($n = 6$, equal to the d.o.f. of the robot system), $M(q) \in \mathcal{R}^{nxn}$ is the inertia matrix and is a symmetric positive definite matrix, $Vm(q,\dot{q}) \in \mathcal{R}^{nxn}$ represents Coriolis and centripetal forces. $F(\dot{q}) \in \mathcal{R}^n$ is the dynamic frictional force matrix, $\chi(q) \in R^n$ is the gravity matrix and $\tau \in \mathcal{R}^n$ denotes generalized input control of the system applied at the joints. The simulation of functional aspects of the PUMA 560 robot such as kinematics, dynamics and trajectory generation have been carried out using Robotics Toolbox[13] with some modifications. This has been used to generate responses namely q $\dot{q}$ $\ddot{q}$ by solving dynamic equations of motion (without friction) using Recursive Newton Euler (RNE) method.

2.3 Development of Equivalent Linear Model

A lot of computational efforts can be simplified if the equations of motion are set in closed forms or are linear in nature. The complexities in the modeling may be attributed to large variations in inertia terms as values of the joint angles vary during desired motion of the arm. The values of inertia at various positions of the joint angles i.e. q_2 , q_3 have been computed and its variation is shown in Figure 2.

A few equivalent linear models of PUMA 560 are proposed[14-16]. However, the formulation[16] incorporates modified parameters and a set of corrected terms in inertia as well as Coriolis terms and the same has been adopted in the present study. The suggested linearization of nonlinear dynamic equations uses the Taylor series expansion of nonlinear functions about a nominal trajectory after neglecting higher order terms (retaining first order term) and are expressed for a function, f as follows.

$$f(q,\dot{q},\ddot{q}) = f(q^*,\dot{q}^*,\ddot{q}^*) + \left\{\frac{\partial f}{\partial q}\Big|_{q^*}\right\}\delta q + \left\{\frac{\partial f}{\partial \dot{q}}\Big|_{\dot{q}^*}\right\}\delta\dot{q} + \left\{\frac{\partial f}{\partial \ddot{q}}\Big|_{\ddot{q}^*}\right\}\delta\ddot{q} \quad (2)$$

The equation 2 can also be written in a linear form as given in equation 3 as applicable to the robot dynamics.

$$\delta\tau = M_1(q^*)\,\delta\ddot{q} + C_0(q^*,\dot{q}^*)\,\delta\dot{q} + K_1(q^*,\dot{q}^*,\ddot{q}^*)\,\delta q \quad (3)$$

Here q^* denotes the nominal trajectory. $M_0, C_0, K_0 \in \mathcal{R}^{nxn}$ are linearized trajectory sensitivity matrices in terms of the nominal trajectory. The expression for matrices M_0, C_0, K_0 is given in Appendix 1. It may be noted that the inertia matrix is not singular in case of the manipulator dynamics under consideration. The equation 3 can be formulated in to state space form as shown in equation 4.

$$\frac{d}{dt}\begin{bmatrix}\partial q\\ \partial \dot{q}\end{bmatrix} = \begin{bmatrix}0 & I\\ -M_0^{-1}K_0 & -M_0^{-1}C_0\end{bmatrix}\begin{pmatrix}\delta q\\ \delta \dot{q}\end{pmatrix} + \begin{bmatrix}0\\ -M_0^{-1}\end{bmatrix}(\delta\tau) \quad (4)$$

Relevant matrices are formulated by comparing equation 4 with the standard state space form (equation 5) where, $x(t)$ is the state vector consisting of $[q\dot{q}]^T \in \mathcal{R}^{2n}$. The input control vector is $u(t) \in \mathcal{R}^{n}$, $y(t) \in \mathcal{R}^{2n}$ is the output vector and $C(t) \in \mathcal{R}^{2nx2n}$ is the output matrix.

$$\dot{x} = Ax + Bu \text{ and, } y = Cx + D \quad (5)$$

The matrices of the state space system (G) as depicted in Figure 3 constitutes matrices A, B, C, D and can be expressed as given below.

$$A = \begin{bmatrix}0 & I\\ -M_0^{-1}K_0 & -M_0^{-1}C_0\end{bmatrix};\ B = \begin{bmatrix}0\\ -M_0^{-1}\end{bmatrix};\ C = \text{diag}([1,1,1,1,1,1,1,1,1,1,1,1]) \text{ and } D=[0]. \quad (6)$$

The net applied torque to the system would be a combination of the nominal incremental torque given by the equation 3 and the feedback torque which can be computed based on an appropriate control technique. The additional term in the form of feedback (gain) effectively takes care of any deviation from nonlinearity while considering an equivalent linear system. This ensures that the system is close to the nominal trajectory point at which linearization has been adopted. It may be noted that even after linearization the equations still remain coupled and are complex in nature.

3. Uncertainties and Faults

The uncertainties in the model arise from non-availabilities of precise geometric and elastic parameters of a robot. Assessment of precise relevant data remains a topic of research[17,18]. Further, dynamic response parameters vary when the values of variables changes during the course of achieving the desired trajectory. The inertia as felt at the first joint with variations in the value of joint rotations q_2 and q_3 is computed and is shown in Figure 2. Large variations are clearly visible and the corresponding values of q_2 and q_3 for two extreme values of resulting inertia and at the midpoint is given in Table 2. There is negligible influence from other joint rotations. A control scheme should be able to accommodate these uncertainties.

