Jurnal Teknologi, 37(A) Dis. 2002: 55–72

CONTROL OF A ROBOT ARM USING ITERATIVE LEARNING ALGORITHM WITH A STOPPING CRITERION

MUSA MAILAH[1] & JONATHAN CHONG WUN SHIUNG

Abstract. The robust performance of a robot control scheme is vital to ensure that the robot accomplishes its tasks desirably in a constraint environment involving disturbances, parametric changes, uncertainties and varied operating conditions. The study introduces the *Active Force Control and Iterative Learning Algorithm* (AFCAIL) scheme with an improved feature in the form of a suitably designed stopping criterion incorporated in the control strategy. The scheme is applied to the control of a horiziontally operated robotic two-link planar manipulator. The proposed stopping criterion is specifically designed to halt the iterative learning process when the conditions related to the accuracy of the performed tasks and the acquisition of appropriate estimated inertia matrix of the robot arm are favourably met. In this way, the robot is said to perform desirably and excellently. The effectiveness of the scheme is also investigated by considering a number different loading and operating conditions.

Key words: Robot, active force control, iterative learning algorithm, stopping criterion

Abstrak. Prestasi lasak bagi skema kawalan robot sangat perlu untuk memastikan robot dapat bekerja dengan berkesan seperti yang dikehendaki dalam persekitaran terbatas melibatkan gangguan, perubahan parameter, ketidaktentuan dan kepelbagaian keadaan operasi. Kajian yang dibuat adalah berkaitan dengan satu skema kawalan daya aktif dan algoritma pembelajaran berlelaran (AFCAIL) yang melibatkan satu ciri pembaikan dalam bentuk penggunaan kriteria memberhenti yang sesuai dimuatkan dalam strategi kawalan. Skema tersebut digunapakai terhadap sistem pengolah robotik planar berlengan-dua yang beroperasi secara mendatar. Kriteria memberhenti yang dicadangkan adalah direka bentuk untuk memberhentikan proses pembelajaran berlelaran apabila syarat atau keadaan berkaitan dengan kejituan ketika melakukan tugas serta perolehan matriks inersia anggaran pengolah yang dikehendaki dapat dipenuhi. Dengan cara demikian, robot dikatakan dapat beroperasi dengan baik sebagaimana yang diarahkan. Keberkesanan skema juga dikaji dengan mengambil kira beberapa keadaan bebanan dan operasi.

Kata kunci: Robot, kawalan daya aktif, algoritma pembelajaran berlelaran, kriterion memberhenti

1.0 INTRODUCTION

With the ever increasing in complexity of the robot tasks, a robot control system engineer has to design and come up with a control scheme that will enable the proposed robotic system to perform the required tasks with a high degree of precision, accuracy and reliability particularly under various conditions involving the robot's interaction

[1] Department of Applied Mechanics, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 Skudai. musa@fkm.utm.my

with its environment. This also implies that the system has to be robust, stable and effectively capable to accomplish the prescribed tasks even in the presence of disturbances, parametric changes, uncertainties and varied operating conditions. There has been a growing trend in the development and implementation of intelligent mechanisms in robot control [1,2]. These mechanisms include the application of neural networks [3,4], fuzzy logics [5,6] and iterative learning algorithms [7–12]. They are normally incorporated into the robot force control strategy to enhance the system's overall performance by generally introducing an automatic decision making facility in the control loop. One such control scheme is the active force control (AFC) strategy [13] applied to the control of a robot arm. This control scheme together with the suitably proposed intelligent mechanism is the main focus of the study. AFC has been shown to be very robust and effective in countering disturbances and varied operating conditions. A number of intelligent AFC schemes has been developed using neural network, fuzzy logics and iterative learning algorithms [14,15]. It is the aim of the paper to improvise the latter scheme known as *Active Force Control And Iterative Learning* (AFCAIL) to include a stopping criterion that is necesssary to halt the operation of the robot arm, upon satisfying a number of prescribed conditions.

The paper is structured as follows; the first part describes the motivation of the study and the basic underlying principles of the AFC and iterative learning theories. It is followed by a narration of the design of the stopping criterion and later, the simulation study to verify the effectiveness of the proposed concept. The analysis of the results ensues and finally a conclusion is drawn plus a numbers of suggestions for future works outlined.

2.0 PROBLEM STATEMENT

The main drawback of the AFC scheme is the task of computing the estimated inertia matrix of the robot arm which is essential in the AFC feed-forward control loop. The application of the iterative learning has been proven to be very effective in estimating the appropriate inertia matrix of the robot arm automatically, continuously and on-line while the robot is performing its task [14]. However, the computation of the matrix is done in a 'free flowing' manner without any reference made to corrective actions to be taken to inform the system that the learning process has been complete. In other words, there is no stopping mechanism to halt the iteration process even though the matrix has been considered appropriately estimated and the system has been performing desirably. There is a danger of the system going into instability as iteration continues infinitely. Thus, a suitable stopping criterion for the AFCAIL scheme should be designed and developed based on a set of pre-defined conditions such as those related to accuracy and stability. The incorporation of the stopping criterion is expected to provide a clearer picture of the system pertaining to its performance and could also minimise the time and resources involved.

