# INTELLIGENT ACTIVE FORCE CONTROL OF A RIGID ROBOT ARM USING EMBEDDED ITERATIVE LEARNING ALGORITHM

Musa Mailah

Department of Applied Mechanics
Faculty of Mechanical Engineering
Universiti Teknologi Malaysia
81310 Skudai, Johor Bahru
MALAYSIA
Email: musa@fkm.utm.my

## ABSTRACT

*The paper presents a novel approach to estimating the inertia matrix of a robot arm adaptively and on-line using an iterative learning algorithm. It is employed in conjunction with an active force control strategy which has been shown to be very effective in accommodating the disturbances. A comprehensive study is performed on a rigid two link manipulator subject to a number of loading conditions. Results clearly indicate the effectiveness of the control scheme in compensating the disturbances and at the same time the estimated inertia matrix is optimized to values corresponding to the converged track error as learning progresses. The viability of the proposed control scheme is illustrated through an experimental work carried out on a robot arm.*

## 1.0 INTRODUCTION

The control of robot arm has been a subject of active research for the last two decades. A number of control methods has been proposed ranging from a simple classical proportional-plus-derivative (PD) control to the more recent adaptive and intelligent control. The PD control [1] though simple is quite efficient and provide stable performance at very low speed operation and without/little disturbance. The performance, however, degrades considerably at high speed and with the presence of disturbances. Thus, there is a need to overcome this critical limitation as practical application is becoming increasingly difficult, complex and challenging. This gives rise to a class of adaptive control techniques [2,3,4], which to certain extent successfully improve the stability and robustness of the system by extending the ability to operate in a wider range of parametric or non-parametric uncertainties. However, this often involves complex mathematical manipulations and assumptions. It is thus common to find that this type of control method is limited to theoretical and simulation study. There is a growing trend in robot control in which intelligent mechanism is incorporated using features such as knowledge-based (expert system) [5], neural network [6,7,8], fuzzy logics [9] and iterative learning algorithm [10,11]. Intelligent robotic system is considered the state-of-the art technology in which the machine is designed to emulate part of human attributes especially in the aspects of learning and decision making. A number of research works in this field demonstrated the advantages of the scheme compared to other methods [11].

In this paper, iterative learning technique is used in conjunction with an active force control (AFC) strategy to control a robot arm. It is shown that the iterative learning control mechanism is able to compute continuously and on-line, the estimated inertia matrix of the robot arm while the AFC component excellently compensates for the disturbances.

The paper is structured as follows. The first part deals with a description of the problem statement and the fundamentals of both the AFC and learning control methods. Next, the integration of the iterative learning algorithm and AFC applied to a two link rigid robot arm is demonstrated in the form of a simulation study. This is followed by an analysis and discussion of the results obtained. Experimental work is also carried out to verify the effectiveness of the proposed scheme. Finally, a conclusion is derived and further works which could be carried out are pointed out.

## 2.0 PROBLEM STATEMENT

Active force control applied to robot arm is first proposed by *Hewit* towards the end of seventies [12]. The aim of this type of control method is to ensure that the system is stable and robust even in the presence of known or unknown disturbances. A distinct advantage about this method is the practical realization of the system in which the method bases its concept on using mainly the estimated or measured values of certain parameters to effect its compensating action. This has the benefits of reducing the mathematical complexity of the robot system which is known to be highly coupled and non-linear.

The main drawback of AFC is the computation of the estimated inertia matrix which is required in the AFC feedforward loop. Previous methods relies heavily on either perfect modelling of the inertia matrix, crude approximation or the reference of a look-up table which although are quite effective in implementation but lack in systematic approach and technique. Thus, a search for a better way of obtaining the estimated inertia matrix is sought. If a suitable method is found, then the practical value of implementing the AFC method is considerably enhanced. A number of proposals may be considered as potential

options - all seemingly aiming towards the incorporation of intelligent mechanism such as using neural network, fuzzy logics, genetic algorithm, optimization technique and iterative learning method. Within this framework, and with a clear direction in mind, a method has been devised which integrates the intelligent control into the AFC strategy.

The paper describes a novel approach to control a robotic arm using an iterative learning method coupled to the active force control (AFC) strategy. It is demonstrated in this paper the effectiveness of the learning algorithm as an on-line parameter estimator which provide the signal iteratively to the AFC section for the compensation of the introduced disturbances. As a result, the control scheme is able to operate within a wide range of parametric and non-parametric uncertainties. In other words, the proposed system is robust against all forms of disturbances.

The idea behind the scheme is to obtain a continuous computation of the estimated inertia matrix of the arm by means of a suitable learning algorithm in which the arm is gradually forced to execute a prescribed task accurately even in the presence of external disturbances. As the arm starts to move the internal mechanism activates the learning process - identifying new inertia values of the links at each iteration which is fed into the AFC loop, performing the required task and eventually reducing the track error. This error is in turn fed back into the learning algorithm section and the process is repeated iteratively until a suitable error goal criterion is achieved. Figure 1 shows a block diagram representing the interlinking of the proposed control scheme.

To provide better insight to the proposed scheme under study, the fundamentals of both AFC and the iterative learning methods are briefly explained in the following sections.

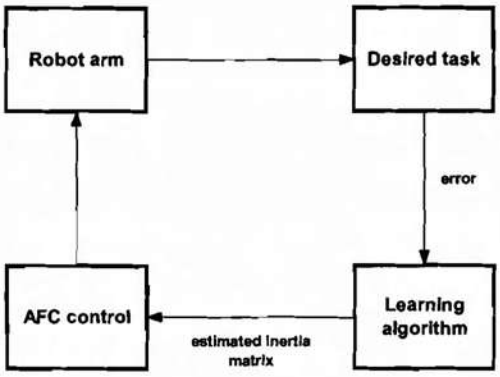



Figure 1 Interlinking of the proposed control scheme

## 3.0 ACTIVE FORCE CONTROL (AFC) AND ITERATIVE LEARNING CONTROL

### 3.1 Active Force Control

In AFC, it is shown that the system subjecting to a number of disturbances remains stable and robust via the compensating action of the control strategy. The full mathematical analysis of the AFC scheme can be found in [12,13,14]. For brevity, the fundamentals of the AFC applied to a robot arm is given in the following paragraph.

From the first principle, we have from *Newton's* second law of motion, for a rotating mass, the sum of all torques ($T$) acting on the body is the product of the mass moment of inertia ($I$) and the angular acceleration ($\alpha$) of the body in the direction of the applied torque, i.e.,

$$\Sigma T = I\,\alpha \tag{1}$$

For a robot system having serial configuration, we have,

$$T + Q = I(\theta)\,\alpha \tag{2}$$

We can obtain a measurement of $Q'$ of $Q$ as

$$Q' = I'\,\alpha' - T' \tag{3}$$

where the superscript' denotes a measured or computed (or estimated) quantity. In this context, $T'$ can be easily measured by means of a current sensor and $\alpha'$ using accelerometer. $I'$ may be obtained by assuming a perfect model or simply crude approximation.