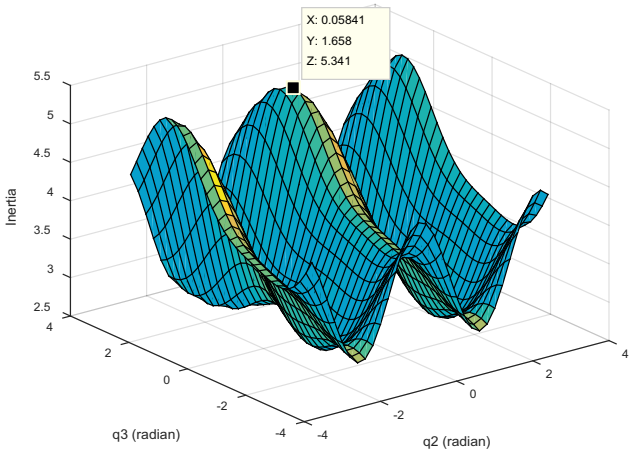

Figure 2. Variation in inertia as q_2 and q_3 varies.

Table 2. Typical position of joint angles as inertia at joint-1 varies.

Ordinate	High Value	Medium Value	Low Value
q^2 (rad)	0.0584	0.0584	-1.742
q^3 (rad)	1.658	-2.142	2.058
Inertia	5.341	3.793	2.67

Further, during operation of a robot, a fault may get developed in any of the component of the system either in the sensor or in the actuator. The resulting changes in the modeling parameters will be different in these two cases. The present study considers the failure of actuators only. To account the influence of faults the equation of motion has an additional term representing fault[19] and can be written as

$$\ddot{q} = M_0^{-1}(q)\ [\ \tau - V_m(q, q^{\cdot})\, q^{\cdot} - \chi(q) - F(\dot{q}) - \tau_d(t)\ ] + \beta(t-T)\,\tilde{\Psi}(q, q^{\cdot}, \tau) \quad (7)$$

The fault is represented by $\beta(t-T)\tilde{\Psi}(q, \dot{q}, \tau)$ in which T is the time of occurrence of the fault, $\tilde{\Psi}$ is a fault function and contains terms due to angular position, velocity and torque. The value of states of robotic parameters $(q, \dot{q})$ is assumed to be bounded. Also, the uncertainties due to faults are considered to be finite in magnitude and are expressed as a:

$$|M_0^{-1}(q)\,(\tau_d)| \le \mu, \quad (8)$$

Here, μ > 0 and is a known constant, which can be realized as limits of parameters imposed on the system and are shown in Table 1. For simulation, the trajectory to be simulated in terms of q vector is from an initial point described by $q_1 1i\ in\ radians = (1, 1, 1, 1, 1, 1)$, for $i = 1\ to\ 6,$ to a target point q_{2i} (1.5, 3, 4, 1.5, 1.5, 1.5) considering rotations of all the six joints. The desired trajectory (q_d) is a seventh degree polynomial[13] fitted between the mentioned two points at a time step (Δt) of 0.056 seconds. In the present simulation, three types of actuator faults mentioned as below are studied.

3.1 Case 1. Lock-in Fault in Actuator

The actuator failure takes place at a joint in which a constant torque is exerted on the system such as at q_2 the torque, $\tau_2 = \pm 40$, at a time period 1.5 ≤ t ≤ 2.5 seconds Similar types of fault was simulated successfully at other joints also.

3.2 Case 2. Sinusoidal Actuator Failure

In this case, the faulty actuator imposes sinusoidal type of torque representing a time varying actuation[20] at joint-2 described as, $\tau_2 is\ \pm 40\ \sin(0.3t)$, at a time period 1.5 ≤ t ≤ 2.5 seconds

3.3 Case 3. Exponential Actuator Failure

In this case, the faulty actuator imposes exponential type of torque representing a time varying actuation at joint-2 described as, $\tau_2 = \pm 40 e^{-0.5t}$, at a time period 1.5 ≤ t ≤ 2.5 seconds

4. Controlling of Robot System

4.1 Design of Controller

The controllers using close loop feedback control system as shown in Figure 3 are designed for positioning the end effect or at desired position and orientation.

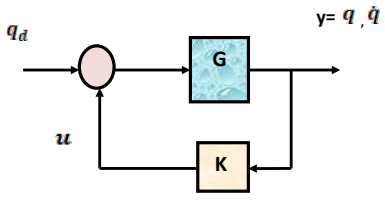

Figure 3. Feedback control of the system G.

The following assumptions regarding the modeling are considered.

1. The initial state of the svstem $x(0)$ is available.
2. The system states $q, \dot{q}$ remain bounded even after occurrence of a fault such that $\{q, \dot{q}\} \in \Omega_{q,}$ where Ω_q is the (finite) region of operation.
3. The capacity of the load (load disturbance) is bound such that the desired (nominal) torque $\|\tau_d\| \leq \tau_m$ remains within certain bounds which may be interpreted as load carrying capacity of the robot ($\tau_1 m$) and its value is known. The tracking error also called residuals is defined as $e(t) = q_d(t) - q(t)$, and its derivative as $\dot{e} = (\dot{q}_d - \dot{q})$, here $q_d(t)$ is the desired trajectory. Based on this formulation the control is designed offline by getting K -gains from a particular technique. These gains are adopted to provide control input torque to the robot system online thus simulating a predefined trajectory of the end effect or. The control techniques are described in the next section.

4.2 Proportional–Integral–Derivative Controller (PID)

In the feedback control mechanism, the part of the output is fed into the system so that the errors get reduced. The plant having input 'u' and output 'y' is described by the model 'G' and the gain by 'K' (Figure 3). An error vector is computed by comparing the observed output and the desired output of joint rotations. The parameters in PID controller are chosen such that the error, $e(t)$ get vanishes in certain finite time. In general, there are three components of a PID controller namely proportional, integral and derivative terms. These terms consist of coefficients denoted as $K_P, K_I\ and\ K_D$ which when multiplied with the error term, integral of error and the derivative term of the error respectively, give feedback gain to the system to be controlled. The feed gain matrix of PID in time domain is expressed as in Equation 9.