3.0 ACTIVE FORCE CONTROL

The underlying principles of the AFC scheme is explained in this section. From *Newton*'s second law of motion, for a rotating mass, the sum of all torques (τ) acting on a rigid rotating body is the product of the mass inertia (I) and the angular acceleration ($\ddot{\theta}$) of the body in the direction of the applied torque.

$$\sum \tau = I\ddot{\theta} \tag{1}$$

The general equation for a robot system with serial configuration is,

$$\tau + \tau_d = I(\theta)\ddot{\theta} \tag{2}$$

where τ is the applied torque by the actuator, τ_d is the disturbance torque, $I(\theta)$ is the mass moment of inertia of the robot arm with q being the joint angle and $\ddot{\theta}$ is the angular acceleration of the robot arm.

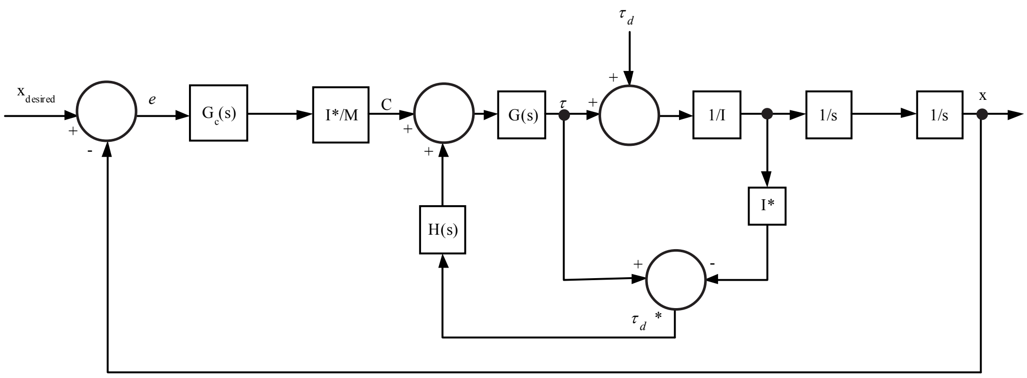


Figure 1 A general schematic diagram of the AFC scheme

In order to maintain robustness and stability, the disturbance torques can be estimated according to the following expression,

$$\tau_d^* = I(\theta)^* \ddot{\theta}^* - \tau^* \tag{3}$$

where the use of the superscript * implies measured or computed (or estimated) quantities. Practically, the quantities τ^* and $\ddot{\theta}^*$ can be measured directly using torque sensor and accelerometer respectively. On the other hand, the mass moment of inerta inertia of the arm, I^*, should be appropriately estimated using a suitable method such as crude approximation, reference of a look-up table or intelligent means [13]. A general schematic diagram of the AFC scheme is shown in Figure 1.

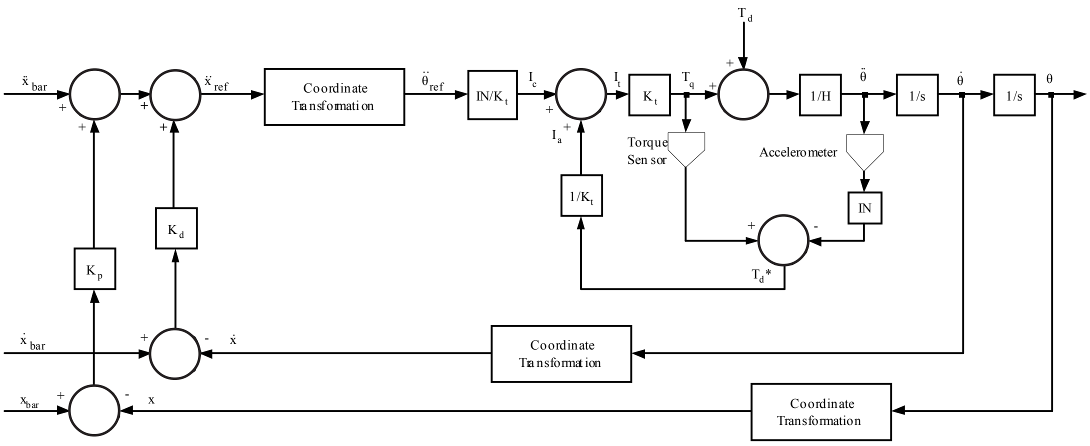


Figure 2 The AFC scheme applied to a robot arm

As mentioned before, in the AFC scheme applied to a robot system, the main computational burden is the multiplication of the estimated inertia matrix with the angular acceleration of the arm before fed into the AFC feed-forward loop. A suitable controller $G_c(s)$ has to be determined and the following expression is considered:

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

where $G(s) = K_t$ and $W(s) = 1/K_t$ with K_t being the motor torque constant.

Like many other control schemes, the AFC scheme also incorporates classical control such as the proportional-derivative (PD) control. Specifically, the controller is the resolved-motion-acceleration-control (RMAC) with PD element that provides the acceleration command vector signal which when multiplied with a suitable transfer function produces the required command vector to the main AFC loop. Referring to Figure 2, the estimated disturbance torques can be described as,

$$T_d^{\,*} = T_q - \mathbf{IN}\,\ddot{\theta} \tag{5}$$

where $T_d^{\,*}$ is the estimated disturbances torques, T_q is the measured actuator torque, **IN** is the estimated inertia matrix, and $\ddot{\theta}$ is the measured angular acceleration

The actuator torque can also be expressed as

$$T_q = K_t I_t \tag{6}$$

where K_t is the motor torque constant available in data sheetsa and I_t is the armature current for the torque motor.

Thus, Eq. (5) becomes

$$T_d^* = K_t I_t - \mathbf{IN}\,\ddot{\theta} \tag{7}$$

Note that the torque current (of the motor), $I_t = I_c + I_a$ where I_c is the current command vector and I_a the compensated current vector.