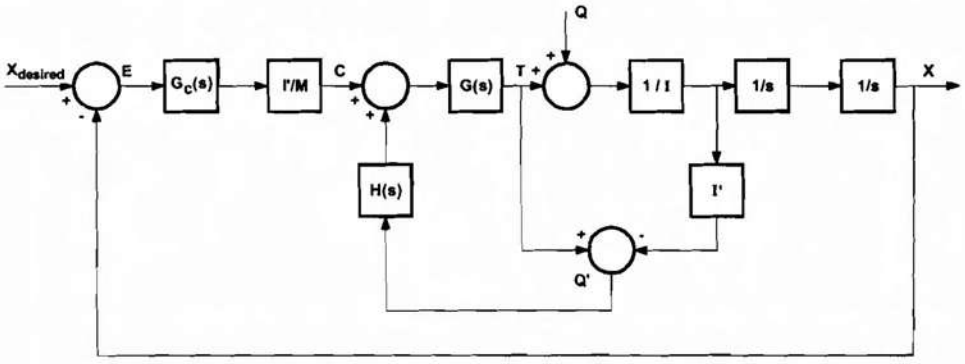



Figure 2 A schematic diagram of an AFC scheme

Consider the schematic diagram in Figure 2. The estimated variable $Q'$ is passed through a function $G(s)$, before subtraction from a command vector $C$ at a summing junction. A suitable choice of $G(s)$ can cause the output $X$ to be made invariant with respect to the disturbances $Q$. A suitable set of control loop is applied to the above open loop system, by first generating the world coordinate error vector, $E = (X_{desired} - X)$ which would then process through a controller function, $G_C(s)$ (e.g., a PD controller). This is followed by a decoupling transfer function $(W')^{-1} = I'/M$ where M is a suitable constant. Thus, the system is reduced to a set of non-interacting loops. An outer positional loop is formed through the world coordinate error vector, $E$.

The main computational burden in AFC is the multiplication of the estimated inertia matrix with the angular acceleration of the arm before being fed into the AFC feedforward loop. Apart from that, the output $X$ needs to be computed from the joint angle $\theta$ via forward kinematics and also the controller $G_C(s)$ be determined.

For a given robot arm, the above expressions may be translated in the form shown in Figure 3.

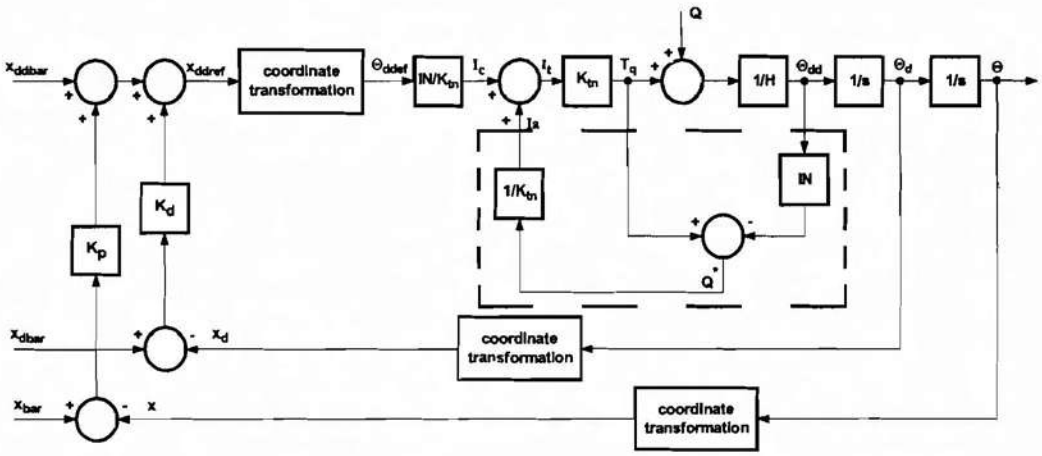



Figure 3 The AFC scheme applied to a robot arm

We have here,

$$G(s)\ W(s) = 1 \tag{4}$$

i.e., $G(s) = K_{tn}$ and $W(s) = 1/K_{tn}$.

The controller $G_C(s)$ chosen here is the resolved-motion-acceleration-control (RMAC) employing a proportional-plus-derivative (PD) component. The RMAC produces the acceleration command vector signal $\theta_{ddref}$ which when multiplied with a decoupling transfer function gives the required command vector

to the main AFC loop. The equation describing the disturbances is given as follows:

$$T_d^* = IN\,\theta_{dd} - T_q \tag{5}$$

In AFC, we can effectively accommodate the disturbances by obtaining the measurements of the acceleration and the torque using physical accelerometer and torque sensor respectively. More conveniently, we can rewrite Equation (5) (based on the torque-current relationship) in the following form:

$$T_d^* = IN\,\theta_{dd} - K_{tn}\,I_t \tag{6}$$

In this way, we can instead measure the controlled current $I_t$ to the motor and obtain exactly the same result. The AFC concept has been successfully implemented to robot arm via simulation and experimental works [16,17,18,19].

The only additional and necessary requirement is the acquisition of an appropriate estimated inertia matrix of the arm to be multiplied with the 'measured' acceleration as in Equation (6). Previous cited works on AFC use traditional techniques which are rather crude, not systematic and mostly based on rough estimation. Thus, it is highly desirable that a method should be devised in such a manner that the inertial parameter can be identified intelligently without having to resort to the conventional approaches described above. A novel method has been proposed here using iterative learning algorithm which is described in the following section. Results obtained through simulation and experimental study demonstrate the effectiveness and simplicity of the proposed method.

### 3.2 Iterative Learning Control

One of the early proposer of the iterative learning method applied to robotic control is *Saguru Arimoto* who proposes a number of learning algorithms and at the same time provides analytical proof of their convergence, stability and

robustness [10, 20,21,22,23]. In other words, it can be shown that as the number of iteration $k(t)$ increases i.e $k \rightarrow \infty$ for $t \in [0, tstop]$, the track error $e$ converges to zero ($e \rightarrow 0$). A suitable algorithm is described in this paper and later implemented to the system under study. A learning algorithm of the following form is chosen:

$$y_{k+1} = y_k + (\phi + \Gamma\, d/dt)\, e_k \tag{7}$$

It is obvious that the algorithm contains a constant and derivative coefficients of the track error. In other words, the expression can be simply called the proportional-derivative or PD type learning algorithm. The above algorithm can be slightly modified to suit our application and is given as follows:

$$IN_{k+1} = IN_k + (\phi + \Gamma\, d/dt)\, TE_k \tag{8}$$

Figure 4 shows a block diagram describing the above expression.

