Research note

Active disturbance rejection control of a parallel manipulator with self learning algorithm for a pulsating trajectory tracking task

A. Noshadi, M. Mailah *

Department of System Dynamics and Control, Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

Abstract A novel and robust intelligent scheme is proposed to control a highly non-linear 3-RRR (revolute-revolute-revolute) planar parallel robotic manipulator, via an Active Force Control (AFC) strategy that is embedded into the classic Proportional-Integral-Derivative (PID) control loop. A PID-type Iterative Learning (IL) algorithm, with randomized initial conditions, is incorporated into the AFC loop to approximate the estimated inertia matrices of the manipulator adaptively while the manipulator is tracking a prescribed pulsating trajectory in the presence of harmonic disturbances. The IL algorithm employs a stopping criterion, which is based on tracking error, to stop the learning process when the desired error goal of the system is reached, to signify a favorable controlled condition. A numerical simulation study was performed to verify the robustness of the proposed methodology in rejecting disturbances, based on given loading and operating environments. The results of the study reveal the superiority of the proposed system, in terms of its excellent tracking performance compared to the AFC, with crude approximation techniques, and Proportional-Integral-Derivative (PID) counterparts.

1. Introduction

In recent years, researchers have proposed new types of mechanisms called parallel manipulators, in order to perform specific tasks that common serial robots are not capable of performing. Some advantages of parallel robots in comparison with serial ones, which have attracted the attention of researchers, include: higher rigidity, greater carrying payload capacity and higher accuracy. Another interesting advantage of parallel robots is the opportunity to locate the actuators on the base, which is suitable for fast and accurate operations. However, the kinematics and dynamics of parallel manipulators are very complex due to their inherent closed-loop kinematic chains, and also parallel manipulators have a relatively smaller workspace in comparison with serial ones.

There are relatively limited resources related to the control of parallel manipulators available in the literature. Ghorbel et al. showed that common methods used for control of serial robots can also be utilized for parallel robots [1]. Amiratet et al. applied a classic Proportional–Integral–Derivative (PID) controller to a 6 Degree-Of-Freedom (6-DOF) parallel robot with C3 links related to equestrian gait simulation [2]. Stan et al. applied a fuzzy controller to a 3-DOF medical parallel robot and obtained more robust results via the fuzzy controller compared to the classic PID controller [3]. Wu and Wang presented a motion control of a 2-DOF parallel manipulator as a machine tool. They used a method for tuning the PID controller gains and combined it with a low-pass filter to improve the system performance and reduce track error when the system moves with large acceleration [4].

Since parallel manipulators have been designed to perform fast and accurate tasks, substantial research work has been devoted to designing a robust controller, to cope with the external disturbances and parameter uncertainties that are deemed to have considerable impact on system performance. As a matter of fact, in an environment which typically involves the interaction of various forms of disturbance, operating condition, parametric change and uncertainty, a robust control scheme is pivotal in guaranteeing the performance of the manipulator to actually execute the desired task to the highest

* Corresponding author. Tel.: +60 75534735; fax: +60 75566159.
E-mail address: musa@fkm.utm.my (M. Mailah).

degree of accuracy possible. Kim et al. designed a robust tracking control design for a 6-DOF parallel manipulator in the presence of disturbances and time-varying uncertainties [5]. The controller was based on the Lyapunov approach and guaranteed a practical stability. Tao et al. suggested an adaptive robust posture control on a pneumatic muscle-driven parallel manipulator with actuation redundancy [6]. A robust nonlinear task space controller equipped with a friction estimator for a 6-DOF parallel manipulator was introduced by Kim et al. [7]. The proposed scheme was based on the Lyapunov redesign method to guarantee practical stability under parameter uncertainties such as inertia, measurement error, modeling error and friction. The results showed that the proposed robust nonlinear task space control scheme was superior to the classic PID controller, and demonstrated its viability in real-world application. Another common approach to compensate for system uncertainties is through Sliding Mode Control (SMC). However, the sliding mode control contains a discontinuous function which can easily cause a chattering phenomenon. Chattering may excite the unmodeled high-frequency dynamics so as to degrade the performance of the system [8]. Active force control (AFC) is yet another form of a robust disturbance rejection controller, which was first proposed by Hewit and Burdess in the early eighties [9]. They applied the AFC technique to a robot arm in the presence of various forms of disturbance and uncertainty. They showed that by using this method, a system subjected to environmental uncertainties, disturbances or any other changes in the system parameters could remain stable and robust. The effectiveness and robustness of the AFC method as an active disturbance rejection control scheme is rigorously verified in the literature [10–12]. Other disturbance rejection controllers based on the PID method can be seen in [13,14].

In an effective AFC scheme, it is fundamental to determine the appropriate value of the estimated inertias used in the AFC loop, to increase system performance in tracking the prescribed trajectory while rejecting disturbances. In this paper, an Iterative Learning (IL) algorithm, based on works described in [15,16], with a suitable stopping criterion, was used to estimate the inertial parameters (**IN**) of the manipulator system in the AFC loop. Thus, a robust control scheme is proposed that is designed to control a 3-RRR planar parallel manipulator. The scheme is known as Active Force Control And Iterative Learning (AFCAIL), in which the IL algorithm is used to compute the estimated inertia matrices of the AFC scheme adaptively, while the manipulator is forced to track a prescribed trajectory and subjected to disturbances. The proposed stopping criterion is designed based on trajectory tracking error and is deliberately introduced into the IL algorithm to halt the learning process when the desired error goal has been achieved. The proposed algorithm is able to start the learning process again if another disturbance is applied to the system and a new **IN** value was required for the AFC loop. The effectiveness and robustness of the AFCAIL scheme is tested via a rigorous simulation study in which the system is commanded to track a prescribed pulsating trajectory in the presence of time varying harmonic disturbances. The results clearly show that the proposed scheme provides a much superior trajectory tracking capability compared to the conventional controller.

For control purposes, in order to solve the dynamic models of multibody systems, first the system is considered as an equivalent tree-structure, and then using d'Alembert's principle or Lagrange multipliers, system constraints can be obtained [17]. Nevertheless, there are other different methods
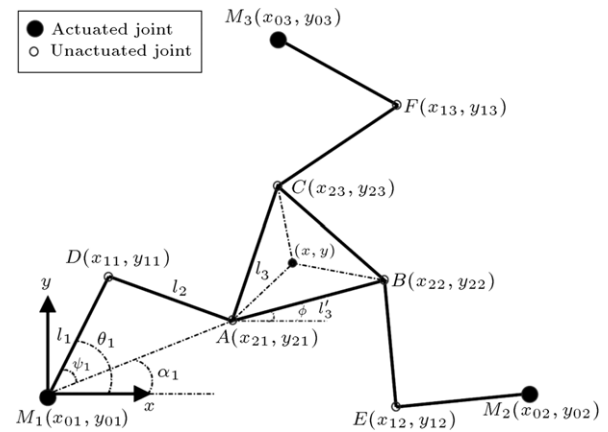

Figure 1: General form of planar 3-DOF parallel manipulator.

in the literature that can be used to solve the dynamics of closed kinematic chains, such as virtual work [18,19], Newton–Euler equations [20] and Hamilton's principle [21]. In the present paper, the Natural Orthogonal Complement (NOC) method is used for solving the dynamics of the 3-RRR robot [22].

2. Modeling of 3-RRR planar parallel manipulator

A 3-DOF planar parallel manipulator is depicted in Figure 1. The system has nine revolute joints, i.e. three actuated joints fixed to the base and six unactuated joints that form three closed kinematic chains. The triangular plate, which is located in the middle of the figure, is supposed to be the end-effector of the system. The manipulator is symmetric, and also each leg of the manipulator has the same length.