4.0 ITERATIVE LEARNING ALGORITHM WITH AFC

Iterative learning control is an approach to improving the transient response performance of systems that operate repetitively over a fixed time interval [7–12]. It is also known as "betterment process" or "repetitive control". In other words, the iterative learning algorithm will cause the performance of a dynamical system (based on some error criterion) to become better as time increases. Symbolically, the track error, TE approaches the zero datum as time heads for infinity, *i.e*., $TE \to 0$ as $t \to \infty$. Arimoto *et al.* [9] has provided a sufficiently in-depth analysis of the convergence, stability and robustness of the iterative learning algorithm incidentally employed in the study.

For the AFC scheme, the iterative learning algorithm is used to estimate the inertia matrix based on the trajectory track error of the arm when describing a reference trajectory. The track error [14] may be defined as,

$$TE_k = \sqrt{(x_{bar} - x)^2 + (y_{bar} - y)^2} \tag{8}$$

where x_{bar} and y_{bar} are the desired *Cartesian* coordinates of the end-effector and x and y are the actual *Cartesian* coordinates of the end-effector.

Equation (8) is also known as the root of sum-squared track error. It differs slightly from the one used by Arimoto *et al*. (absolute track error was used instead) [7–9]. The purpose of using the sum-squared terms in the equation is to ensure that only positive values for the TE_k were generated since it has been theoretically asserted that the values of the inertia matrix of the arm should be always positive definite [16]. For the proposed scheme, the following equation was used,

$$\mathbf{IN}_{k+1} = \mathbf{IN}_k + (\phi + \Gamma d/dt)TE_k \tag{9}$$

where $\mathbf{IN}_{k+1}$ is the next step value of the estimated inertia matrix

$\mathbf{IN}_k$ is the current estimated inertia matrix

TE_k is the current root of sum-squared track error

ϕ and Γ are suitable positive definite constants or learning parameters

The incorporation of the above learning algorithm into the AFC scheme enhances the system's performance in terms of its ability to compute **IN** effectively considering suitable initial conditions. The inertia matrix could then be fed into the AFC feed-forward loop to effect the control action. Figure 3 illustrates how the AFCAIL scheme works. Information about the robot's trajectory or motion while performing a specific task is relayed to a feedback sub-system in the outer control loop of the overall control scheme when the robot moves. This information is compared with the desired reference trajectory and the difference between the two gives the track error. The resulting track error is then used by the iterative learning algorithm to determine the inertia matrix of the arm for the next cycle (the next time instant data is taken or the next sampling time).

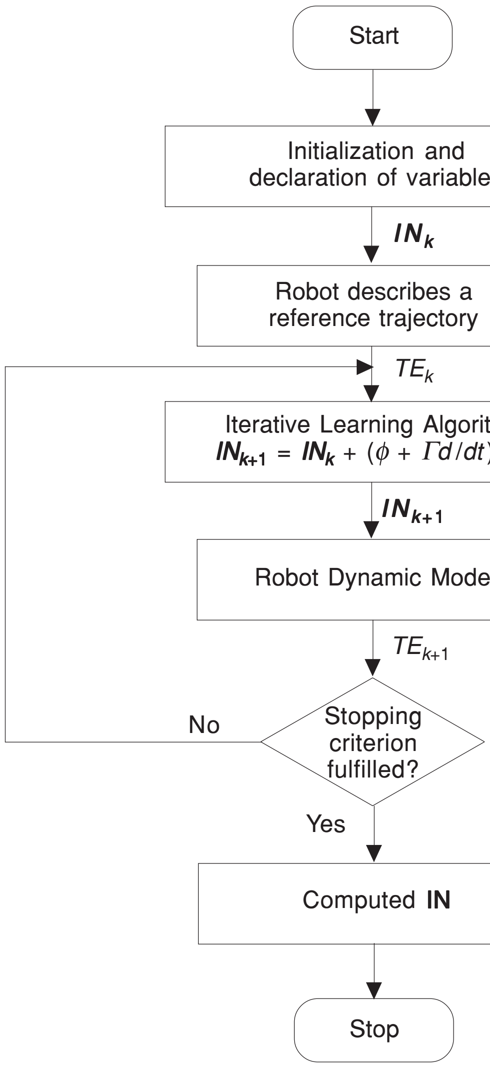


Figure 3 The AFC and iterative learning control scheme

Consequently, this estimated inertia matrix is introduced into the main AFC loop where the compensation of the disturbances by way of estimating the disturbance torques takes place. The robot is thus 'prepared' to counter the effects that cause the previous error and ultimately force the system to behave robustly. The track error was gradually reduced to minimal value as iteration proceeded and hence producing a good appropriate value of the inertia matrix in the process.

A schematic block diagram representing the AFCAIL scheme is shown in Figure 4. Note that the dashed box contains the iterative learning mechanism to compute the required estimated inertia matrix based on the trajectory track error. A PD control with suitable controller gains (K_p and K_d) embeds in the resolved-motion-acceleration-control (RMAC) can be seen in the left hand side of the diagram. By way of applying a coordinate transformation method, an angular acceleration vector command $\ddot{\theta}_{ref}$ is produced. This is later multiplied with a suitable transfer function before being fed into the torque control loop. Disturbances (external), T_d, can be introduced to test the system's robustness. A number of disturbances are modelled in the study.