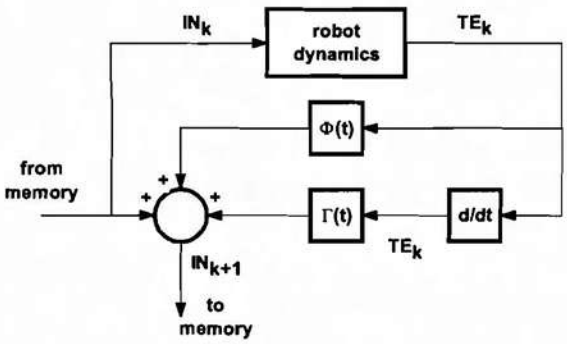



Figure 4 A PD-type learning algorithm

### 3.3 The Proposed Control Scheme

We shall call the proposed scheme AFCAIL, an acronym for Active Force Control And Iterative Learning. Since we use the PD type learning algorithm in the study, the control scheme is known as AFCAIL-PD. The implementation of

the PD type learning algorithm in AFC is shown in Figure 5. The box (shown by dashed lines) represents the most important part of of the proposed scheme which integrates AFC and the learning algorithm. Note that the learning algorithm is easily embedded into the parent AFC scheme with the track error *TE* as the input and the estimated inertia matrix **IN** as the output of the learning algorithm.

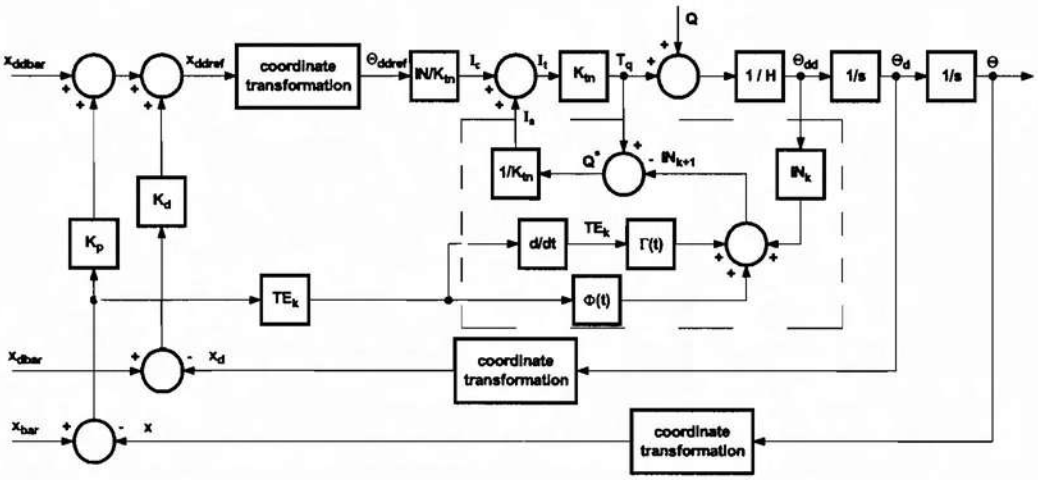



Figure 5 The AFCAIL-PD control scheme

A flow chart showing the logical flow of the main algorithm is illustrated in Figure 6.

A simulation study of the above integrated scheme is made considering a number of varying parameters. The robot arm chosen is a rigid two-link arm assuming to operate in a horizontal plane. Before proceeding to the main simulation study, a brief description of the mathematical model of the robot arm is given in the following section.

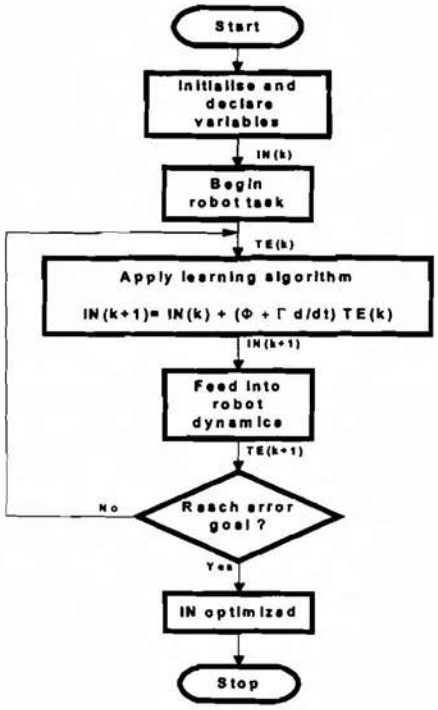



Figure 6 The learning algorithm applied to the robot arm

## 4.0 MATHEMATICAL MODEL OF THE ROBOT ARM

The general equation of motion (dynamic model) of a robot arm can be described as follows:

$$T_q = H(\theta)\,\theta_{dd} + h(\theta, \theta_d) + G(\theta) + T_d \qquad (9)$$

For a horizontal two-link rigid planar manipulator as shown in Figure 7, its dynamic model is given by,

$$T_{q1} = H_{11}\theta_{dd1} + H_{12}\theta_{dd2} - h\theta_{d2}^2 - 2h\theta_{d1}\theta_{d2} \qquad (10)$$

$$T_{q2} = H_{22}\theta_{dd2} + H_{21}\theta_{dd1} - h\theta_{d1}^2 \qquad (11)$$

where

$$H_{11} = m_2 l_{c1}^2 + I_1 + m_2 (l_{c1}^2 + l_{c2}^2 + 2 l_1 l_{c2} \cos \theta_2) + I_2 \quad (12)$$

$$H_{12} = H_{21} = m_2 l_1 l_{c2} \cos \theta_2 + m_2 l_{c2}^2 + I_2 \quad (13)$$

$$H_{22} = m_2 l_{c2}^2 + I_2 \quad (14)$$

$$h = m_2 l_1 l_{c2} \sin \theta_2 \quad (15)$$

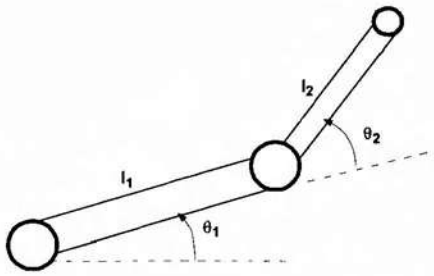



Figure 7 A representation of a two-link arm

It should be noted that the gravitational term of Equation (9) has been ommitted since the arm is assumed to move only in a horizontal plane.