2.1. Inverse kinematics

The aim of solving the inverse kinematics is to obtain the angles of active joints from the position and orientation of the end-effector, which are essential for the position control of the parallel manipulators. The inverse and direct kinematic analyses of the manipulator have been derived and presented in [23]. Solving the inverse kinematics is very useful, since robot tasks are commonly formulated in terms of the end-effector's specified position and motion.

2.2. Direct kinematics

Most parallel manipulators have easier inverse kinematic solutions compared to serial counterparts, while direct kinematic problems are very challenging in the majority of parallel robots. In the direct kinematic problem, with a given position of joint angle, the position and direction of the end-effector can be obtained without considering the forces or masses that cause the motion.

Unlike serial manipulators, there is no unique solution for direct kinematics of parallel robots. Although several computational methods have been proposed to determine the pose of the end-effector in Cartesian space, these methods are more often than not very time consuming. For direct kinematics of a 3-RRR manipulator, Gosselin and Angeles showed that a maximum of six results is possible [23]. However, due to the trajectory tracking procedure, only one of the solutions is deemed correct.
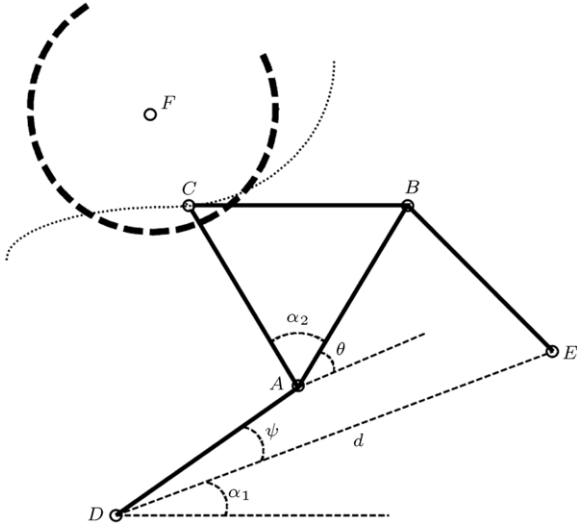
Figure 2: Equivalent four bar linkage.

In this study, the previous position of the platform is used as the initial condition, which enables us to have a better chance of finding the correct solution. From Figure 1, if the three actuated joint angles are specified, the positions of points D, E and F in Figure 2 can be easily computed. In fact, the chain DABE could be considered as a four-bar linkage, as illustrated in Figure 2. Then, point C in Figure 2 is a point of the coupler link generating a coupler curve. A solution for the closure of the whole kinematics chain is determined via contraction of the circle and the coupler curve. Direct kinematics of the manipulator is vital for addressing the system control objectives. Note that further relevant kinematic expressions applied to a 3-DOF parallel manipulator can be found in the Appendix.

2.3. Dynamics of 3-RRR parallel manipulator

Solving the dynamics of the system is necessary for studying robot control strategies. To control the manipulator, the direct dynamics of the system are modeled and simulated, in order to predict the motion of the manipulator, given the driving forces of the system. A general form of manipulator dynamics is expressed as follows:

$$\mathbf{T}^T\mathbf{M_{total}}\mathbf{T}\ddot{q} + \left(\mathbf{T}^T\mathbf{M_{total}}\dot{\mathbf{T}} + \mathbf{T}^T\boldsymbol{\Omega}\mathbf{M_{total}}\mathbf{T}\right)\dot{q}^a - \mathbf{T}^T\mathbf{W}^g = \boldsymbol{\tau}^a, \quad (1)$$

where q^a, $\dot{q}^a$, and $\ddot{q}^a$ are corresponding displacement, velocity and acceleration of three actuated joints, respectively. Also, $\mathbf{T}$ is the NOC matrix and $\mathbf{M}$ and $\boldsymbol{\Omega}$ are block-diagonal matrices defined as:

$$\mathbf{M_{total}} = \mathrm{diag}(\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_r), \quad (2)$$

$$\boldsymbol{\Omega} = \mathrm{diag}(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \ldots, \boldsymbol{\Omega}_r). \quad (3)$$

Apparently, the dimension of both matrices is $6m \times 6m$ (m is number of joints). In the dynamic model, the generalized inertia wrench, $\mathbf{w}^g$, can be evaluated in a straightforward manner, if the twist and time derivative of each body of the system are known. Evaluation of the generalized gravity wrench is rather simple and can be expressed as:

$$\mathbf{W}^g = \begin{bmatrix} \mathbf{0} & m_1\mathbf{g} & \mathbf{0} & m_2\mathbf{g} & \cdots & m_r\mathbf{g} \end{bmatrix}^T. \quad (4)$$

It should be noted that $\mathbf{0}$ in Eq. (4) denotes the 3-D zero vector, m_r is the mass of the rth movable rigid body, and $\mathbf{g}$ is the
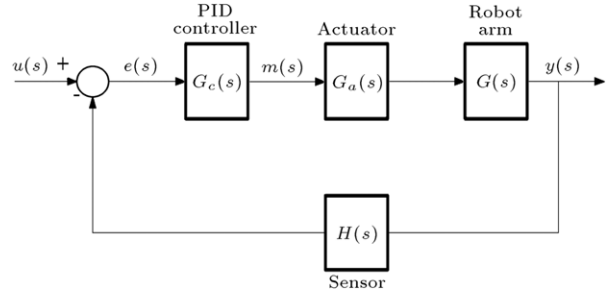

Figure 3: General scheme of a PID controller.

gravity acceleration vector. Obviously, $\mathbf{g}$ is a 3-D constant vector in a base coordinate frame. Note also that $r = 1, 2, \ldots, 7$ denotes the number of movable rigid bodies in the system. For manipulator applications, the dynamic model of a system is usually represented in joint space. Substituting equations that are previously introduced, one obtains the model in terms of the independent joint coordinates as follows:

$$\mathbf{M}(q)\ddot{q}^a(t) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{q}^a + \mathbf{G}(q) = \boldsymbol{\tau}^a, \quad (5)$$

where:

$$\mathbf{M}(q) = \mathbf{T}^T\mathbf{M_{total}}\mathbf{T}, \quad (6)$$

$$\mathbf{C}(q \cdot \dot{q}) = \mathbf{T}^T\mathbf{M_{total}}\dot{\mathbf{T}} + \mathbf{T}^T\boldsymbol{\Omega}\mathbf{M_{Total}}\mathbf{T}, \quad (7)$$

$$\mathbf{G}(q) = -\mathbf{T}^T\mathbf{W}^g, \quad (8)$$

where $q_i = [q_1, \ldots, q_9]^T$ is the generalized coordinate; q_i for $1 \leq i \leq 3$ denotes the angles of active joints and q_i for $4 \leq i \leq 9$ represents relative angles of passive joints. In Eq. (5), $M(q) \in \mathfrak{R}^{3\times3}$ is the inertia matrix, $C(q, \dot{q}) \in \mathfrak{R}^{3\times3}$ is the coefficient matrix of Coriolis and centrifugal forces, and $\tau^a \in \mathfrak{R}^{3\times1}$ marks the required torques of actuated joints. The direct dynamic problem is defined as obtaining $\ddot{q}^a$, when the values of q, $\dot{q}$ and τ^a are given. In this study, direct dynamics are used to simulate the manipulator. However, in the simulating process, the parameters related to τ^a, angles of active joints (q_i^a), and their respective angular velocities ($\dot{q}_i^a$), are considered. Hence, direct kinematics should be utilized to calculate q and $\dot{q}$. In the direct dynamic problem, Eq. (5) should be solved as a differential equation using a suitable numerical method.