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt} \quad (9)$$

Here, proportional term signifies the present value of the error, integral term accounts for the previous value of the error, and derivative term accounts for the future changes by adopting the gradient of the error. The integral term may also be seen as the accumulative effect of the

error. This method does not guarantee the optimal control. However, it may provide good solution in certain situations. In fact, it is not necessary that all the three terms are used which may be adopted depending upon the severity of the control problem as it involves the cost of computation. It may be noted that the derivative term is sensitive to measurement noise. Also, the absence of an integral term may leave a gap between the desired and achieved output values.

4.3 Pole Placement Control

It is a time domain control technique. The technique of pole placement control involves computing state feedback gain matrix (K_g) by choosing poles of the closed loop system. The control signal (u) is given by equation 10. The gain matrix K_g consists of two components matching with q and $\dot{q}$ as:

$$u = -K_g\, x \text{ , and } K_g = [K_{ge}, K_{ged}]. \quad (10)$$

In the present case, $A \in \mathcal{R}^{2nx2n}$, $x \in \mathcal{R}^{2nx2n}$ and $K_g \in \mathcal{R}^{2nx2n}$.The necessary condition to satisfy is that the system should be state controllable and all the state variables are measurable. The poles of the matrix $(A - BK_g)$ can be arbitrarily chosen so as to have desirable performance cost resulting in desired level of transient or frequency response[21]. One way of choosing poles is to place dominant poles based on root locus design and remaining poles to be much farther to the left i.e. negative real, which are normally a pair of complex conjugates[21]. If the dominant poles are far (left) from jw axis, the response will be faster, however, signals will also be large in magnitude and may increase non-controllability in the system. The response parameters may be the magnitude of overshoot in the response, rising time, settling time, bandwidth (in Bode plot) of the system. It may be noted that the large bandwidth in the system may induce noise. The complex poles considered for the pole placement for getting the control gains have negative real component and are chosen appropriately as described in the next section.

5. Results and Discussion

5.1 Influence of Uncertainties and Payload

The robot arm moves under gravity from a position defined as q_{1i} to q_{2i} (Section 3) with payload but no applied torque. The achieved trajectory defined in terms of angular positions traversed during movement of the arm for the first three joints in this situation are shown in Figure 4 as a reference (the dashed line is desired and solid line is achieved trajectory).

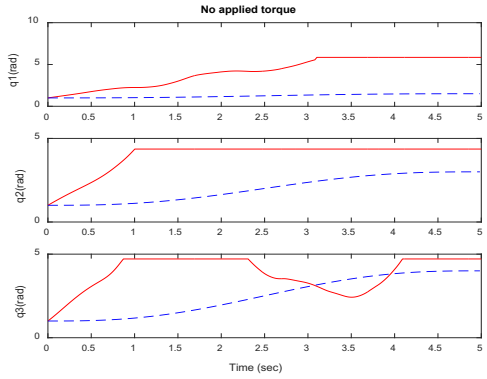

Figure 4. Observed trajectory without applied torque.

The limits of the payload[22] of PUMA560 are 2.5 kg to 4 kg. The torque requirement for the trajectory to be achieved for different applied payload has been observed to be maximum for the joint 2. The case of no payload represents the torque demand due to the effect of gravity and uncertainties in modeling. The cases of payload incorporates effect of the additional values of 2.5 kg or 4 kg as applied vertically at the end of the end effectors and the variation in torque demand at the joint 2 is shown in Figure 5.

5.2 Control using PID

The values of coefficients of PID as per section 4.2 are iteratively computed which are diagonal matrices as given below:

$$K_p = (450, 1200, 200, 120, 120, 120),$$
$$K_I = (300, 1200, 50000, 500, 500, 500), \text{ and}$$
$$K_D = (-20, -200, -20, -50, -50, -50).$$

The feedback control has been applied for the three cases of the fault. During the reported simulations a payload of 2.5 kg has been applied at the end effectors. The torque demand for six joints is shown in Figure 6 when a fault (case-1) occurs at the joint 2.

It may be observed in Figure 6 that at some instances the joint reaches its torque capacity which then reduces as per the level of the corresponding residual. The variation of residuals for the first three joints (e_1, e_2, e_3) during the operation of the robot manipulator is shown in Figure 7 (the influence of $\dot{e}$ has not been shown here). It may be observed in this figure that the residuals reach asymptotically zero at the end of the simulation.

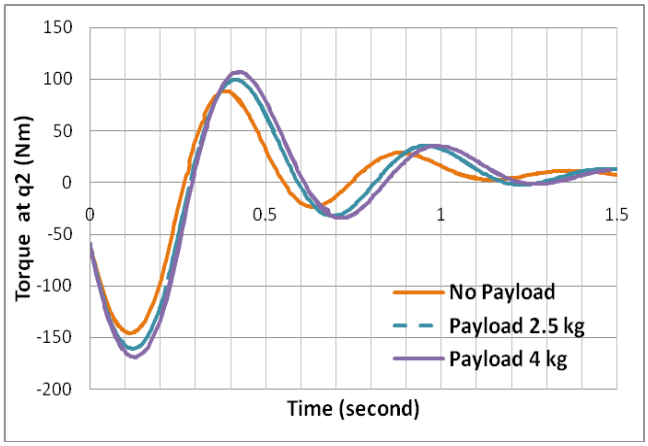

Figure 5. Observed variation in torque demand at joint-2 due to payload.