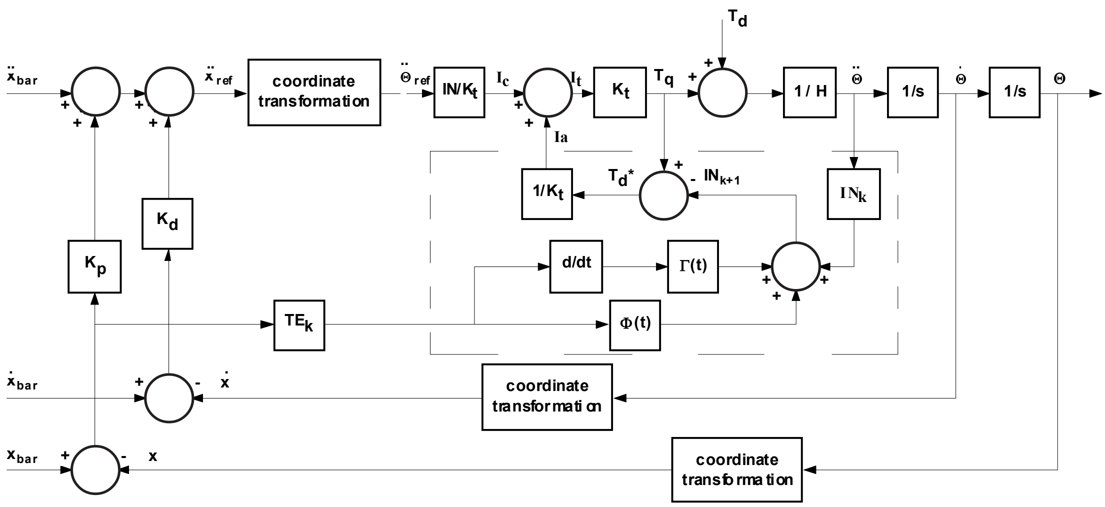


Figure 4 The AFCAIL scheme

5.0 THE STOPPING CRITERION

An iterative method computes successive approximations to the solution of a system such that the output of the system approaches an appropriate value as the time increases. The learning process, as it is oftenly referred to, however, is accomplished infinitely with the possibility of *over learning*, a term to describe a condition or an instant when the system is said to have performed excellently but the system keeps on executing the learning algorithm iteratively but irrelevantly due to the absence of a stopping mechanism. This condition could lead to instability of the system once it enters a 'dangerous zone' where the estimated parameter of the system could no longer maintains a suitable range of values that ultimately triggers a severe degradation of the system's performance [14]. Thus, a mechanism must be incorporated into the algorithm so that the system knows how to react positively (i.e., stop the iteration) in the event of the afore-mentioned occurrence. This leads to the design of the stopping criterion as proposed in the study. The main purpose of the stopping criterion used in this study is to stop the simulation of the AFCAIL scheme (for a two-link planar manipulator) automatically if a set of pre-defined stability and accuracy conditions are fulfilled.

The estimated inertia matrix can then be obtained when the system is deemed to be performing excellently and with a high degree of stability and accuracy. Generally, the design of stopping criterion has to take into account one or more of the five aspects below [17]:

(i) ***The convergence test***

A convergence test is the gradual decreasing of error until it reaches a certain minimal value or zero depending on the accuracy requirements of the test. When the error reaches an acceptable range, the iterations are stopped.

(ii) ***The iterations count limit***

It is common practice to limit the number of iterations to prevent situations where the iterative system exhibits no progress towards the solution or progress that is unacceptably slow. This would be a waste of time and seemingly an unsystematic and impractical approach.

(iii) ***Resources to perform/continue the algorithm***

Insufficient-resource errors occur when the iterative solver does not have enough workspace to perform its operations. Therefore the size of the workspace has to be checked before beginning the iterations. However, with the highly advanced computers of today, this problem can easily be tackled.

(iv) ***Breaking down of the algorithm***

This is the mathematical condition when the iterative algorithm involves division by zeros that is undefined.

(v) ***External error***

External errors are errors that are 'outside' the iterative system. Being a subsystem, the iterative learning interacts with other subsystems from which it receives and sends out signals to. Errors in these signals can cause a breakdown in the learning algorithm.

Based on the past simulation results of the AFCAIL scheme for a two-link horizontal planar manipulator, a design of the stopping criterion that is based on the convergence test is chosen as the most appropriate. This is because the iterative learning algorithm computes the estimated inertia matrix, which is a function of the trajectory error that is observed to be gradually decreasing with time. Thus, the logical choice would be to base the design of the stopping criterion on the sum-squared track error from Eq. (8). The following criteria have been proposed for the AFCAIL scheme:

(i) ***Criterion 1 – Trajectory error***

The trajectory error magnitude converges towards zero and goes below a magnitude of n m determined by the requirements of the system.

(ii) ***Criterion 2 – Length of time, t***

The trajectory error remains continuously within this range $0 \leq TE_k \leq n$ for a specified period of time, t that will be based on the time taken for each cycle, *tstop*. For example, the time required is $t = a \times tstop$ where a is the number of cycles.

The values of n and a are not specified as yet, to create a more general stopping criterion that can be implemented into any other AFCAIL scheme. To illustrate the idea of the proposed stopping criterion design, a reference should be made to Figure 5 where a flow chart of the system is shown.

5.1 Stopping Criterion Model

The proposed stopping criterion model is developed using MATLAB and SIMULINK tools. Having known the requirements of the system, the task at hand is to design a subsystem that is able to interact with other subsystems and ensure that an appropriate level of performance is maintained. The proposed model is shown in Figure 6.