3. Controller design

3.1. Active Force Control (AFC)

Prior to developing the AFC scheme, a conventional PID controller, such as that shown in Figure 3, needs to be first designed. It is a well-known fact that such a controller does not require specific knowledge of the robot's model and are computationally simple, as well as reasonably robust. These factors make them attractive for implementation in a real robotic system.

When there is an error (e) between the feedback and input signals, the control mechanism will compute the energy needed by the system to eliminate this error. In short, the system has to have the error signal or deviation to initiate the compensation action of the controller, without which the system would not effectively operate. The main drawbacks of this controller are the problems related to efficient acquisition of the PID gains to maintain robust system performance, particularly in the presence of disturbances. This issue is still a subject of active
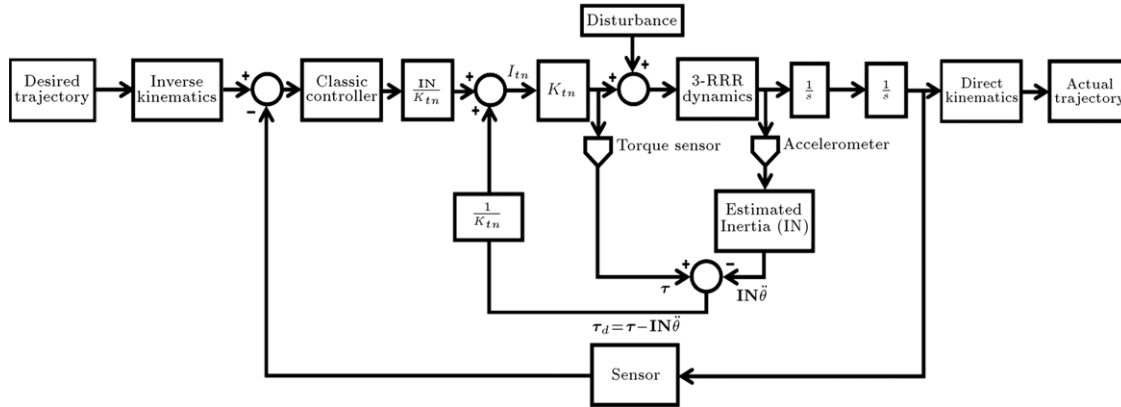
Figure 4: Schematic diagram of the general AFC scheme.

research, and for the purpose of the study, the controller gains were assumed to be satisfactorily tuned via a heuristic strategy. It can be later demonstrated that the gains can be made relatively unaffected by complimenting the loop with another control method, even in the presence of disturbances. Thus, the next important step is to incorporate the AFC mechanism into the PID control scheme by simply adding the AFC loop in series with the classic PID control loop, as depicted in Figure 4. This constitutes a 2-DOF controller that could potentially and considerably enhance the system overall performance.

The main aim of the AFC scheme is to provide an effective compensation for dynamics that arise from disturbances and uncertainties, without reducing the performance of the overall system. It thereby renders the system very robust and stable, provided that the AFC mechanism is appropriately designed and implemented. The majority of the adaptive controllers (designed as robust schemes) in the literature typically need a linearized model of the system and previous knowledge about the disturbances bounds. AFC, on the other hand, is easy to understand and very applicable (to real-time dynamical systems), since it makes use of a simple algorithm that is based on estimated or measured values of relevant parameters. Therefore, it does not necessitate large computation and mathematical manipulation in order to execute the disturbance rejection capability.

From *Newton*'s second law of motion for rotational bodies, the sum of all torques applied to the system is equal to the product of the mass moment of Inertia (I) and the angular acceleration (α) of the system:

$$\Sigma\tau = I\alpha. \tag{9}$$

When the disturbance, τ_d, is considered and the mass is rotating with a joint angle, θ, Eq. (9) becomes:

$$\tau + \tau_d = I\ddot{\theta}, \tag{10}$$

where:

   τ is the applied torque to the actuated joints
   τ_d is the total of applied disturbance torques to the actuated joints
$\theta, \ddot{\theta}$ are the actuated joint angles and angular acceleration, respectively.

The disturbance torque, τ_d^*, can be approximated through the AFC loop as follows:

$$\tau_d^* = \tau - \mathbf{IN}\ddot{\theta}. \tag{11}$$

The variable **IN** is the estimated inertia matrix that can be obtained by crude approximation or other intelligent methods, such as iterative learning, fuzzy logic and so forth. τ is the measured applied control torque that can be readily obtained using a torque sensor or indirectly through a current sensor and the measured angular acceleration, i.e. $\ddot{\theta}$ using an accelerometer. From Eq. (11), it is clear that if the total applied torque to the system and angular acceleration of each actuated joint are accurately obtained, and the estimated inertial matrices (**IN**) appropriately approximated, then the total of disturbance torque can be computed via the AFC loop, without having to acquire exact knowledge of the actual magnitude of the disturbances.

It is important to note that at this juncture, with incorporation of the AFC-based scheme into the PID controller, the gains need not be retuned or adjusted, even in the wake of disturbances or uncertainties due to the compensating effect of the AFC algorithm, which is in stark contrast to that of the pure PID scheme. A case study involving its implementation in a complex parallel manipulator is presented and subsequently verified in the proposed study. In addition, an Iterative Learning (IL) algorithm with a stopping feature was employed to compute the **IN** matrices used in the AFC loop iteratively, and based on the trajectory track error of each actuated joint of the 3-RRR parallel manipulator. The **IN** matrices shall be continuously updated via the IL algorithm, in which the manipulator is deemed to have gradually learnt to perform the prescribed task in the process.

3.2. Iterative learning control

The Iterative Learning (IL) method is typically concerned with a mathematical algorithm that contains a repetitive operation, which is mainly intended to provide a mechanism that could improve system performance with time. In other words, IL control is a learning method to generate an optimal output response as close as possible to the desired output, i.e. the error approaching zero datum, as time increases. From the literature, it is extensively accepted that the first idea of Iterative Learning Control (ILC) formulation was proposed by Uchiyama in 1978 [15] and later aggressively researched and implemented by Arimoto et al. [16]. They proposed a number of learning algorithms and proved their convergence, stability and robustness. Since then, ILC has been successfully applied to different repetitive systems, such as packaging systems, injection moulding and robot manipulators [24]. An IL algorithm is said to be efficient if it is able to estimate the
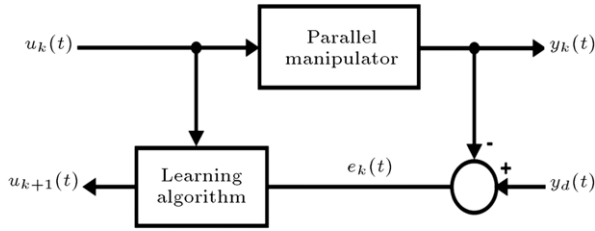
Figure 5: Iterative learning algorithm.

next input value, and in doing so, making system performance better and better in successive trials or as time increases. Often, the learning algorithm is independent of any prior knowledge of the system model. In fact, it is a self learning algorithm and is not dependent on the prescribed task to be executed. Thus, there is no need to reconfigure the algorithm when the system is operating. Convergence should also be achieved even if there is an uncertainty in the plant model [25]. It is one of the most desirable properties in an IL controller, to ensure that the dynamic system is in a stable and desirable state, as the algorithm progressively iterates. The stability of a system may be affected at some point if iteration is allowed to continue indefinitely. It is thus useful to have an appropriate stopping criterion to stop the learning process, once a satisfactory performance of the dynamic system is actually achieved. If needs be, the learning process could be retriggered (and accordingly restopped) if the system is further subjected to different forms of disturbance, parametric change or other operating and loading conditions. In most IL algorithms, it is the 'stop time' that is typically selected as the stopping criterion of the algorithm. Thus, the learning process will cease processing at a specific time to prevent the system from instability, and more often than not, the controller will produce the same output response, even if another disturbance is applied to the system thereafter.