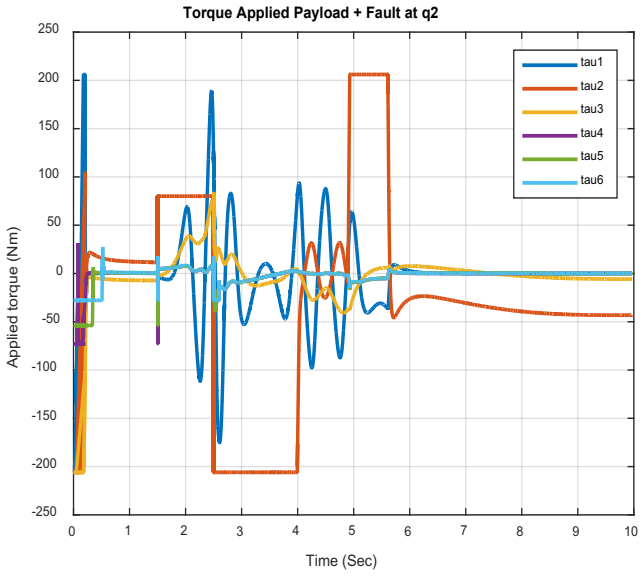

Figure 6. Torque demand at joints.

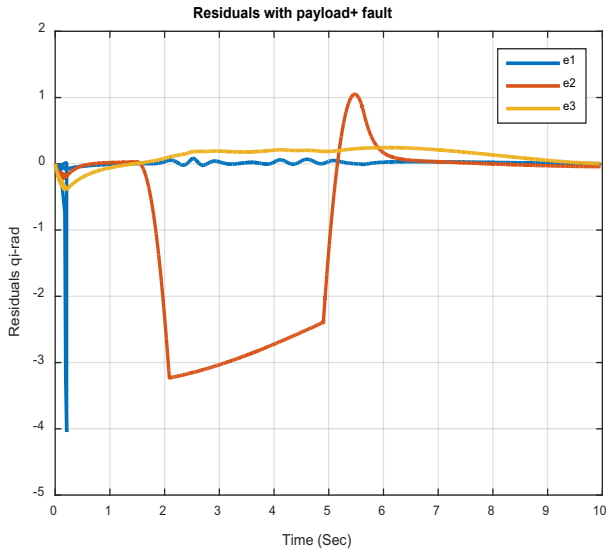

Figure 7. Variation of first three residuals.

The resulting trajectory for the induced fault case 1 in the joint 2 is shown in Figures 8 and 9. It may be observed that the position control of the end effectors has been simulated successfully. In Figure 10, the disturbance caused by the fault between 1.5 sec and 2.5 sec is clearly visible and makes the arm to oscillate for a short duration before stabilizing (dashed line is desired trajectory).

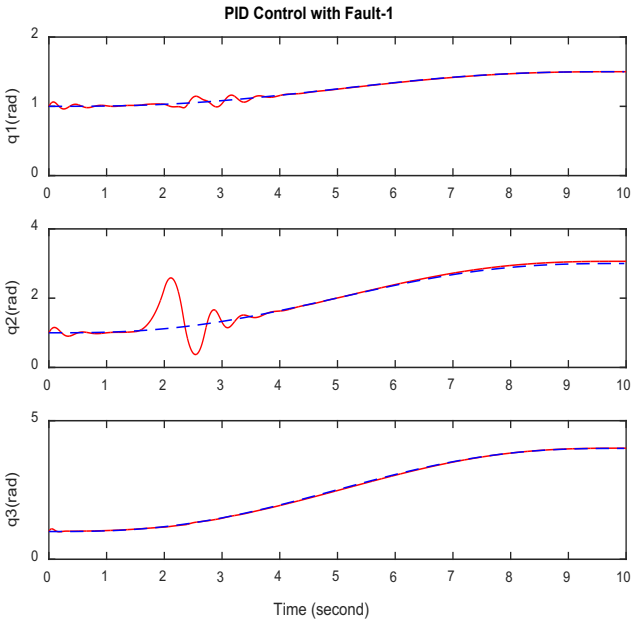

Figure 8. PID control under fault at $q_{1\text{-}3}$ (fault Case-1).

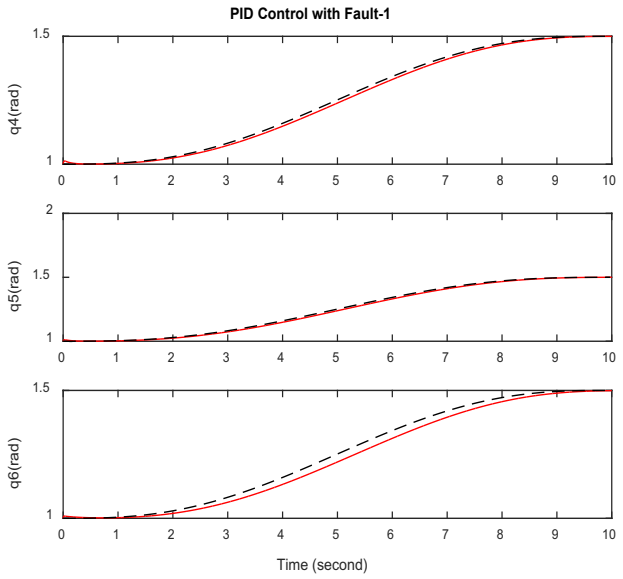

Figure 9. PID control under fault at $q_{4\text{-}6}$ (fault Case-1).