## 5.0 SIMULATION

Simulation work is performed using the MATLAB® and SIMULINK® software packages. The SIMULINK® block diagram for the proposed scheme is shown in Figure 8. It comprises a number of components and subsystems; the trajectory planner, the RMAC section, main AFC loop, robot dynamics, iterative learning model and the disturbance model. These are interlinked by means of connecting lines representing the flow of signals and the relevant building blocks acquired from the SIMULINK® library. In the simulation program, a number of

® MATLAB and SIMULINK are registered trademarks of The Math Works Inc.

disturbance torques can be described and introduced to the system. The disturbance torques considered in the simulation are the harmonic force at end of the second link.

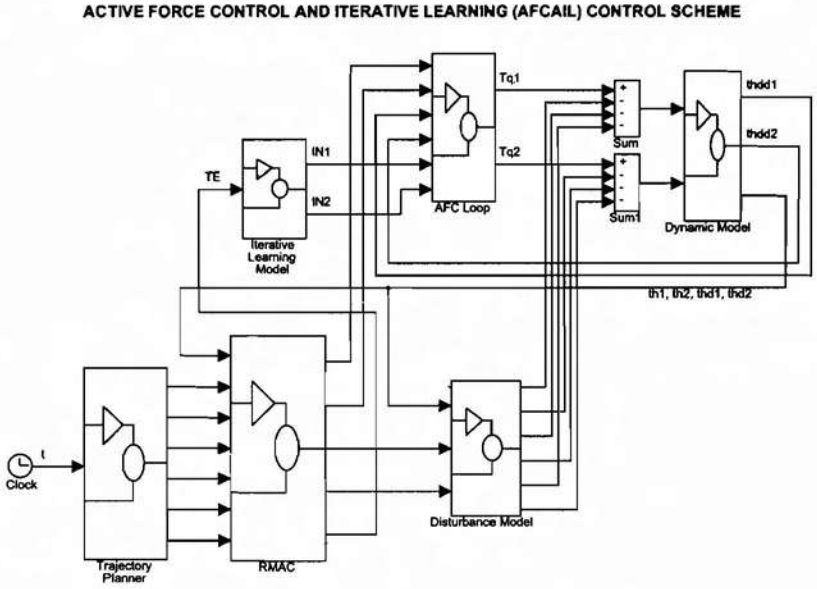



Figure 8 A SIMULINK® representation of the AFCAIL scheme

## 5.1 Simulation parameters

The following parameters are used in the simulation study.

**Robot parameters :**

| | | |
|---|---|---|
| Link length, | $l_1 = 0.25$ m, | $l_2 = 0.2236$ m |
| Link mass, | $m_1 = 0.3$ kg, | $m_2 = 0.25$ kg |
| Motor mass, | $mot_{11} = 1.3$ kg, | $mot_{21} = 0.8$ kg |
| Payload mass, | $mot_{22} = 0.1$ kg | |

**Controller parameters :**

| | | |
|---|---|---|
| Controller gain, | $K_p = 750$ /s, | $K_d = 500$ /s$^2$ |
| Motor torque constant, | $K_{tn} = 0.263$ Nm/A | |

**Iterative learning parameters :**

| | |
|---|---|
| Proportional term, | $\Phi = 0.005$ |
| Derivative term, | $\Gamma = 0.0075$ |

**Main simulation parameters :**

| | | |
|---|---|---|
| Integration algorithm | : | Gear |
| Simulation time start, tstart | : | 0.0 |
| Simulation time stop, tstop | : | 25 s |
| Minimum step size | : | 0.01 |
| Maximum step size | : | 0.1 |

The gain constants, $K_p$ and $K_d$ of the control scheme are assumed to be satisfactorily tuned heuristically prior to the simulation work. The motor torque constant $K_{tn}$=0.263 Nm/A is obtained from the actual data sheet for the DC torque motor. Simulation is performed first without considering any external disturbances acting on the system. Later, the applied disturbance is assumed.

## 5.2 The Initial Conditions

In this study, we consider the initial state of the estimated inertia matrix of the arm to start from 0 kgm$^2$ or specifically $IN_1$ = 0.0 kgm$^2$ and $IN_2$ = 0.0 kgm$^2$. In other words, no prior knowledge of the inertial parameter is assumed - which is a very important contribution of the research work.

### 5.3 The Sample Time

The sample time instant for the learning algorithms is set to 0.01s. This implies that the next step value of **IN** is updated every 0.01s, i.e., if the initial value of $\mathbf{IN}_k$ (initial $k$=0) is 0 s then the next value $\mathbf{IN}_{k+1}$ is sampled at 0.01s and so forth. For the purpose of the analysis of the results obtained, rather than using iteration number, it is preferred that the description of the convergence of the algorithms is based on the number of cycles of the circular trajectory generated which is calculated from the time it takes for the trajectory to complete a perfect circle (for the simulation this time is calculated to be $t_C$ =3.14 s). Thus, 2 cycles would be $2*t_C$, 5 cycles is $5*t_C$ etc.

### 5.4 The Stopping Criteria

A stopping criterion should be specified especially if convergence of the desired parameter (or learning) has taken place. In this paper, we consider a suitable elapsed time of 25s or approximately 8 cycles ($8t_C$) of a complete circular trajectory as the stopping mechanism so that the behaviour of the system can be observed and critically analyzed especially on-line.

### 5.5 The Prescribed Trajectory

A prescribed circular trajectory as shown in Figure 9 is considered in the simulation study. The trajectory is generated using the following time ($t$) dependent functions:

$$x_{bar1} = 0.25 + 0.1 \sin (V_{cut}\, t/0.1) \tag{16}$$

$$x_{bar2} = 0.1 + 0.1 \cos(V_{cut}\, t/0.1) \tag{17}$$

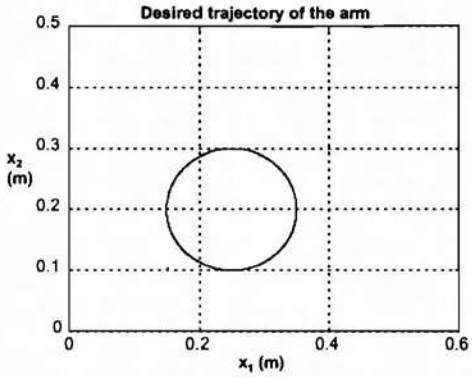



Figure 9 The desired trajectory of the arm

### 5.6 The Applied Disturbances

An explicit account of the effect of the disturbances applied to the robot arm is given. This is performed to investigate the robustness of the proposed scheme. The simulation is first carried out without considering any external disturbances acting on the system. Later, we introduce a harmonic force which is applied at the end of second link. We have the harmonic force,

$$F_h = \mathrm{h} \sin t \tag{18}$$

where the magnitude of force is h = 30 N.