Arimoto et al. proposed a number of learning algorithms and proved their convergence, stability and robustness [16]. In the proposed IL algorithm, for each iteration of the learning process, input signals '$u_k(t)$' are recorded in the memory in order to use them in the next iteration. The error obtained from the current iteration is saved in the memory as '$e_k(t)$',

and the output signal as '$y_k(t)$'. Notice that character 'k' denotes the current value, while time variable 't' may be continuous or discrete. The learning algorithm then evaluates the system's performance error, which can be obtained as:

$$e_k(t) = y_d(t) - y_k(t), \tag{12}$$

where '$y_d(t)$' is supposed to be the desired trajectory of the manipulator and '$y_k(t)$' is the actual trajectory of the manipulator in each iteration. These data are used to compute the new input signal, '$u_{k+1}(t)$', in the next iteration to modify control inputs and reduce system error gradually after some iterations. The basic principle of the IL method can be seen in Figure 5.

Most algorithms proposed by Arimoto et al. [16], in the literature, show that the $(k + 1)$th input to the system can be obtained by kth input plus an error, derivative and integral coefficients of the track error. These mathematical expressions are similar to the description of the classic PID controller. A PID-type learning algorithm used in the study can be expressed as follows:

$$IN_{k+1} = IN_k + \left(\varnothing + \psi \int dt + \Gamma \frac{d}{dt} \right) TE_k, \tag{13}$$

where:

IN_{k+1} is the estimated inertia of the next step for the AFC loop,
IN_k is the current estimated inertia for the AFC loop,
TE_k is the current output track error ($TE = q_{desired} - q_{actual}$),
ϕ, ψ, Γ are the learning parameters.

Figure 6 shows the proposed AFCAIL scheme in which the AFC loop is added in series with the PID controller, with the IL algorithms embedded in the AFC loop to compute the estimated inertias of the three links of the manipulator.

A flow chart of the proposed IL algorithm with stopping criterion is depicted in Figure 7. The data gathered from the desired and actual trajectories were used to estimate the next inertia matrices of the active links of the manipulator via the IL algorithm. The acceptable range of Track Error (TE) to stop the learning process is proposed, considering a fixed time interval. The TE is heuristically selected for the system to perform accurately in the presence of disturbances in the
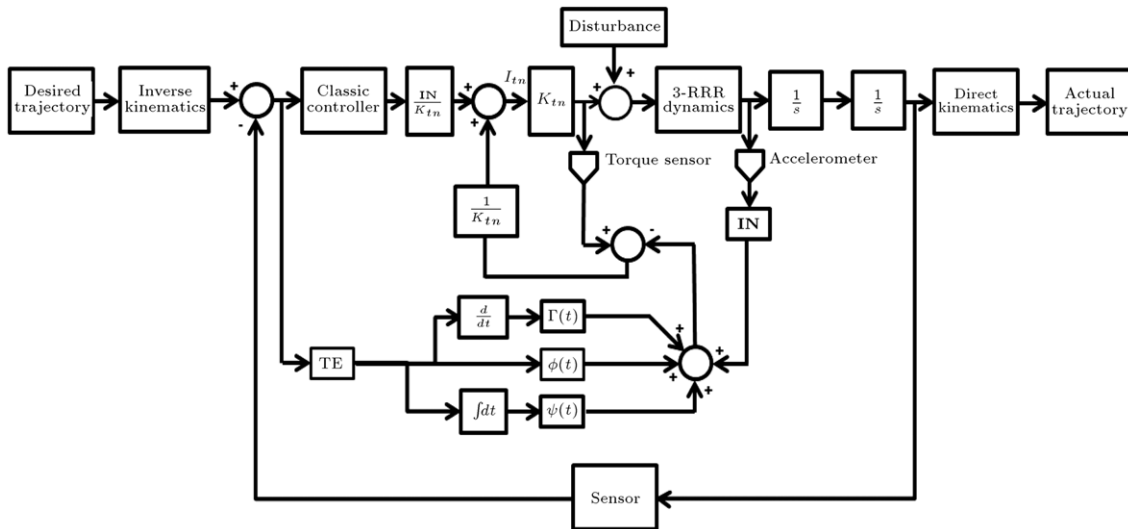

Figure 6: Schematic diagram of the AFCAIL scheme.

range $0 \leq \text{TE} \leq 0.001$ m. This means that the IL loop will keep on changing and updating the values of **IN** in order to reject the applied disturbances to the system. Once track error is below the 0.001 m threshold, the IL algorithm will stop and, hence, an appropriate or 'optimized' **IN** is deemed to have been obtained for these specific operating and loading conditions, until another different disturbance or input is applied to the system, which in turn requires a new **IN** for the AFC loop to counter the current (new) disturbance, continuously and robustly. Therefore, the learning algorithm starts finding a new 'optimized' value of **IN** in the AFC scheme.

4. Simulation

To validate the effectiveness of the proposed control method applied to the 3-RRR manipulator, a desired end-effector trajectory was introduced for the trajectory tracking control problem. Three control schemes were considered for the purpose of benchmarking the system performance, namely, pure PID controller, AFC with crude approximation method and AFCAIL. Note also that the desired trajectory was defined as an irregular pulsating closed loop trajectory in which the manipulator was forced to start from its initial position and return to the original position, according to the following time (t) dependent functions:

$$x_p = 0.6 - 0.1 \cos(0.4\pi t) \quad 0 \leq t \leq 5 \, \text{s}, \tag{14}$$

$$y_p = 0.3 - 0.1 \sin(2\pi t) \quad 0 \leq t \leq 5 \, \text{s}. \tag{15}$$

The MATLAB/Simulink model of the final proposed AFCAIL scheme is shown in Figure 8 for one actuated joint.

Figure 9 shows the snapshots taken from the manipulator's trajectory tracking. From Figure 9, it can be seen that the desired trajectory is within the manipulator's workspace. Also, there is no singularity located at the manipulator's prescribed trajectory.

Controller parameters, loading and operating conditions are as follows:

$l_1 = 0.4$ m, $l_2 = 0.6$ m, $l_3' = 0.4$ m,
$m_1 = m_2 = m_3 = 3$ kg,
$I_1 = I_2 = I_3 = 0.04$ kg m^2,
$m_4 = m_5 = m_6 = 1$ kg,
$I_4 = I_5 = I_6 = 0.12$ kg m^2,
$m_7 = 8$ kg, $I_7 = 0.0817$ kg m^2,
PID: $K_p = 100, K_i = 20, K_d = 40$,
Initial estimated inertia matrix: **IN** = [1 0.2 0.3] kg m^2,
Learning parameters of IL: $\phi = 0.75, \psi = 0.35, \Gamma = 0.01$,
Desired track error goal: TE = 0.001 m,
Disturbances to the three actuated joints: $\tau_{h(1,2,3)} = 15 \sin t$ Nm.

In the first step, three separate classic PID controllers were designed and applied to the three actuated joints to demonstrate the basic and stable response of the manipulator in performing trajectory tracking tasks. The PID gains were heuristically tuned after a number of trial runs, taking into account the initial ideal situation, i.e. in the absence of disturbances. The results obtained from the PID controller are presented in Figure 10.