5.3 State Space Modeling

The central point as defined in Section 3 is taken as a point (q) with mid-value of inertia (Table 2) and $\dot{q}, \bar{q}$ as higher value from Table 1, which are given:

$q^* = (1.5, 0.0584, -2.142, 1.5, 1.5, 1.5);$

$\dot{q}^* = (2.6, 2.6, 2.6, 2.6, 4.3, 5.5);$ and

$\bar{q}^* = (10.4, 10.4, 10.4, 10.4, 17.3, 22.3).$

The resulting state space matrices $M_0, C_0, K_0 \in \mathcal{R}^{nxn}$ (here, n = 6, based on modeling discussed in Section 3) and their values as computed are given in Appendix 2. Matrices A, B, C, D may be constructed from Equation 6. The Eigen values (poles) of matrix A , that is an unstable condition and after applying control i.e. $(A - BK_g)$ for the considered control approach have been computed and are shown in Figure 10. The poles of unstable A have some positive values and lies in the range of +0.85 to -0.089 and remaining are of zero values. The complex poles of pole place technique have been chosen with a range of pole place-1 as -3.6 to -0.5/ 0.0 and pole place-2 in the range of -10.26 to -0.82/ 0.0. The control gains obtained for pole place-2 does not provide satisfactory simulation. Further, another set of poles are chosen as pole place-1 and the feedback control gains are reported below.

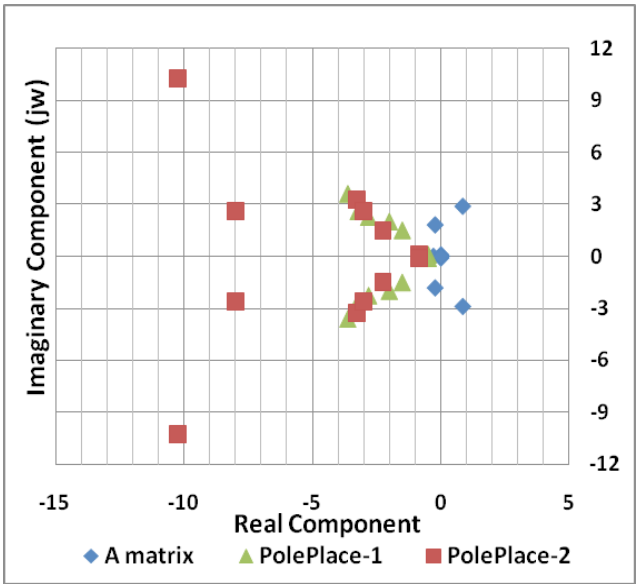

Figure 10. Observed location of poles before and after control.

The gains by pole placement as given in expression 10based on pole place-1are provided below:

Kge=
[15.0872 25.4436 -10.4161 -1.0873 14.3841 6.1963;
-5.5871 -12.4523 12.8511 11.1151 -4.3217 9.0817;
-1.9839 -4.9308 9.4352 1.5359 -2.3569-0.2036;
-0.0182 0.5148 0.0506 1.6620 -0.0034 -0.1272;
0.3280 -0.0577 -0.2513 0.2969 1.7653 0.0344;
0.2188 0.1709 -0.2671 -0.1084 0.0020 1.7023];

Kged=
[20.6373 11.9515 -1.0261 -4.2327 4.3033 0.2580;
-13.9627 19.2503 8.1597 1.2187 3.3442 2.9022;
-2.8805 -1.0248 5.8981 -0.2449 -0.1335 -1.4641;
0.2062 0.0741 0.1425 0.8941 0.0855 -0.0724;
-0.0241 -0.0927 -0.0444 -0.0225 0.8858 0;
-0.0153 -0.0216 0.1413 0.0805 0.0362 0.8977];

Kgei=diag([-43000 -3300 -5950 -3900 -4000 -700]).

In this case, the input torque is assessed from integral term (Kgei) of PID and the feedback torque based on equation 10. The resulting position of the end effector has been achieved as shown in Figures 11 and 12 for all the six joints.

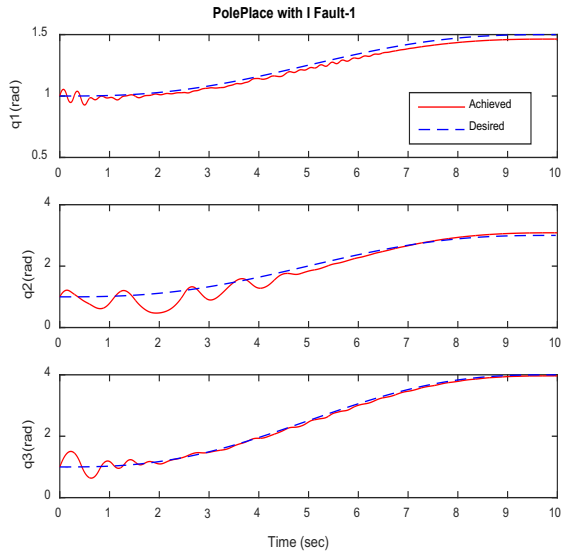

Figure 11. Simulated trajectory by pole placement and integral term.

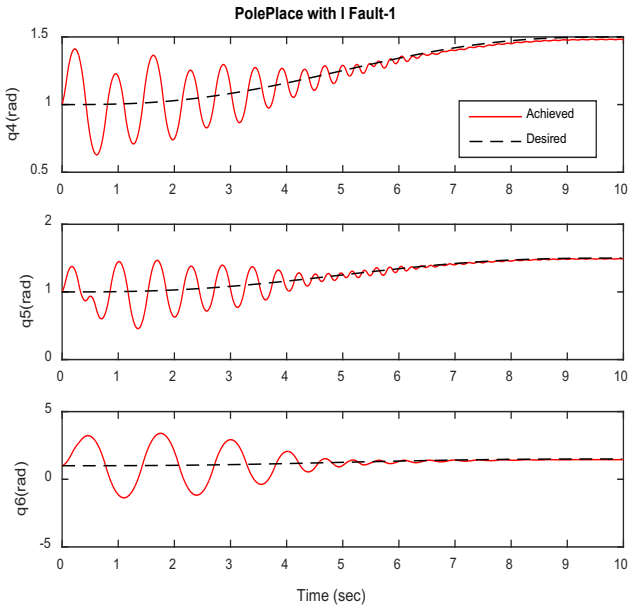

Figure 12. Simulated trajectory by pole placement and integral term.