The track error of Eq. (8) is fed into the sub-system via the *In* port as shown in the figure. To obtain a positive value, it is passed through an *Abs* function. Referring to Section 5.0, for criterion 1, the trajectory error is handled by a relay switch. For example, the range of the track error set for the system's performance deemed to be acceptably accurate and stable is $0 \leq TE \leq 0.0002$ m. Thus, the accuracy of the tracking task is limited within the range. The relay switch dialog box is as shown in Figure 7.

Figure 7 also shows two conditions that are taken into account by the relay switch in the form of two different outputs considered, one at a time. When the value of TE goes below 0.0002 m, the switch is turned off thus sending out an output signal of 1. At any other time, if the value of TE rises above 0.0002 m, the switch will be turned on giving out an output signal of 0. It is important to note that the frequency of outputs from the

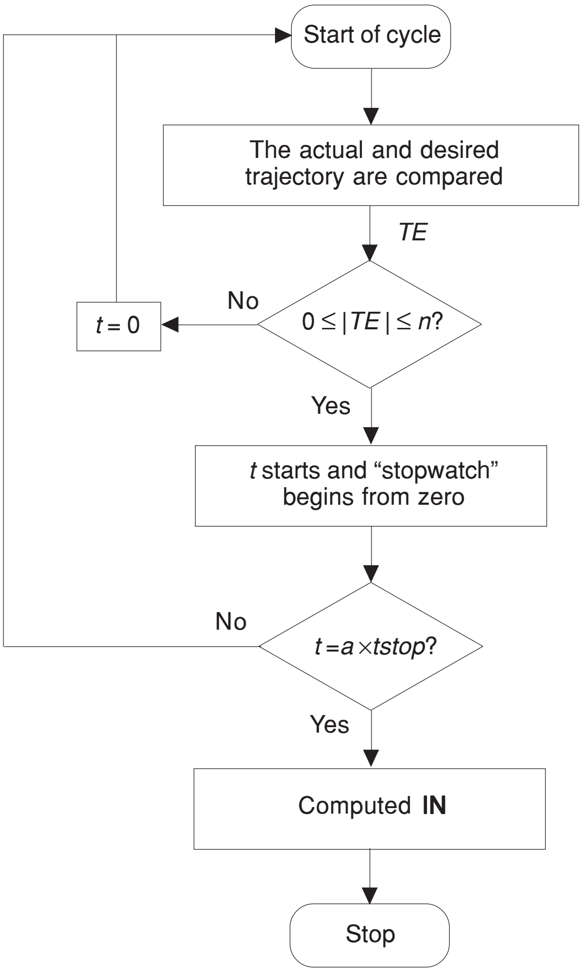


Figure 5 Flow chart of the proposed stopping criterion

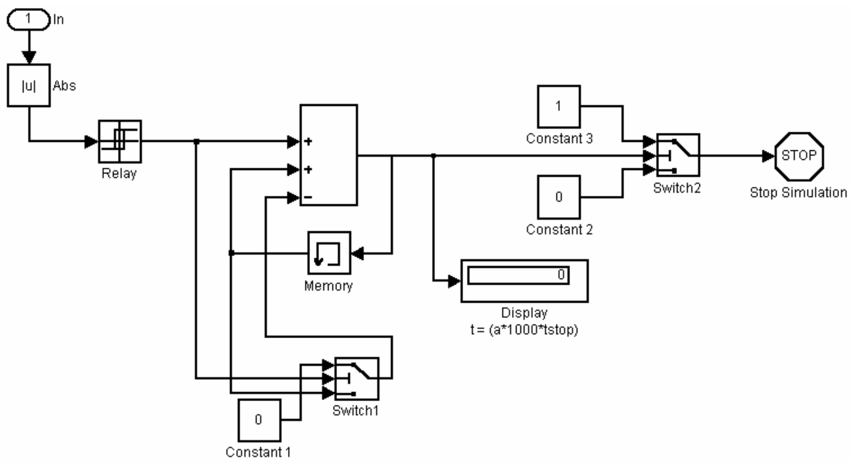


Figure 6 Proposed stopping criterion model

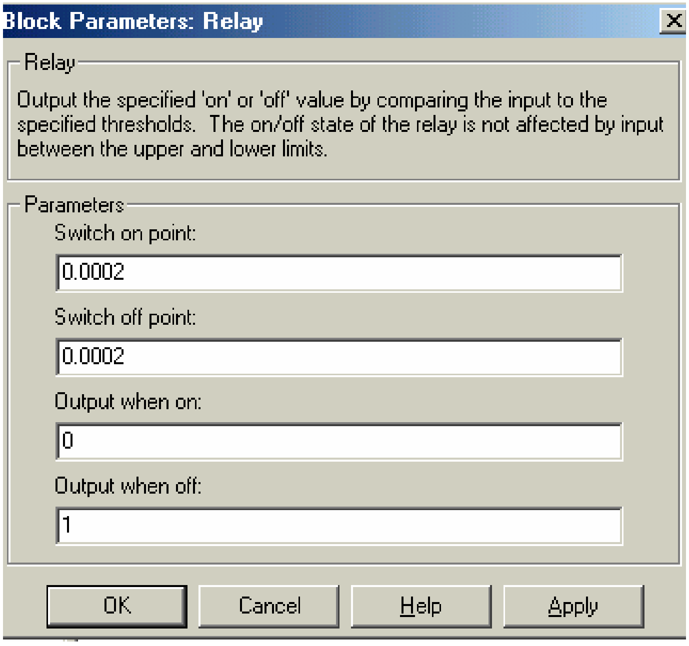


Figure 7 Relay switch dialog box

relay switch depends on the sampling time of the track error system, *e.g.*, if the sampling time is 0.001 s (track error values are updated for every 0.001 s), the relay switch will be sending outputs for every 0.001 s.