Note also that the black arrows in Figure 10 indicate the reciprocating motion of the end-effector from its initial position, in an up-and-down manner, towards the extreme right of the prescribed trajectory, while the green arrows show the returning trajectory to the original position. Subsequently, three harmonic disturbances were later introduced into the
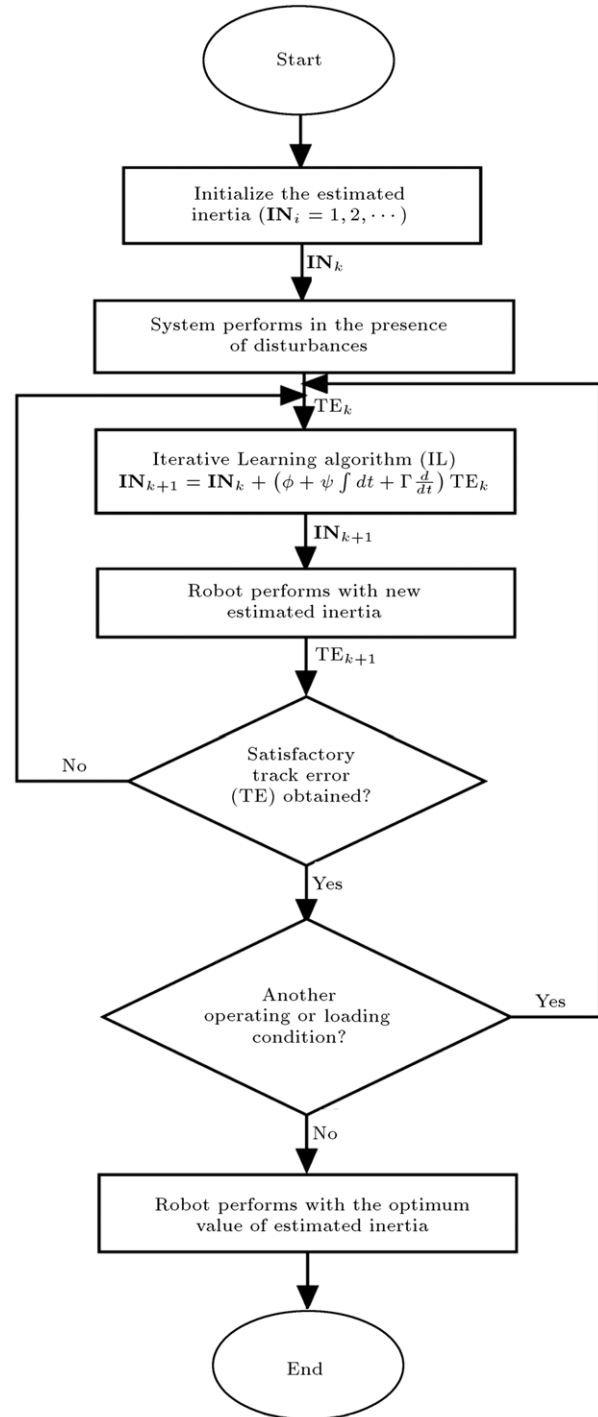

Figure 7: Flowchart of the IL algorithm with stopping criterion.

system. The result obtained from PID controller is presented in Figure 11. From the figure, it is obvious that the system was not able to follow the desired trajectory in the presence of the introduced harmonic disturbances, and the trajectory generated by the PID controller was very much distorted, as demonstrated by the wavy-like profile throughout the tracking period, resulting in large fluctuating error.

Next, the AFC loop (with constant **IN** values obtained from previous research by the same authors [26,27]) was added in series with the PID controller to improve the
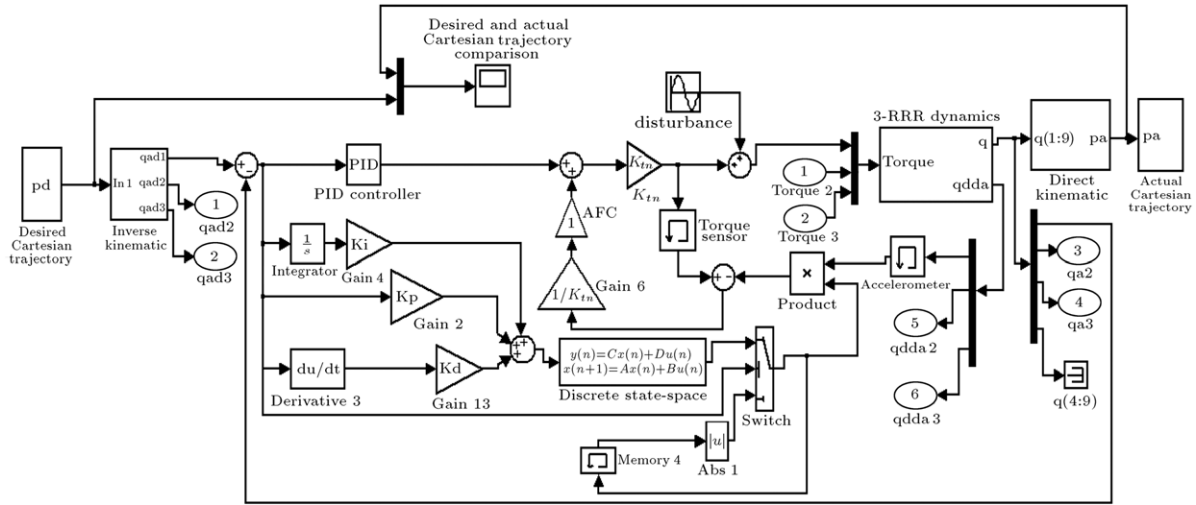
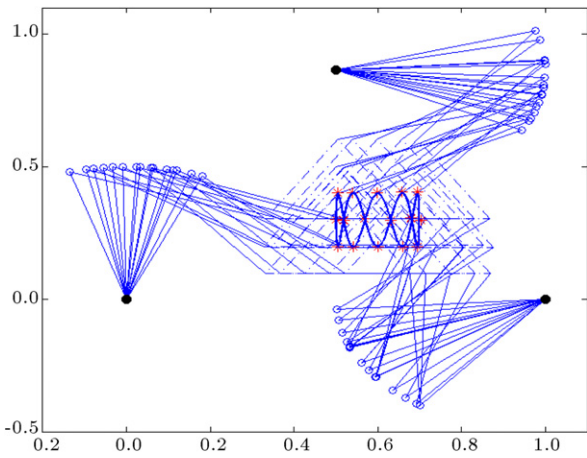Figure 8: MATLAB/simulink diagram of the proposed system.

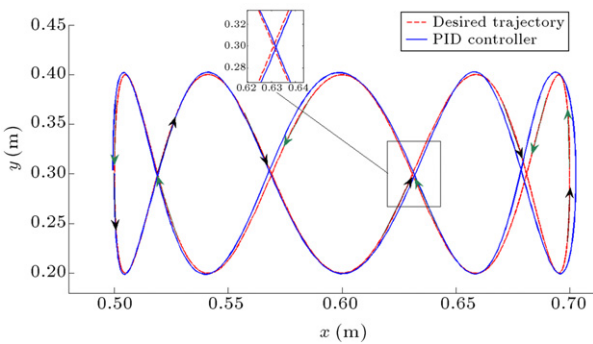Figure 9: Snapshots of the manipulator motion.

Figure 11: Trajectory tracking of the end-effector using PID controller in the presence of harmonic disturbances.

Figure 10: Trajectory tracking of the manipulator using PID controller in the absence of disturbances.