It may be observed in Figures 8-9 and 11-12 that PID simulates rather smooth trajectory compared to pole placement control. It may be noted that the time taken during simulation of the position in case of pole placement is smaller compared to the case of PID control[23].

6. Conclusions

A hybrid control framework has been proposed for controlling a standard robot arm of PUMA560 which has PID along with pole placement control. The feedback control utilizes the control gains obtained from an offline linearized model of the robot considered. The input control can be based on PID. Alternately, pole placement with integral term (of PID) can be used to control the arm. Based on relative performance of achieving target position, it has been observed that PID tracks smoothly the trajectory thus achieving precisely the position of the end effectors compared to the pole placement technique in the presence of uncertain actuator failure.

7. Acknowledgement

The first author acknowledges the kind support provided by Dr. R.K. Agarwal, Director and Dr. P.K. Chopra, HOD, ECE at Ajay Kumar Garg Engg. College, Ghaziabad and Dr. Tasleem Burney, Ph.D. Coordinator at Al Falah University, Dhauj, India.

8. References

1. Gao Z, Cecati C, Ding SX. A Survey of Fault Diagnosis And Fault-Tolerant Techniques- Part II: Fault Diagnosis with Knowledge Based and Hybrid/ Active Approaches, IEEE Trans. on Industrial Electronics. 2015; 62(6):3768–74.
2. Kenneth NG, Maciejewski AA, Balakrishnan V. Real-Time Failure-Tolerant Control of Kinematically Redundant Manipulators, IEEE Trans. on Robotics and Automation. 1999; 15(6):1109–16.
3. Zilong S, Gang Z, Denis E, Wilfrid P. Modelling and Control for Position-Controlled Modular Robot Manipulators. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, p. 3290–95.
4. Wu E, Hwang J, Chladek J. Fault Tolerant Joint Development for the Space Shuttle Remote Manipulator System: Analysis and Experiment, Proceedings 4th International Symposium on Robotics Manufacture, (ISRAM '92), Sante Fe. 1992; 9(5):505–10.
5. Lei SY, Joo M. Hybrid Fuzzy Control of Robotics Systems, IEEE Transaction on Fuzzy Systems. 2004, 12(1):755–65.
6. Song Z, Yi J, Zhao D, Li X. A Computed Torque Controller for Uncertain Robotic Manipulator Systems: Fuzzy Approach, Fuzzy Sets Systems, Elsevier. 2005; 154(2):208–26.

7. Khoobjou E. Controlling the Arm of Flexible Robot by using New Hybrid Approach, Indian Journal of Science and Technology. 2016; 9(7):1–5.
8. Bansal HO, Sharma R, Shreeraman PR. PID Controller Tuning Techniques: A Review, Journal of Control Engineering and Technology (JCET). 2012 Oct; 4(2):168–76.
9. Meena AR, Senthil KS. Design of GA Tuned Two-degree Freedom of PID Controller for an Interconnected Three Area Automatic Generation Control System, Indian Journal of Science and Technology. 2015; 8(12):1–10.
10. Asada HH, Faye W, Girard A, Mayalu M. A Data-Driven Approach to Precise Linearization of Nonlinear Dynamical Systems in Augmented Latent Space. Proceeding American Control Conference, Boston, USA, 2016 Jul, p. 1838–44.
11. Jerry Rutherford, Using The PUMA 560 Robot. Date accessed: 01/09/2015. Available at: http://rutherford-robotics.com/puma/using%20the%20puma%20robot.pdf.
12. Corke PI, Armstrong HB. Search for Consensus among Model Parameters Reported for the PUMA 560 Robot, Proc. IEEE International Conference, Robotics and Automation, San Diego: USA, 1994 May 2, p. 1608–13.
13. Corke P. Robotics, Vision and Control, Springer 2011.
14. Swarup A, Gopal M. Linearized Dynamic Models for Trajectory Control of Robot Manipulators, Journal of Intelligent and Robotic Systems. 1993; 7(3):287–300.
15. Lieh J. Computer Oriented Closed Form Algorithm for Constrained Multi-body Dynamics for Robotics Applications, Mechanism and Machine Theory. 1994; 29(3):357–71.
16. Clover CL. Control System Design for Robots used in Simulating Dynamic Force and Moment Interaction in Virtual Reality Applications, Retrospective Theses and Dissertations. Paper No. 11141, 1996; p. 1-185.
17. Yan D, Lu Y, Levy D. Parameter Identification of Robot Manipulators: A Heuristic Particle Swarm Search Approach, PLoS ONE. 2015; 10(6):129–57.
18. Acosta L, Marichal GN, Moreno L, Rodrigo JJ, Hamilton A, Mendez JA. A Robotic System Based on Neural Network Controllers, Artificial Intelligence in Engineering. 1999; 13(4):393–98.
19. Vemuri AT, Polycarpou MM. Neural-Network-Based Robust Fault Diagnosis in Robotic Systems, IEEE Transactions on Neural Networks. 1997; 8(6):1410–20.
20. Rugthum T, Tao G. An Adaptive Actuator Failure Compensation Scheme for a Cooperative Manipulator System with Parameter Uncertainties, IEEE 54th Annual Conference, Decision and Control, Osaka: Japan, 2015Dec, p. 6282–87.
21. Katsuhiko O. Modern Control Engineering. Fifth Edition, PHI/Pearson, 2012.
22. Robot 560 C Arm Manual {C340.005.05.A}. Staubli Unimation Ltd. 1990, Quoted in Programmable Universal Machine for Assembly-Model 560 C. Date accessed 01/09/2015. Available at: www.en.wikipedia.org.
23. Mittal S, Dave MP, Kumar Anil. Fault-Tolerant Position Control of the Manipulator of *PUMA* Robot using Hybrid Control Approach, International Journal of Current Engineering and Technology. 2016; 6(5):1818–25.