## 6.0 RESULTS AND DISCUSSION

Figure 10 and 11 show the detailed results obtained through the simulation work. The graphical results are related to the trajectory obtained, the track error produced, the estimated inertia matrix computed and the resulting torque actuated.

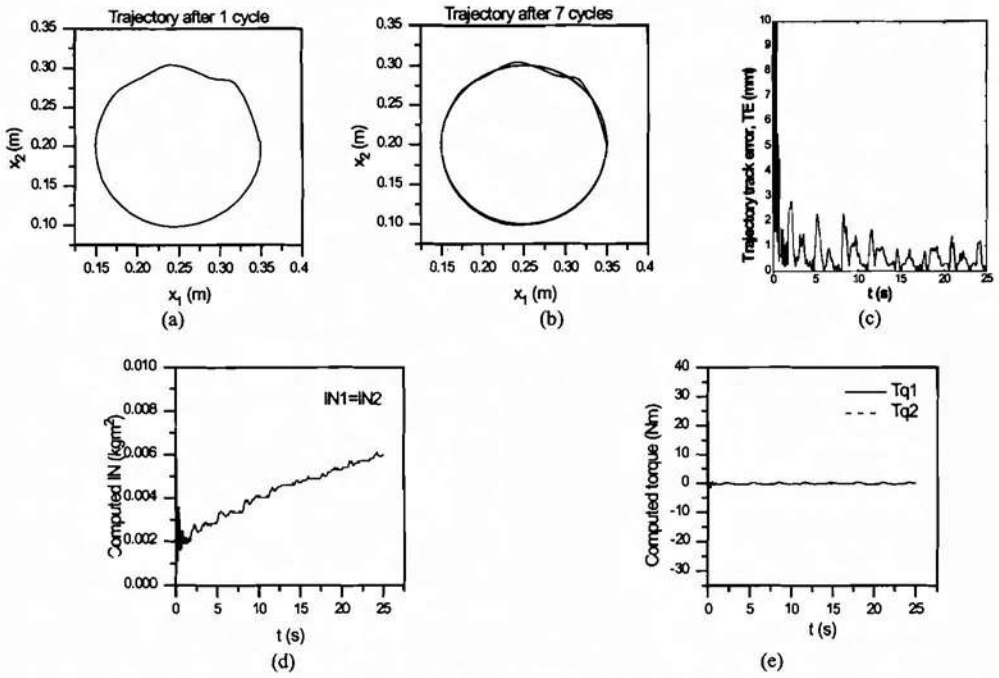



Figure 10 Results for the AFCAIL-PD scheme, no disturbance, $F_O$

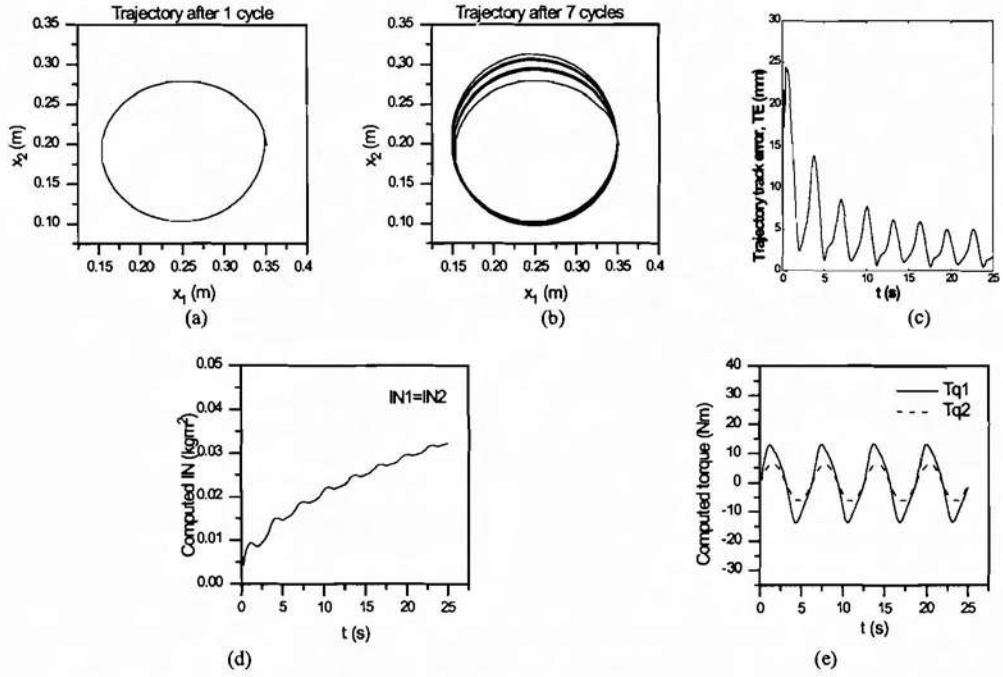



Figure 11 Results for the AFCAIL-PD scheme, harmonic force, $F_h$

It is obvious that the learning process is accomplished gradually with the initial generated trajectory appearing distorted and disoriented. Figure 10(a) and 11(a) show the actual trajectories of the arm after the robot performs 1 complete cycle of trajectory tracking. As the iteration proceeds, the eventual trajectory almost replicates the desired one, signifying that learning process has indeed ocurred. This is especially observed after the learning has reached 7 cycles as depicted in Figure 10(b) and 11(b). The track error curves confirm this fact - the initial stage is characterized by very large error but as learning takes place, the track error is observed to continually and gradually decrease and this applies for all conditions considering with or without external disturbances as shown in Figure 10(c) and 11(c). In addition to that, it is also noticed that the pattern of the trajectory track error corresponds to the types of external disturbances applied to the system.

The proposed control scheme is shown to have fast learning characteristic - the trajectory track error for all the cases dips to below the 2.5 mm margin at the end of the simulation period. The gradual mapping of the trajectory to the desired one is found to be smooth and the final track error is acceptably small.

Generally, the computed **IN** for all the cases, is seen to rise steadily and upwardly from the given initial conditions as shown in Figure 10(d) and 11(d). The initial state of the inertia matrix is characterized by a sharp jump to a value which more or less stabilizes and later gradually and steadily increases in a non-linear fashion as iteration continues and the learning rule computes new value of **IN** for the next sample instant. It is essential to ensure that **IN** should be positive definite which would otherwise causes the system to become unstable. The choice of suitable learning parameters ($\phi$ and $\Gamma$) for the learning rules is also important as this affects the slope of the **IN** curve. For instance, inappropriate value of the parameters would cause the gradient of **IN** curve to increase dramatically and thus lead to instability. A slight and gradual increase in the slope would be sufficient to yield a very good response. A number of trials is

performed to determine the value of these constants and the results suggested that the values assumed are acceptable.