Figure 12: Trajectory tracking of the system using AFC in the presence of the harmonic disturbances.

overall system dynamic performance. The result in Figure 12 clearly illustrates the capability of the scheme in rejecting disturbances, compared to the pure PID controller. The end point of the manipulator is shown to follow the prescribed trajectory, without any fluctuation, but tracking accuracy remains relatively high. This is mainly due to the constant **IN** values assigned for the AFC component to each link, which obviously could not accommodate precisely the given operating
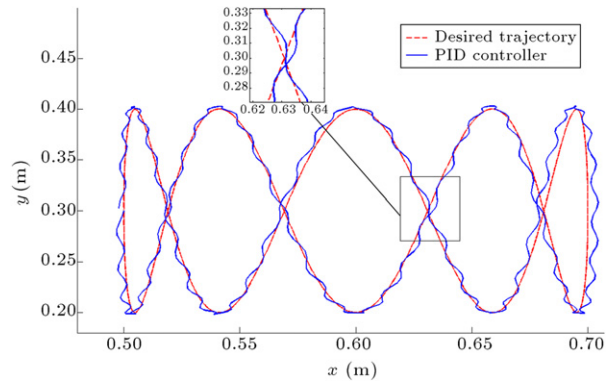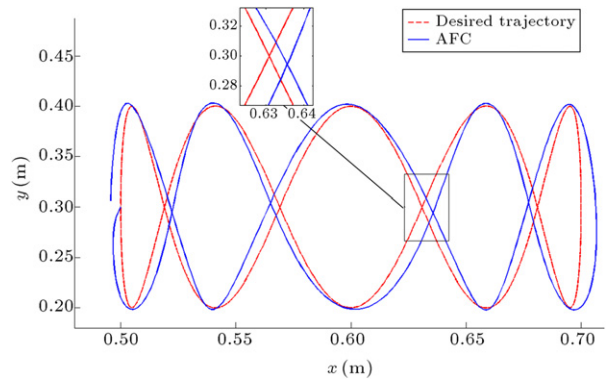
conditions and the complexity of the linkage mechanisms with its kinematics and dynamic elements.

To improve the capability of the system to reject disturbances more efficiently and systematically, an IL algorithm was introduced to the AFC loop to compute the inertia matrices adaptively, while the system is following the desired trajectory. Thus, the simulation was executed based on the desired trajectory track error, i.e. $0 \leq \text{TE} \leq 0.001$ m as the stopping criterion. Once the TE dips below '0.001 m', the IL algorithm stops
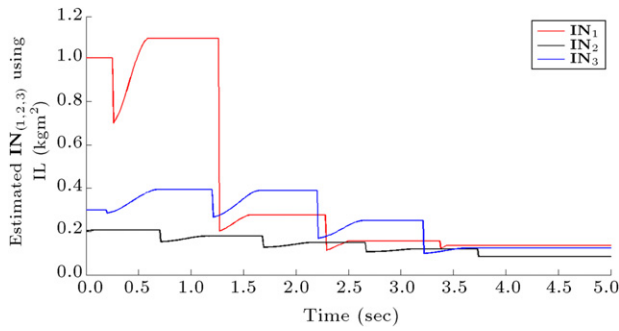
Figure 13:   Estimated inertial parameters of the three actuated joints using IL algorithm.
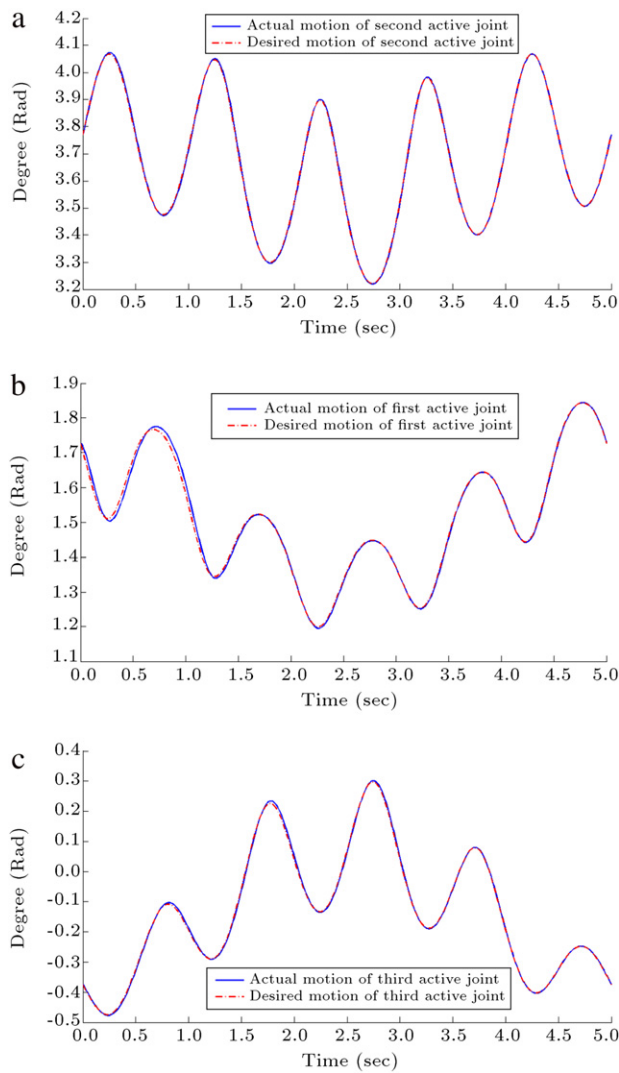

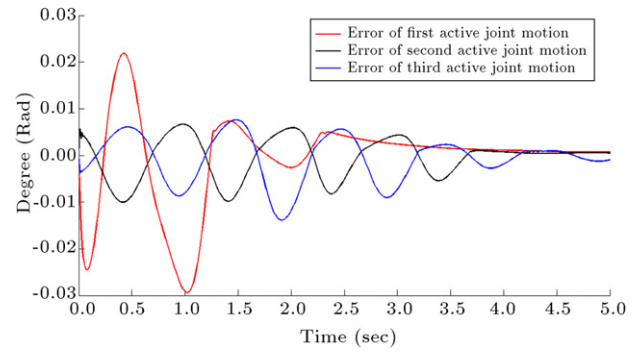Figure 14:   Three actuated joints' motion using AFCAIL.
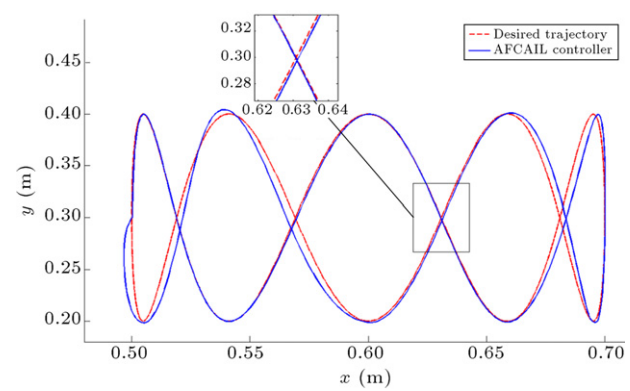

Figure 15:   Trajectory tracking error of the three actuated joints.

Figure 16:   Trajectory tracking of the system using AFCAIL in the presence of the harmonic disturbances.

iterating; thereby implying that the learning process has actually reached a desirable state. Hence, it can be said that the estimated inertial parameter, **IN** (as the output of the IL algorithm), is deemed to be appropriate for the given operating and loading conditions. Later, if another disturbance is applied to the system, the IL algorithm will then compute a new set of **IN** matrices that are required for the AFC loop to reject the new disturbance. The computed inertia matrices obtained by using the IL algorithm over time are shown in Figure 13. Note that the initial values of the **IN** parameters were selected randomly to prove the ability of the proposed algorithm to converge to 'optimized' conditions, regardless of initial values.

The desired and actual trajectory tracking of three actuated joints are presented in Figure 14a–c. The effectiveness of the IL process is clearly demonstrated in the figures, where the initial track error is observed to rapidly converge to near zero datum in all three joints, as time increases.