Appendix 1: Linearised Model

The matrices M_0, C_0 and K_0 are of size 6X6. The value of elements of these matrices is to be set to zero whose expression is not provided below. The matrix M_0 is symmetric (set M_0 (J,I)=M_0 (I,J), for I=1:6; for J=1:6).

% M_o INERTIA MATRIX

M_0 (1,1)=3.64+0.8*cos(2*q_2) -0.1*sin(2*q_3 +2*q_2) -0.01*cos(q_3 +2*q_2) +0.37*sin(q_3 +2*q_2) -0.15*cos(2*q_3 +2*q_2) +0.37*sin(q_3) -0.01*cos(q_3);
M_0 (1,2)=0.69*sin(q_2)-0.13*cos(q_3 +q_2)+0.02*cos(q_2);
M_0 (1,3)=-0.13*cos(q_3 +q_2);
M_0 (2,2)=4.4-0.02*cos(q_3)+0.74*sin(q_3);
M_0 (2,3)=0.33-0.01*cos(q_3)+0.37*sin(q_3);
M_0 (3,3)=1.16;
M_0 (4,4)=0.20;
M_0 (5,5)=0.18;
M_0 (6,6)=0.19;

% C_o JOINT VELOCITY SENSITIVITY

C_0 (1,1)={-1.6*sin(2*q_2)+0.3*sin(2*q_3 +2*q_2)+ 0.74*sin(q_3 +2*q_2)-0.02*cos(2*q_3 +2*q_2) +0.02*sin(q_3 +2*q_2)-0.01*cos(2*q_2)}*$\dot{q}_2$+ {0.37*cos(q_3)+0.37*cos(q_3 +2*q_2)+0.3*sin(2*q_3 +2*) -0.02*cos(2*q_3 +2*q_2)+0.01*sin(q_3) +0.01*sin(q_3 +2*q_2)}*($\dot{q}_3$);
C_0 (1,2)=(-1.6*sin(2*q_2)+0.3*sin(2*q_3 +2*q_2)+ 0.74*cos(q_3 +2*q_2)-0.02*cos(2*q_3 +2*q_2) +0.02*sin(q_3 +2*q_2)-0.01*cos(2*q_2)*$\dot{q}_1$)+ (1.38*cos(q_2)+0.27*sin(q_3 +q_2)-0.05*sin(q_2)*$\dot{q}_2$ +(0.27*sin(q_3 +q_2)*$\dot{q}_3$);
C_0 (1,3)=(0.37*cos(q_3)+0.37*cos(q3+2*q_2)+0.3*sin(2*q_3 + 2*q_2)-0.02*cos(2*q_3 +2*q_2)+0.01*sin(q_3)+ 0.01*sin(q_3 +2*q_2)*$\dot{q}_1$ +(0.27*sin(q_3 +q_2)*$\dot{q}_2$) + (0.27*sin(q_3 +q_2)*$\dot{q}_3$;
C_0 (2,1)=(1.6*sin(2*q_2)-0.3*sin(2*q_3 +2*q_2)-0.74*cos(q_3 + 2*q_2)+0.02*cos(2*q_3 +2*q_2)

-0.02*sin(q_3 +2*q_2)+0.01*cos(2*q_2)*$\dot{q}_1$;
C_0 (2,2)=(0.74*cos(q_3)+0.02*sin(q_3)*$\dot{q}_3$);
C_0 (2,3)=(0.74*cos(q_3)+0.02*sin(q_3)*$\dot{q}_3$)
+(0.74*cos(q_3)
+0.02*sin(q_3)*$\dot{q}_2$);
C_0 (3,1)=(-0.37*cos(q_3)-0.37*cos(q_3 +2*q_2)-0.3*sin(2*q_3 +
2*q_2)+0.02*cos(2*q_3 +2*q_2)
-0.01*sin(q_3)-0.01*sin(q_3 +2*q_2)*$\dot{q}_1$);
C_0 (3,2)=(-0.74*cos(q_3)-0.02*sin(q_3)*$\dot{q}_2$);