5.2 Stopwatch Mechanism

The output signals from the relay switch will then be passed on to a summing junction and also to *Switch 1* (Figure 6). When the input to the junction equals to 1 (which is desirable), two things will happen:

(i) The value 1 will be added to the previous sum stored in the memory block (for the first time, the sum in the memory block is equal to 0).

(ii) *Switch* 1 will pass on the constant 0 to be added at the summing junction and therefore there will be no effect on the summing operation at this point in time. A threshold value of the switch is set to 0.5, which means if the input is greater or equal to 0.5, the upper channel of the switch will be switched on and the lower channel off and vice versa. The dialog box of the switch is shown in Figure 8.

The adding of 1 s will continue throughout the simulation until the track error goes above the desired range, thus giving the output 0 from the relay switch. When this happens,

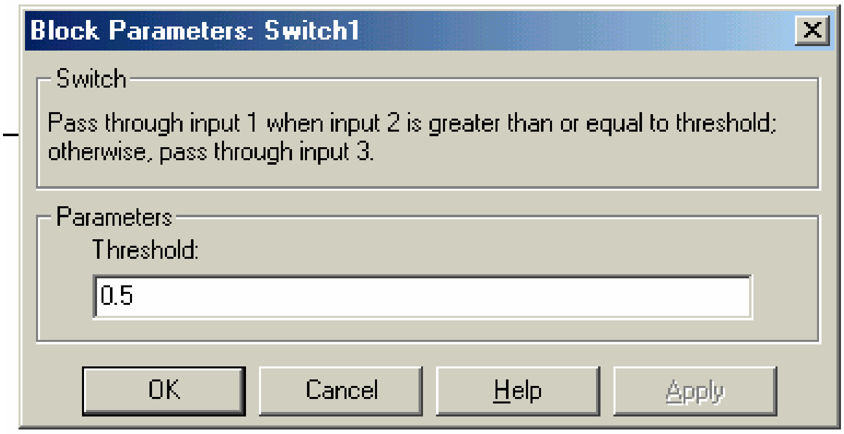


Figure 8 Dialog box of switch threshold (set at 0.5)

(i) *Switch* 1 will turn on the lower channel, thereby sending the memory value to the subtraction input of the summing junction. This will cause the previous cumulative value to be set back to zero.

(ii) The input 0 from the relay switch to the summing junction will not affect the summation.

The above-mentioned mechanism is similar in operation to how a stopwatch works. When the value of the track error begins to 'hit' the lower limit of the desired range, the stopwatch is started and kept running whenever the values stay in that range. On the other hand, if at any moment, the value reaches the upper limit of the desired range, the stopwatch is stopped and then reset back to zero. This process might have to be repeated until the stopping criterion are met depending on the behaviour of the track error pattern.

5.3 Stopping the Simulation

As mentioned before in Section 5.0, in order for the simulation to come to a stop, the track error needs to be in the desired range for a certain period of time and that it is a function of the number of repeated cycles the robot is performing its task. The purpose of including *Switch* 2 (Figure 6) is to compare the cumulative sums of 1 s with a 'time criterion', t (merely a representation of time) that is given by the equation,

$$t = a \times t_{cycle} \times (1/t_s) \tag{11}$$

where a is the number of cycles the robot is performing desirably, t_{cycle} is the time for the robot to complete a task in one cycle, and t_s is the sampling time.

The value of a has to be at least 2 (indicating two cycles) in order to enable a clearer observation of the **IN** values and the track errors computed by the system. When the sum of 1 s equals t, *Switch* 2 (that has a threshold value as described by Equation (11)), will switch on the upper channel of the switch allowing the constant 1 to pass through. If the sum of 1 s is below t, the constant 0 will be passed through instead.

The final block is the *STOP* block that stops the simulation when its input is non-zero. Thus when the criterion, sum of 1 s equals t, the simulation will be stopped automatically.

6.0 SIMULATION RESULTS

Simulation was performed on the AFCAIL scheme to investigate the effectiveness of the stopping criterion model. Two loading conditions were considered; one without disturbances while the other with a set of introduced disturbances.

The parameters of the simulation used in the study were:

Magnitude of the desired track error goal, $TE \leq 0.0002$ m
Number of desirable cycles, $a = 3$ (value must be at least 2)
Time for one cycle, $t_{cycle} = 3.142$ s
Maximum total simulation time = 25 s
Sampling time, $t_s = 0.001$ s

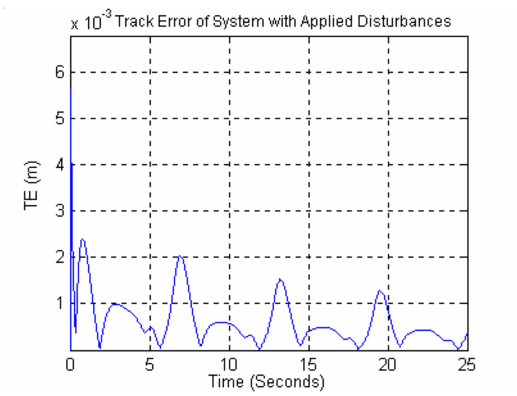
Initial conditions (for the iterative learning algorithm):
$IN_{11} = 0.004$ kgm^2 and $IN_{22} = 0.002$ kgm^2