It is also obvious that the inertia matrix varies with the type of applied disturbances. The **IN** curve shows small fluctuating pattern corresponding to the harmonic input as hown in Figure 11(d).

### 6.1 Initial Value of the Estimated Inertia Matrix

A significant and indeed a desirable feature of the employed learning scheme is realized when the initial estimated inertia matrix is assumed to be of the same value. This implies that only a single value of the estimated inertia matrix is needed to perform the required task. Thus, by putting the initial value $IN_1=IN_2=0.0$ kgm$^2$, we have both the computed values to be exactly the same and yet producing stable performances with the trajectory track error reduced to acceptable tolerance. This has the added advantage in terms of less computation is required to determine **IN** especially if the degrees of freedom of the arm is greater than two. In this context, we can assume all the diagonal terms of the inertia matrix to be of the same value. As far as the two-link arm is concerned, the simulation results indicate that the assumption is valid.

Thus, as an example, for a three degree of freedom robot arm, we may have,

$$\mathbf{IN} = \begin{bmatrix} IN_1 & 0 & 0 \\ 0 & IN_2 & 0 \\ 0 & 0 & IN_3 \end{bmatrix} \tag{19}$$

where $IN_1=IN_2=IN_3$ assuming that the learning algorithm has optimized' the **IN**.

Figure 10(d) and 11(d) show the computed **IN** for the different types of disturbances but having the same initial conditions of **IN**.

## 6.2 The Computed Inertia Matrix

The computed estimated inertia matrix varies in a range of values as summarized in Table 1.

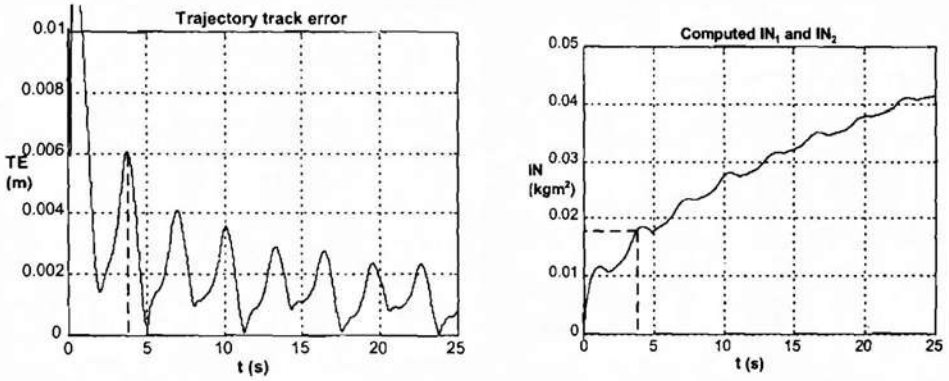
Table 1 Range of computed estimated inertia matrix

| **Types of disturbances** | **Range of computed IN** $IN_1 = IN_2$ (kgm$^2$) |
|---|---|
| No disturbance, $F_o$ | 0.0 - 0.007 |
| Harmonic force, $F_h$ | 0.0 - 0.042 |

From the table, it can be seen that the range of **IN** varies positively from 0 kgm$^2$ to a maximum value of 0.042 kgm$^2$. The harmonic force produces higher **IN** due to the fact that the force is continuously acting on the system. In other words, **IN** is influenced considerably by the nature of the forces applied. The robot arm responds to these disturbances by adaptively updates the **IN** via the control strategy. Since the error is considered 'minimum' at the end of the simulation period, this corresponds to the optimum value of the computed **IN**. On the whole, even though there is a variation in the computed **IN** for different cases, the results have clearly indicated that the trajectory track error converges to acceptable value signifying that the system is very stable and robust. It also suggests that the intelligent mechanism has successfully adapted the inertial parameter to the changes in the environment. The other reason is attributed to the fact that the AFC scheme is very tolerant to variation in the estimated inertia matrix - which somehow explains the successful implementation of AFC by using only crude approximation.

An interesting feature can be derived from the study - we can identify a '*safety zone*' for a range of suitable **IN**. We know that as learning progresses, **IN** approaches the tolerable range in which the system behaves robustly even in the

presence of large external disturbances. We can determine this range by observing the characteristics of the trajectory track error from which an error margin can be specified and thus, a range of **IN** values can be easily determined. As an example, by assuming a starting error margin, we can obtain the corresponding time **t** it occurs and then relates it to the **IN** curve. Figure 12(a) and (b) shows how this is accomplished. By setting the track error *TE* to 6.0 mm, we project a line (dashed lines) to the horizontal axis and obtain $t$ to be about 3.8 s. Referring to the **IN** curve at time $t$=3.8 s, we have **IN** value equals 0.018 kgm$^2$. This can be regarded as the starting **IN** at which the system behaves stably and robustly. The other extreme end of the range can be obtained by extending the simulation time to such an extent that the error margin increases dramatically or when the system starts to behave awkwardly or erratically.




(a) (b)

Figure 12 Determining the 'safety zone'

## 6.3 The Actuated Torques

It can be seen that the controlling torque varies non-linearly and is influenced by the disturbances applied (see Figure 10(e) and 11(e)). In this study, the saturation value of the torque is not considered at all in order to observe the system's

robustness against large forces. The saturation value of the torque can be easily calculated from the following equation:

$$T_q = I_t \ K_t \tag{20}$$

From the actual data sheet, $K_{tn}$ is found to be 0.263 Nm/A and the maximum permissible motor current is 12.3A. Thus the maximum torque is ±3.3 Nm. In practice, this value must be kept within the maximum allowable limit to avoid damage to the motor. From the results, it can be deduced that, apart from the case where there is no disturbances acting on the robot arm, others with various applied disturbances produce torques which are by far greater than the saturated value. This implies that the AFC scheme is excellent in `cancelling' the disturbances even though the forces present are much larger than the system can physically tolerate. It is also obvious that the controlling torque at the first joint is greater than that at the second joint, i.e. $T_{q1} > T_{q2}$. This may be contributed to the coupling effect of the arm and the accumulated load with respect to the first joint.

## 7.0 EXPERIMENTAL RESULTS

An experiment was carried out using a single link robot arm as shown in Figure 13. We use a non-linear spring of unknown stiffness attached to the tip of the arm as the force controlled element.