Also, the tracking errors of the three actuated joints are shown in Figure 15. From the figure, it is clear that the error of each actuated joint (in radian) is gradually reduced after some iterations to meet the error goal ($0 \leq \text{TE} \leq 0.001$ m).

The fast convergence of the track errors to the desired values, thereby, signifies the excellent learning capability of the IL algorithm. The Cartesian trajectory tracking of the manipulator subjected to three harmonic disturbances, using AFCAIL, is shown in Figure 16. From the figure, it is evident that the system trajectory was distorted at the beginning stage of the tracking, due to the randomized initial values of the **IN**, which are obviously not 'optimized' values. However, as time increases, system performance gradually improves, as can be seen from the superior trajectory track performance at a later stage, when learning is said to be completed. The rate of convergence in IL is influenced by the IL parameters, ϕ, ψ, and Γ in which a good choice of these learning parameters is essential to ensure excellent control performance.

Figure 17 gives a better insight into the effect of using the AFC scheme, when the 'optimized' values of estimated inertias (**IN**) obtained from the IL algorithm, were applied directly to the AFC loop from the starting point of the trajectory.
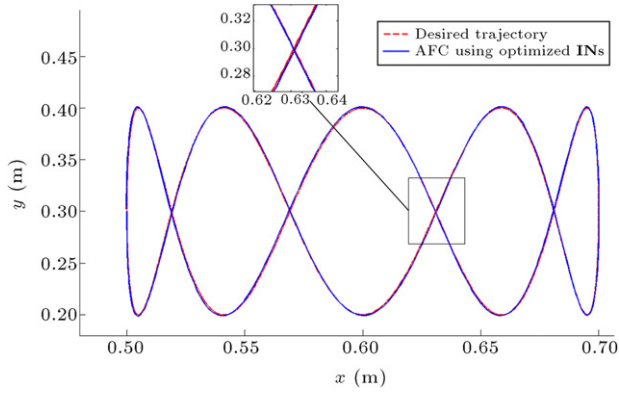
Figure 17: Trajectory tracking of the end-effector using AFC scheme with optimised **IN**s.
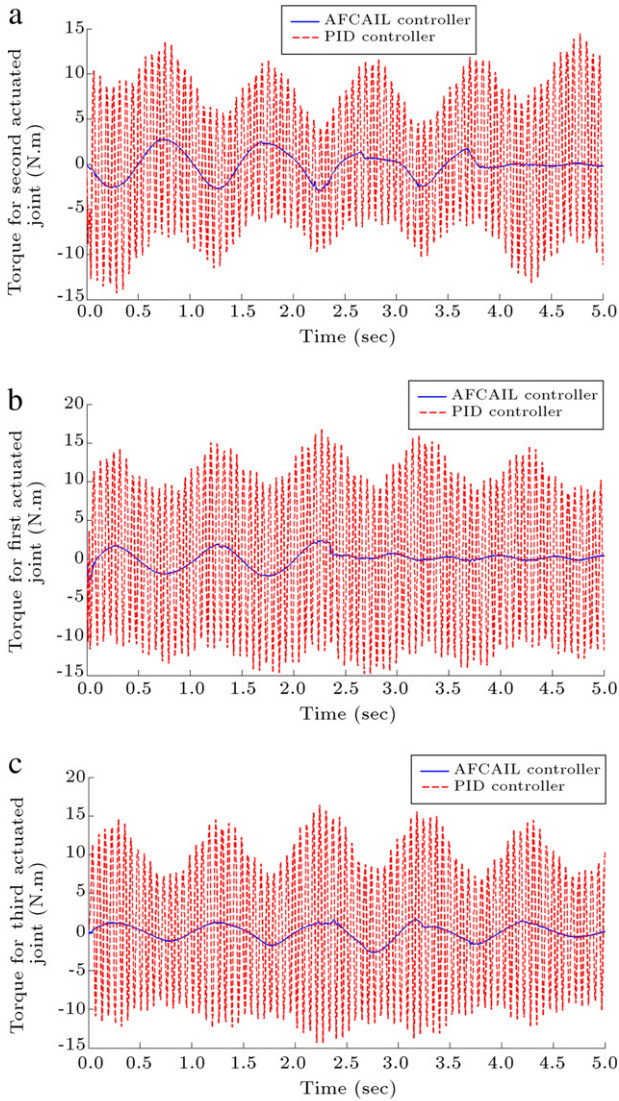

Figure 18: Required torques for the three actuated joints.

In Figure 18a–c, the required torques for each actuated joint by the two controllers, i.e. pure PID and AFCAIL, are illustrated and compared in the presence of disturbances. From the figures, it can be concluded that the torques applied to each actuator by the traditional PID scheme to reject disturbances are much higher in magnitude than in the AFCAIL scheme, and may cause saturation in the actuators. Therefore, it is obvious that AFCAIL outperforms the PID controller under the same conditions, without having to increase the maximum torques of the actuated joints.

5. Conclusion

The case study shows that the proposed feedback control schemes are effective in controlling the highly non-linear 3-RRR planar parallel manipulator performing a pulsating trajectory, in the presence of introduced harmonic disturbances. The AFCAIL scheme is shown to be the most superior, since it produces an almost error-free performance when the **IN** parameters are deemed appropriately computed via the PID-type algorithm, and the disturbances are completely rejected. It is also shown that the IL algorithm with the stopping criterion managed to adapt to different applied disturbance conditions to produce another set of 'optimized' inertial values based on similar initial conditions. Future work could include investigation into sensitivity analysis, based on other different operating and loading conditions, and the validation of the work by experimental means.

Appendix

Inverse kinematic solution

A general solution of the inverse kinematic for leg i is expressed as follows:

$$\theta_i = \alpha_i \pm \psi_i, \quad i = 1, 2, 3, \tag{A.1}$$

$$\alpha_i = a \tan 2(x_{2i}, y_{2i}). \tag{A.2}$$

ψ_i can be obtained from the following equation:

$$\psi_i = \cos^{-1}\left[\frac{l_1^2 - l_2^2 + x_{2i}^2 + y_{2i}^2}{2l_1\sqrt{x_{2i}^2 + y_{2i}^2}}\right], \quad 0 \le \psi_i \le \pi. \tag{A.3}$$

Coordinates x_{2i} and y_{2i} are defined as:

$$x_{2i} = x - l_3 \cos \phi_i - x_{oi}, \tag{A.4}$$

$$y_{2i} = y - l_3 \sin \phi_i - y_{oi}, \tag{A.5}$$

where angles $\{\phi_i\}_1^3$ are given by:

$$\phi_1 = \phi + \frac{\pi}{6}, \tag{A.6}$$

$$\phi_2 = \phi + \frac{5\pi}{6}, \tag{A.7}$$

$$\phi_3 = \phi - \frac{\pi}{2}. \tag{A.8}$$

The Cartesian positions of the centers of motors are considered as follows:

$$x_{oi} = \left\{0, 1, \frac{1}{2}\right\}, \tag{A.9}$$

$$y_{oi} = \left\{0, 1, \frac{\sqrt{3}}{2}\right\}. \tag{A.10}$$

A.1. Direct kinematic solution

$$x_{23} = x_{11} + l_2 \cos(\alpha_1 + \psi) + \sqrt{3}l_3 \cos(\alpha_1 + \alpha_2 + \theta), \quad (A.11)$$

$$y_{23} = y_{11} + l_2 \sin(\alpha_1 + \psi) + \sqrt{3}l_3 \sin(\alpha_1 + \alpha_2 + \theta), \quad (A.12)$$

where:

$$\alpha_1 = a\tan 2\left[\frac{y_{12} - y_{11}}{x_{12} - x_{11}}\right], \qquad \alpha_2 = \frac{\pi}{3}, \quad (A.13)$$