Applied simultaneous disturbances to the robot arm:
Harmonic force, $F_h = h \sin t$ where the amplitude, $h = 40$ N
Spring force, $F_s = kx$ where the spring constant, $k = 150$ N/m

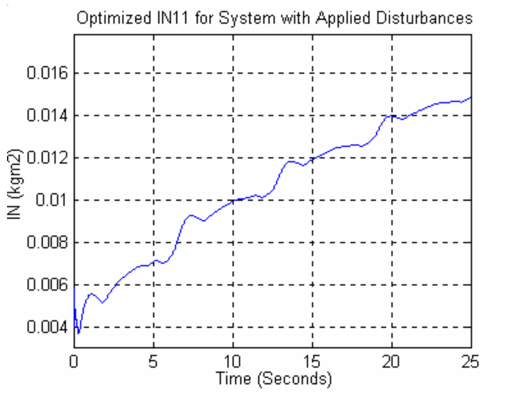
The results of the simulation can be seen in Figures 9 and 10. A summary of the results is shown in Table 1.

7.0 ANALYSIS AND DISCUSSION

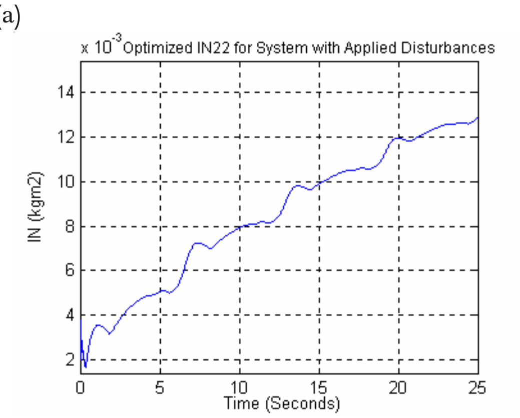
From graphs of Figure 9, it is observed that the system does not meet the accuracy requirements determined in the simulation parameters. Thus, the simulation were executed for the full 25 s (or approximately 8 cycles) without stopping, based on the criterion set. This provides a clearer picture for the designer of the control system who has to make necessary alterations to the system if the control scheme is required to perform excellently in the presence of the introduced disturbances. On the other hand, the graphs of Figure 10 show that that the system without disturbances readily meets the set accuracy criterion. The magnitude of the trajectory error remains below 0.0002



(a)



(b)



(c)

Figure 9 Simulation results with the applied disturbances

m for three continuous cycles, enabling the simulation to be stopped automatically. The total simulation time is approximately 9.775 s. Thus the system is said to be fulfilling criterion 1 (refer to Section 5.0), where the track error goes below 0.0002 m after about 0.35 s of operation from the starting time.

The set of suitable **IN** values for each good cycle (refer to Table 1) can be obtained using the 'backward method'. The **IN** values for the third good cycle are obtained by taking the **IN** values (both IN_{11} and IN_{22}) for the period of time, t_{cycle} (one cycle) counted backwards from the total simulation time, the time range being 6.633 – 9.775 s. To obtain the **IN** values for the second good cycle, the values for the period of time 3.492 – 6.683 s are taken. The **IN** values from the period 0.35 – 3.492 s are considered the appropriate **IN** as illustrated in Figures 10(b) and 10(c). The sets of **IN** values are therefore more systematically obtained through this method. Another observation is that the **IN** values are increasing non-linearly with time from the initial conditions. Besides that, the **IN** values for the system with applied disturbances, are generally larger than the values for the system without disturbances.