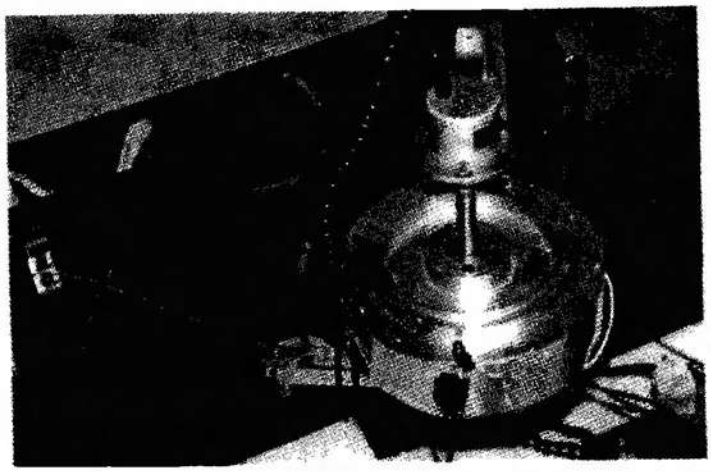


Figure 13 A view of the experimental robot arm with a spring attached to the tip of the link

The following parameters were used in the experiment :

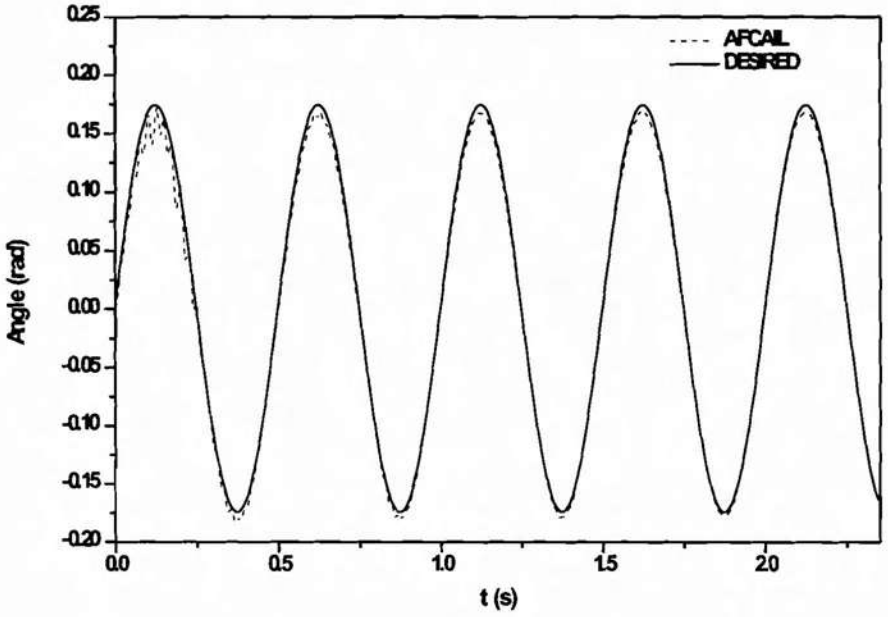
$K_p = 115$, $K_i = 20$, $K_d = 1$

$K_t = 0.3387$ Nm/A

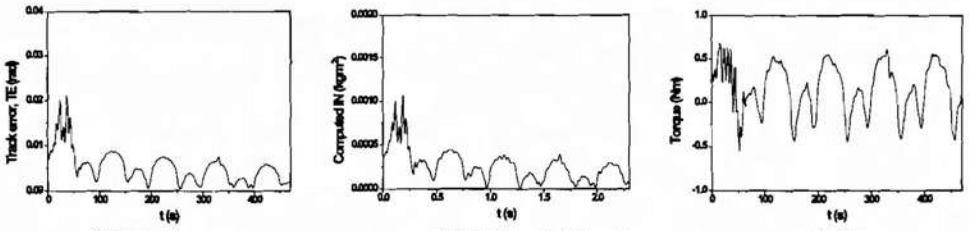
Learning parameter : $\Phi = 0.05$, $\Gamma = 0.0001$

A sinusoidal input with peak amplitude of 0.1745 rad.

Referring to Figure 14, the trajectory of the arm as shown in (a) gradually improves with the number of iteration showing that the learning process has indeed taken place. The initial stage of the experiment is characterized by the inconsistent fluctuating movement of the arm for the first 0.5 s or so but as the learning process commences, the trajectory considerably improves.




(a) Trajectory



(b) Track error (c) Estimated inertia (d) Torque

Figure 14 Experimental results

This is verified by observing the characteristic of the track error curve in (b) which shows the corresponding gradual decrease in the error. The learning algorithm is evidently forcing the track error to converge to values approaching the zero datum as time increases. It is also noted that the curve of the estimated inertia of the arm (c) follows the same pattern to that the track error curve, which

is understandable since the estimated inertia matrix is a function of the track error according to the learning rule in Equation (8). The varying inertias of the arm are obviously having positive values as expected from the theoretical point of view. The maximum inertia occurs at the beginning of the cycle which is approximately 0.01 $kgm^2$. The value drops to around 0.0025 $kgm^2$ at $t = 2.1$ s. We can calculate the average inertia value to be 0.00032 $kgm^2$. The torque curve can seen to vary in sinusoidal fashion peaking at approximately ±0.6 Nm. Again we can see that the initial stage is governed by the high oscillatory feature of the torque prior to the learning process

We can deduce that the iterative learning control algorithm has been effectively and successfully incorporated into the AFC scheme without any difficulty. The learning process is found to be very fast; about 1 s for the sinusoidal responses. It is thought that one of the contributing factors that makes it easy to be implemented is due to the fact that the actual inertia of the arm is comparatively small (about 0.002 $kgm^2$). Since **IN** is assumed to start at 0 $kgm^2$ and the characteristic of the proposed learning algorithm is such that it only computes positive values (**IN**>0) at every successive trials, the outcome is expectedly to be very favourable. Thus, the experimental results obtained verify the above condition.