$$\theta_{1,2} = 2\tan^{-1}\left[\frac{b \pm \sqrt{b^2 - ac}}{a}\right]. \quad (A.14)$$

a, b and c are given as follows:

$$a = \frac{-d^2 - 3l_3^2}{2\sqrt{3}l_2l_3} - \frac{d}{l_2} + \left(1 + \frac{d}{\sqrt{3}l_3}\right)\cos\psi, \quad (A.15)$$

$$b = \sin\psi, \quad (A.16)$$

$$c = \frac{-d^2 - 3l_3^2}{2\sqrt{3}l_2l_3} + \frac{d}{l_2} + \left(\frac{d}{\sqrt{3}l_3} - 1\right)\cos\psi, \quad (A.17)$$

and:

$$d = \sqrt{(x_{12} - x_{11})^2 + (y_{12} - y_{11})^2}. \quad (A.18)$$

The coupler curve intersects the circle defined by the rotation of link FC around point F. Therefore, the nonlinear equation to be solved is given by:

$$(x_{23} - x_{13})^2 + (y_{23} - y_{13})^2 = l_2^2. \quad (A.19)$$

Eq. (A.19) can be solved for angle ψ using a standard numerical procedure.

References

[1] Ghorbel, F., Chetelat, O. and Longchamp, R. "On the modelling and control of closed-chain mechanisms", *Proc. IMACS/IEEE Int. Conf. Comput. Eng. Syst. Appl.*, pp. 714–720 (1998).

[2] Amirat, Y., Francois, C., Fried, G., Pontnau, J. and Dafaoiu, M. "Design and control of a new six DOF parallel robot: application to equestrian fait simulation", *Mechatron.*, 6(2), pp. 227–239 (1996).

[3] Stan, S.D., Gogu, G., Manic, M., Balan, R. and Rad, C. "Fuzzy control of a three degree of freedom parallel robot", *Technol. Dev. Educ. Autom.*, pp. 437–442 (2010).

[4] Wu, J. and Wang, L. "Motion control of the 2-DOF parallel manipulator of a hybrid machine tool", *Robot.*, 28, pp. 861–868 (2009).

[5] Kim, D., Kang, J. and Lee, K. "Robust tracking control design for a 6 DOF parallel manipulator", *J. Robot. Syst.*, 17(10), pp. 527–547 (2000).

[6] Tao, G., Zhu, X., Yao, B. and Cao, J. "Adaptive robust posture control of a pneumatic muscles driven parallel manipulator with redundancy", *Proc. Am. Control. Conf.*, pp. 3408–3413 (2007).

[7] Kim, H., Cho, Y. and Lee, K. "Robust nonlinear task space control for 6 DOF parallel manipulator", *Autom.*, 41(9), pp. 1591–1600 (2005).

[8] Yangjun, P. and Xuanyin, W. "Trajectory tracking control of a 6-DOF hydraulic parallel robot manipulator with uncertain load disturbances", *Control. Eng. Pract.*, 19, pp. 185–193 (2011).

[9] Hewit, J.R. and Burdess, J.S. "Fast dynamic decoupled control for robotics using active force control", *Mech. Mach. Theory*, 16(5), pp. 535–542 (1981).

[10] Priyandoko, G., Mailah, M. and Jamaluddin, H. "Vehicle active suspension system using skyhook adaptive neuro active force control", *Mech. Syst. Signal. Process.*, 23, pp. 855–868 (2009).

[11] Varatharajoo, R., Wooi, C.T. and Mailah, M. "Two degree-of-freedom spacecraft attitude controller", *Adv. Space Res.*, 47, pp. 685–689 (2011).

[12] Noshadi, A., Mailah, M. and Zolfagharian, A. "Intelligent active force control of a 3-RRR parallel manipulator incorporating fuzzy resolved acceleration control", *Appl. Math. Modell.*, (2011), doi:10.1016/j.apm.2011.08.033.

[13] Han, J. "From PID to active disturbance rejection control", *IEEE Trans. Ind. Elect.*, 56(3), pp. 900–906 (2009).

[14] Gorman, J. and Dagalakis, N.G. "Modelling and disturbance rejection control of a nanopositioner with application to beam steering", *Proc. ASME Int. Mech. Eng. Congress* (2003).

[15] Uchiyama, M. "Formation of high-speed motion pattern of a mechanical arm by trial", *Trans. Soc. Implement. Control. Eng.*, pp. 706–712 (1978).

[16] Arimoto, S., Kawamura, S. and Miyazaki, F. "Bettering operation of robots by learning", *J. Robot. Syst.*, pp. 123–140 (1984).

[17] Liu, M.J., Li, C.X. and Li, C.N. "Dynamics analysis of the Gough–Stewart platform manipulator", *IEEE Trans. Robot. Autom.*, 16(1), pp. 94–98 (2000).

[18] Gallardo, J., Rico, J.M. and Frisoli, A. "Dynamics of parallel manipulators by means of screw theory", *Mech. Mach. Theory*, 38(11), pp. 1113–1131 (2003).

[19] Tsai, L.W. "Solving the inverse dynamics of a Stewart–Gough manipulator by the principle of virtual work", *ASME. J. Mech. Des.*, 122(1), pp. 3–9 (2000).

[20] Dasgupta, B. and Choudhury, P. "A general strategy based on the Newton–Euler approach for the dynamic formulation of parallel manipulators", *Mech. Mach. Theory*, 34(6), pp. 801–824 (1999).

[21] Miller, K. "Optimal design and modelling of spatial parallel manipulators", *Int. J. Robot. Res.*, 23(2), pp. 127–140 (2004).

[22] Ma, O. and Angeles, J. "Direct kinematics and dynamics of a planar 3-DOF parallel manipulator", *Proc. ASME Conf. Des. Autom.*, 3, pp. 313–320 (1989).

[23] Gosselin, C. and Angeles, J. "Kinematics of parallel manipulators", Ph.D. Thesis, McGill University Montreal, Quebec, Canada (1989).

[24] Bein, Z. and Xu, J.X., *Iterative Learning Control: Analysis, Design, Integration and Applications*, Kluwer Academic Publishers, Norwell (1998).

[25] Owens, D.H. and Hatonen, J. "Iterative learning control—an optimization paradigm", *Annu. Rev. Cont.*, 29, pp. 57–70 (2005).

[26] Noshadi, A., Mailah, M. and Zolfagharian, A. "Active force control of 3-RRR planar parallel manipulator", *Proc. IEEE Int. Conf. Mech. Electr. Technol.*, pp. 77–81 (2010).

[27] Noshadi, A., Zolfagharian, A. and Mailah, M. "Performance analysis of the computed torque based active force control for a planar parallel manipulator", *Appl. Mech. Mater.*, 110–116, pp. 4932–4940 (2012), doi:10.4028/www.scientific.net/AMM.110-116.493.

Amin Noshadi is an M.S. degree engineering student in the Department of System Dynamics and Control, in the Faculty of Mechanical Engineering at Universiti Teknologi Malaysia. He received his B.S. degree in Mechanical Engineering from Ferdowsi University of Mashhad, Iran, in 2007. His current research interests include robotics (specifically in the design and control of parallel manipulators), active force control, iterative learning control, optimization and artificial intelligence (fuzzy techniques and neural network based systems).

Musa Mailah received his B.S. degree in Mechanical Engineering from UTM in 1988 and, later, his M.S. and Ph.D. degrees in Robotic Control and Mechatronics from the University of Dundee, UK, in 1992 and 1998, respectively. Currently, he is Professor in the Department of System Dynamics and Control in the Faculty of Mechanical Engineering, at Universiti Teknologi Malaysia. His current research interests include robotics, mechatronics, and intelligent systems and active force control of dynamical systems.