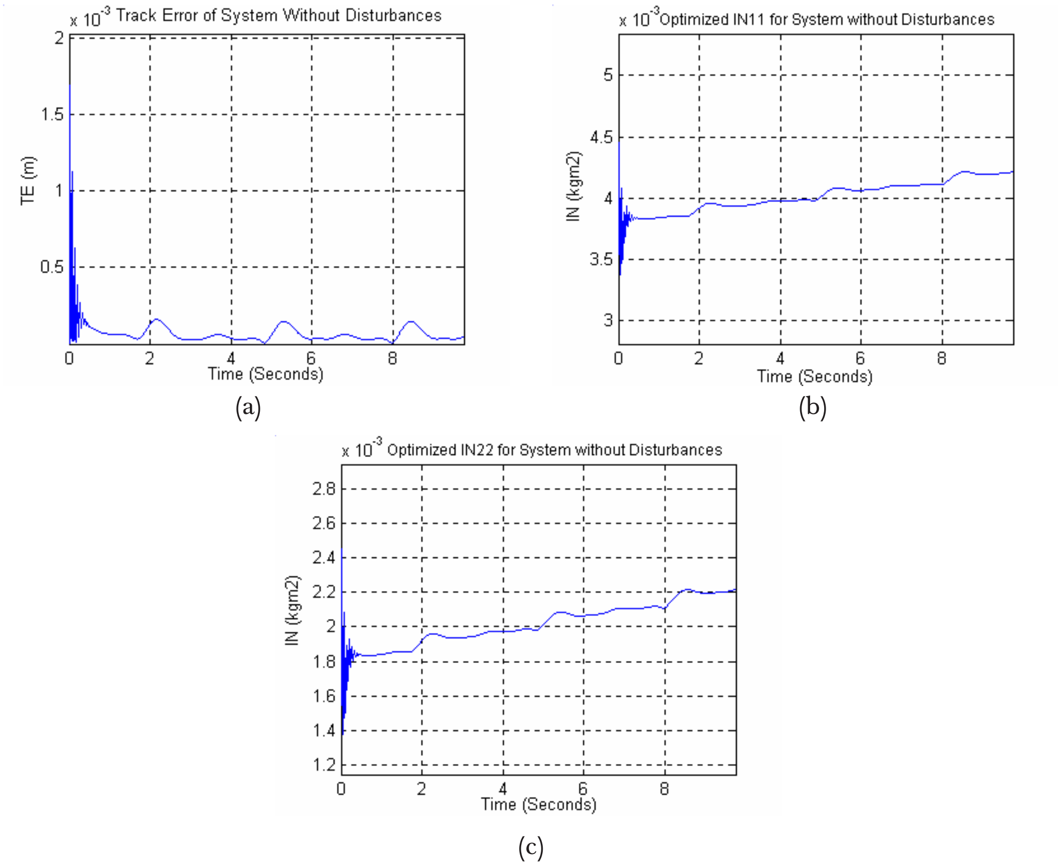


Figure 10 Simulation results without disturbances

Table 1 Summary of the simulation results

DISTURBANCES	SIMULATION TIME (s)	APPROXIMATE SUITABLE IN RANGE ($\times 10^{-3}$ kgm^2)					
		IN_{11}			IN_{22}		
		Cycle 1	Cycle 2	Cycle 3	Cycle 1	Cycle 2	Cycle 3
F_h and F_s	25	–	–	–	–	–	–
None	9.775	3.81-3.94	3.94-4.08	4.08-4.23	1.83-1.95	1.95-2.09	2.09-2.23

8.0 CONCLUSIONS

The stopping criterion proposed in the study was indeed effective in stopping the simulation of the AFCAIL scheme automatically when conditions related to the accuracy and stability of the system are met. There are two main criteria that need to be determined, namely, the suitable range of the desired track error and the length of time when the error is said to remain within the acceptable range. The proposed stopping criterion model has been shown to provide a systematic method in obtaining suitable values of **IN**. In addition to that, it can also be used as a tool to study the effects of changes made to the control scheme related to the parameters, robotic tasks, operating and loading conditions particularly when the simulation results are not easily predictable. These could be a subject of future works that could be carried out.

ACKNOWLEDGEMENTS

The authors would like to thank the Ministry of Science and Technology and the Universiti Teknologi Malaysia for providing the financial support and facilities. This research was supported using an IRPA grant of Vot. No. 72361.

REFERENCES

[1] Astrom, K. J., and T. J. McAvoy. 1992. Intelligent Control: An Overview and Evaluation. In David A. White, Donald A. Sofge, Eds., *Handbook of Intelligent Control : Neural, Fuzzy and Adaptive Approaches*, New York: Van Nostrand Reinhold, 3-34.

[2] Tzafestas, S. G., 1991. *Engineering Systems with Intelligence – Concepts, Tools and Applications*, London: Kluwer Academic Publishers.

[3] Saad, M., P. Bigras, L. A. Dessaint, and K. Al-Haddad. 1994. Adaptive Robot Control Using Neural Networks. *IEEE Transactions on Industrial Electronics*. 41(2): 173-181.

[4] Gardner, J. F., A. Brandt, and G. Luecke. 1993. Applications of Neural Networks for Coordinate Transformations in Robotics. *Journal of Intelligent and Robotic Systems*. 8: 361-373.

[5] Ohnishi, K., M. Shibata, T. Murakami. 1996. A Unified Approach to Position and Force Control by Fuzzy Logic. *IEEE Transactions on Industrial Electronics*. 43(1): 81-7.

[6] Jamshidi, M., N. Vadiee, and T. J. Ross. 1993. *Fuzzy Logic and Control*. New Jersey: Prentice-Hall International, Inc.

[7] Arimoto, S., S. Kawamura and F. Miyazaki. 1985. Hybrid Position/Force Control of Robot Manipulators Based on Learning Method. *Proc. of Int'l. Conf. on Advanced Robotics*. 235-242.

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

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

[10] Moore, K. L., 1993. *Iterative Learning Control For Deterministic Systems*. London: Springer Verlag.

[11] Moore, K. L., M. Dahleh, and S.P. Bhattacharyya. 1990. Adaptive Gain Adjustment for a Learning Control Method for Robotics. *Procs. of Int'l Conf. On Robotics and Automation*. Ohio, 2095-2099.

[12] Bondi, P., G. Casalino, L. Gambardella, 1988. On the Iterative Learning Control Theory for Robotic Manipulators. *IEEE Journal on Robotics and Automation*. 4(1): 14-22.

[13] Hewit, J. R., and J. S. Burdess. 1981. Fast Dynamic Decoupled Control for Robotics Using Active Force Control. *Mechanism and Machine Theory*. 16(5): 535-542.

[14] Mailah, M. 1998. *Intelligent Active Force Control of a Rigid Robot Arm Using Neural Network and Iterative Learning Algorithms*. Ph.D. Thesis, University of Dundee.

[15] Mailah, M. and N. I Ab. Rahim. 2000. Intelligent Active Force Control of A Robot Arm Using Fuzzy Logic. *Procs. of IEEE Int'l Conf. on Intelligent Systems and Technologies*. TENCON 2000, Kuala Lumpur, II: 291-297.

[16] Asada, H., and J. E. Slotine. 1986. *Robot Analysis and Control*. New York: John Wiley and Sons Inc.

[17] John, K. W. 1995. *Stopping Criterion*, http://www-users.cs.umn.edu/~kewu/iters/node16.html, modified: 2 June 1999.