## 8.0 CONCLUSION

The proposed scheme has been shown to be very effective in generating the estimated inertia matrix automatically and on-line. The learning strategy causes the trajectory track error to gradually converge to acceptable error margin indicating that the learning process is successful. The AFC part is found to be very robust in tackling the disturbances and can operate effectively in a wide

range of estimated inertia values supplied iteratively by the learning algorithm. Thus the 'twinning' of both the learning and AFC schemes proves to be compatible and very feasible. The learning of the system is reasonably fast - it takes about 25 s to produce an error margin of below 2.5 mm. A significant feature of the proposed scheme is that only a single value of the estimated inertia matrix **IN** can be used for the same initial condition. This implies less computational burden since the scheme is considered to have optimized the value of computed **IN** for the robot arm. The experimental results prove the viability and the effectiveness of the proposed control scheme and also illustrate the ease of embedding the learning rule in the parent AFC section. The performance of the scheme should be further investigated, considering a number of varying parameters such as the changes in the payload mass, learning parameters, initial values of **IN**, types of disturbances and other prescribed trajectories

**NOTATION:**

| | |
|---|---|
| $e_k$ | current positional error input given by $e_k = x_d - x_k$ |
| $G$ | vector of the gravitational torques |
| **H** | $N$x$N$ dimensional manipulator and actuator inertia matrix |
| $H$ | vector of the Coriolis and centrifugal torques |
| $I$ | mass moment of inertia of the link |
| $I(\theta)$ | mass moment of inertia of the robot arm and $\theta$ is the robot joint angle |
| **IN** | estimated inertia matrix |
| $\mathbf{IN}_k$ | current value of the estimated inertia matrix |
| $\mathbf{IN}_{k+1}$ | next step value of estimated inertia matrix |
| $I_t$ | armature current for the torque motor |
| $K_{tn}$ | motor torque constant |
| $\vec{l}$ | vector of link lengths |

| | |
|---|---|
| $l_c$ | vector of link lengths from the joint to the centre of gravity of link |
| $m$ | vector of link masses |
| $Q$ | disturbance torques |
| $T$ | applied torque |
| $T_d$ | vector of the external disturbance torques |
| $T_d$* | estimate of all the disturbance torques |
| $TE_k$ | current root of sum-squared positional track error, $TE_k = \sqrt{(x_{bar}-x_k)^2}$ |
| $T_q$ | applied control torque |
| $T_q$ | vector of actuator torques |
| $y_k$ | current ($k$) output value |
| $y_{k+1}$ | next step value of the output |
| $\alpha$ | angular acceleration of the robot arm |
| $\phi$, $\Gamma$ | suitable constants or learning parameters |
| $\theta_d$ | vector of joint velocity |
| $\theta_{dd}$ | vector of joint acceleration |

## REFERENCES

1. Groover, M.P., Weiss, M., Nagel, R.N., and Odrey, N.G., *Industrial Robotics: Technology, Programming and Applications,* McGraw-Hill Book Co., 1986.
2. Slotine, J.E. and Li, W., Adaptive Manipulator Control: A Case Study, *IEEE Transactions on Automatic Control,* Vol. 38, No. 11, November 1988, pp 995-1003.

3. Sinha, A.S.C., Kayalar, S. and Yourtseven, H.O., Nonlinear Adaptive Control of Robot Manipulators, *IEEE Transaction on Robotics and Automation*, Vol. 2, 1990, pp 2084-2087.

4. Tomizuka, M. and Yao, B., Adaptive Control of Robot Manipulators in Constrained Motion - Controller Design, *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, Vol. 117, September 1995, pp 321-328.

5. Shibata, T., Fukuda, T., Shiotani, S., Mitsuoka T., and Tokita, M., Hiearchical Hybrid Neuromorphic Control System, *JSME International Journal*, Series C, Vol. 36, No. 1, 1993, pp 100-109.

6. Kawato, M., Uno, Y., Isobe, R., and Suzuki, R., Hiearchical Neural Network Model For Voluntary Movement With Aplication to Robotics, *IEEE Control Systems Magazine (8)*, 1988, pp 8-15.

7. Goldberg, K. and Pearlmutter, B., Using A Neural Network to Learn the Dynamics of the CMU Direct-Drive Arm II, *Technical Report CMU-CS-88-160*, Carnegie Mellon University, Pittsburgh, August, 1988.

8. Jung, S. and Hsia, T.C., *A New Neural Network Control Technique For Robot Manipulators*, Robotica (1995), Vol. 13, pp. 477-484, 1995.

9. Ohnishi, K., Shibata, M. and Murakami, T., *A Unified Approach to Position and Force Control by Fuzzy Logic,* IEEE Transactions on Industrial Electronics, Vol. 43, No. 1, February 1996, pp 81-7.

10. Arimoto, S., Kawamura, S. and Miyazaki, F., Bettering Operation of Robots by Learning, *Robotic Systems*, 1984, pp 123-140.

11. Bondi, P., Casalino, G., and Gambardella, On the Iterative Learning Control Theory for Robotic Manipulators, *IEEE Journal of Robotics and Automation*, Vol. 4, No. 1, February 1988, pp 14-22.

12. Astrom, K.J. and McAvoy, T.J., *Intelligent Control : An Overview and Evaluation*, David A. White, Donald A. Sofge, Eds., Handbook of

Intelligent *Control : Neural, Fuzzy and Adaptive Approaches*, Van Nostrand Reinhold, New York, 1992, pp. 3-34.

13. Hewit, J.R., and Burdess, J.S., Fast Dynamic Decoupled Control for Robotics Using Active Force Control, *Mechanism and Machine Theory*, Vol. 16, No. 5, 1981, pp. 535-542.
14. Hewit, J.R., and Burdess, J.S., An Active Method for the Control of Mechanical Systems in The Presence of Unmeasurable Forcing, *Transactions on Mechanism and Machine Theory*, Vol. 21, No.3, 1986, pp 393-400.
15. Hewit, J.R., *Advances in Teleoperations*, Lecture Notes on Control Aspects, CISM, May 1988.
16. Hewit, J.R., and Bouazza-Marouf, K., Practical Control Enhancement via Mechatronics Design, *IEEE Transactions on Industrial Electronics*, Vol. 43, No. 1, February 1996, pp 16-22.
17. Hewit, J.R., Disturbance Cancellation Control, *Proc. of Int'l. Conference on Mechatronics*, Turkey, 1996, pp.135-143.
18. Filippi, E., *Experimental Robot Arm*, Technical Report, Loughborough University of Technology, Loughborough, 1993.
19. Jones, C., *Robot Control,* B.Eng. Thesis, University of Dundee, Dundee, 1995.
20. Arimoto, S., Kawamura, S., and Miyazaki, F., Bettering Operation of Robots by Learning, *Robotic Systems*, 1984, pp 123-140.
21. Arimoto, S., Kawamura, S., and Miyazaki, F., Hybrid Position/Force Control of Robot Manipulators Based On Learning Method, *Proc. of Int'l. Conf. on Advanced Robotics*, 1985, pp 235-242.
22. Arimoto, S., Kawamura, S., and Miyazaki, F., Applications of Learning Method For Dynamic Control of Robot Manipulators, *Proc. of 24$^{th}$ Conf. on Decision and Control*, Ft. Lauderdale, 1985, pp1381-6.

23. Arimoto, S., Kawamura, S. and Miyazaki, F., Convergence, Stability and Robustness of Learning Control Schemes for Robot Manipulators, *Recent Trends in Robotics: Modelling, Control and Education*, ed. by Jamshidi, M., Luh, L.Y.S., and Shahinpoor, M., 1986, pp 307-316.