% K_0 JOINT POSITION SENSITIVITY

K_0 (1,2)=(-1.6*sin(2*q_2)+0.74*cos(q_3 +2
*q_2)+0.3*sin(2*q_3 +
2*q_2)-0.02*cos(2*q_3 +2*q_2)+0.02*sin(q_3 +2*q_2)-
0.01*cos(2*q_2)*$\bar{q}_1$ +(0.69*cos(q_2)+0.13*sin(q_3 +
q_2)-0.02*sin(q_2)*$\bar{q}_2$ +(0.13*sin(q_3 +q_2)*$\bar{q}_3$ +
(-3.2*cos(2*q_2)-1.5*sin(q_3 +2*q_2)+0.6*cos(2*q_3 +2*q_2)+
0.04*cos(q_3 +2*q_2)+0.04*sin(2*q_3 +2*q_2)+
0.03*sin(2*q_2))*$\dot{q}_1$)*+(-0.74*sin(q_3 +2*q_2)+
0.6*cos(2*q_3 +2*q_2)+0.04*sin(2*q_3 +2*q_2)+
0.02*cos(q_3 +2*q_2))*$\dot{q}_1$)*$\dot{q}_3$)
+(0.27*cos(q_3 +q_2))*$\dot{q}_2$)$\dot{q}_3$)+(-0.69*sin(q_2)+
0.13*cos(q_3 +q_2)-0.02*cos(q_2))*$\dot{q}_2$)*$\dot{q}_2$)+
(0.13*cos(q_3 +q_2))*$\dot{q}_3$)$\dot{q}_3$)
K_0 (1,3)=(0.370*cos(q_3)+0.37*cos(q_3 +2*q_2)+
0.3*sin(2*q_3 +2*q_2)-0.02*cos(2*q_3 +2*q_2)
+0.01*sin(q_3 +2*q_2)+0.01*sin(q_3))*$\bar{q}_1$ +
(0.13*sin(q_3 +q_2))*($\bar{q}_2$ +$\bar{q}_3$)+
(-0.74*sin(q_3 +2*q_2)+0.6*cos(2*q_3 +2*q_2)
+0.04*sin(2*q_3 +
2*q_2)+0.02*cos(q_3 +2*q_2))*$\dot{q}_1$)*$\dot{q}_2$)+
(0.6*cos(2*q_3 +2*q_2)-0.37*sin(q_3)-0.37*sin(q_3 +2*
q_2) $+0.04 * sin(2 * q_3 + 2 * q_2)$
+0.01*cos(q_3)+0.01*cos(q_3 +2*q_2))*$\dot{q}_1$)*$\dot{q}_3$)
(+0.27*cos(q_3 +q_2)*$\dot{q}_2$)*$\dot{q}_3$)+
(0.13*cos(q_3 +q_2)*($\dot{q}_2$ *$\dot{q}_2$ +$\dot{q}_3$ *$\dot{q}_3$);
K_0 (2,2)=37.23*sin(q_2)-8.45*cos(q_3 +q_2)+1.02*cos (q_2)-
0.25*sin(q_3 +q_2)-0.01*cos(-q_5 +q_3 +q_2)-
0.01*cos(q_5 +q_3 +q_2)+(0.69*cos(q_2)+0.13*sin(q_3 +q_2)-
0.02*sin(q_2))*$\bar{q}_1$ +(1.6*cos(2*q_2)+0.74*sin(q_3 +2*q_2)-
0.3*cos(2*q_3 +2*q_2 q_2)-0.02*sin(2*q_3 q_3 +2*q_2 q_2)-
0.02*cos(q_3 +2*q_2)-0.01*sin(2*q_2))*$\dot{q}_1$)*$\dot{q}_1$);
K_0 (2,3)=-8.54*cos(q_3 +q_2)-0.25*sin(q_3 +q_2)-
0.01*cos(-q_5 +q_3 +q_2)-0.01*cos(q_5 +q_3 +q_2)+
0.13*sin(q_3)*$\bar{q}_1$ +(0.74*cos(q_3 +0.02*sin(q_3))*$\bar{q}_2$ +
(0.37*cos(q_3)+0.01*sin(q_3))*$\bar{q}_3$)+(-0.74*sin(q_3)+
0.02*cos(q_3))*$\dot{q}_2$)*$\dot{q}_3$)+(-0.37*sin(q_3)+
0.01*cos(q_3))*$\dot{q}_3$)*$\dot{q}_3$)+(0.37*sin(q_3 +2*q_2)-
0.3*cos(2*q_3 +2*q_2)-0.02*sin(2*q_3 +q_2)-
0.01*cos(q_3 +2*q_2))*$\dot{q}_1$)*$\dot{q}_1$);
K_0 (2,5)=0.01*cos(-q_5 +q_3 +q_2)-0.01*cos(q_5 +q_3 +q_2);
K_0 (3,2)=-8.54*cos(q_3 +q_2)-0.25*sin(q_3 +q_2)-
0.01*cos(-q_5 +q_3 +q_2)-0.01*cos(q_5 +q_3 +q_2)+
(0.13*sin(q_3 +q_2))*$\bar{q}_1$ +(0.37*sin(q_3 +q_2)-
0.3*cos(2*q_3 +2*q_2)-0.02*sin(2*q_3 +2*q_2)-
0.01*cos(q_3 +q_2))*$\dot{q}_1$)*$\dot{q}_1$);
K_0 (3,3)=-8.54*cos(q_3 +q_2)-0.25*sin(q_3 +q_2)-
0.01*cos(-q5+q_3 +q_2)-0.01*cos(q5+q_3 +q_2)+
(0.13*sin(q_3 +q_2))*$\bar{q}_1$ +(0.37*cos(q_3)+
0.01*sin(q_3)*$\bar{q}_2$ +(0.37*sin(q_3) -0.01*cos(q_3))*($\dot{q}_2$ *$\dot{q}_2$)
+(0.19*sin(q_3 +q_2)+0.19*sin(q_3)-
0.30*cos(2*q_3 +2*q_2)-0.02*sin(2*q_3 +2*q_2)*($\dot{q}_1$ *$\dot{q}_1$);
K_0 (3,5)=0.01*cos(-q_5 +q_3 +q_2)-0.01*cos(q_5 +q_3 +q_2);
K_0 (5,2)=0.01*cos(-q_5 +q_3 +q_2)-0.01*cos(q_5 +q_3 +q_2);
K_0 (5,3)=0.01*cos(-q_5 +q_3 +q_2)-0.01*cos(q_5 +q_3 +q_2);
K_0 (5,5)=(-0.01*cos(-q_5 +q_3 +q_2))-0.01*cos(q_5 +q_3 +);

Appendix 2: The State Space Matrices

M_0 =					
4.3280	0.1902	0.1300	0	0	0
0.1902	4.4814	0.3707	0	0	0
0.1300	0.3707	1.1600	0	0	0
0	0	0	0.2000	0	0
0	0	0	0	0.1800	0
0	0	0	0	0	0.1900
C_0 =					
-2.6770	1.0832	-1.9746	0	0	0
2.5259	-1.9130	-3.8261	0	0	0
2.0094	1.9130	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
K_0 =					
0	-20.1559	-3.7029	0	0	0
0	27.5016	-5.7046	0	0.0005	0
0	6.6769	3.1350	0	0.0005	0
0	0	0	0	0	0
0	0.0005	0.0005	0	0.0014	0.