Adaptive Collaborative Position Control of a Tendon-Driven Robotic Finger

Omer Saleem*, Faisal Abbas*, Muhammad Usman Khan*, Muhammad Anas Imtiaz*, Salman Khalid**

** Electrical Engineering Department, National University of Computer and Emerging Sciences, Lahore, Pakistan (e-mail: omer.saleem, abbas.faisal, usman.muhammad@nu.edu.pk, anas_imtiaz@yahoo.com)*
*** BiSMiL Lab, Information Technology University, Lahore, Pakistan (e-mail: m.salmankhalid29@gmail.com)*

Abstract: This paper presents a collaborative control scheme to optimize the trajectory-tracking performance of the motor that is installed at the metacarpophalangeal (MCP) joint of a robotic finger. A dynamically compensated adaptive-proportional-derivative (APD) tracking controller is used to nullify the tracking errors during free-space motion. An APD compliance controller is used to alleviate disturbance caused by the application of contact-force during the grasping operation. It achieves this objective by altering the reference trajectory, based on the magnitude of applied force. The PD gains of both controllers are adaptively tuned in order to quickly respond to the changes in the system dynamics. Two different intelligent self-tuning mechanisms are used and comparatively analyzed to adaptively adjust the PD gains of these controllers; namely, fuzzy inference system (FIS) and iterative learning algorithm (ILA). An adaptive-neuro-fuzzy-inference-system (ANFIS) is used as an inverse model to transform the reference trajectory into joint-angle dynamics of the finger. It also acts as a feed-forward controller and supervises the trajectory tracking. The feed-forward and tracking controller outputs are beneficially combined via a linearized feedback control law to deliver optimal motor torque commands. The results of real-time experiments are presented to validate the robustness of the proposed controller.

Keywords: Robotic finger, adaptive tracking controller, compliance controller, ANFIS, fuzzy inference system, iterative learning algorithm.

1. INTRODUCTION

The five-fingered robotic hands (FFRH) are capable of imitating the human hand movements. Control of FFRH requires the robotic finger joints to precisely track the input trajectory patterns. The FFRH is constructed by keeping in view the anatomy of a human hand, as shown in Fig. 1. Various designs of anthropomorphic robotic arms are presented in (Carbone and Ceccarelli, 2008; Chen et al., 2012; Xu et al., 2012). A robust optimal tracking control system can improve the agility and flexibility of these end-effectors and allows for dexterous grasping and manipulation of objects, especially in tele-operation applications. Hence, in this paper, an adaptive collaborative position control mechanism is presented to improve the trajectory tracking response of a robotic finger, even in the presence of time-varying reference inputs or bounded disturbances.

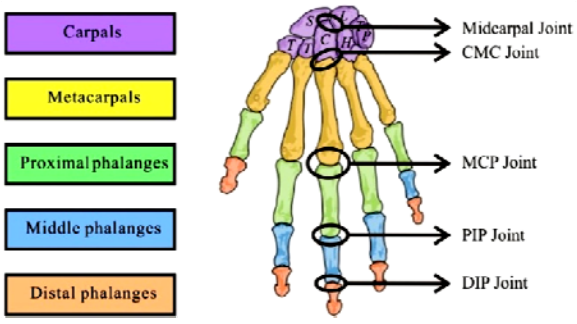

Fig. 1. Anatomy of human hand (Chen and Naidu, 2013).

A lot of work has been done in the past to develop novel and innovative tracking control schemes for robotic hands (Chen and Naidu, 2011; Niehues et al., 2014). Hybrid tracking control strategies offer the best attributes of the different controllers that are being utilized by them (Chen et al., 2014). Therefore, the proposed control scheme combines the control outputs of a classical feedback controller and an intelligent feed-forward controller to achieve an optimal tracking control performance. The feed-forward controller ensures efficient and accurate tracking of reference trajectory by augmenting the tracking controller with the dynamic changes in the reference trajectory. An intelligent learning and inference system is used for this purpose. Primarily, it acts as an inverse kinematic model to translate the trajectory (given in Cartesian space) into joint-space. Secondarily, it acts a feed-forward controller to dynamically actuate the finger joints according to the reference trajectory (Rezaeeian et al., 2008). The studies show that the ANFIS is a superior intelligent control mechanism than the fuzzy logic or the artificial neural controllers. It adaptively combines the inference and learning based features of both controllers, respectively (Carbureanu, 2014; Potluri et al., 2010; Saini and Rani, 2012; Zaki et al., 2010). In this research, a well-postulated ANFIS is used as an inverse model to generate the necessary motor torque commands based on the dynamic changes in the desired trajectory (Duka, 2015).

The trajectory tracking controller serves to minimize the tracking deviations, occurring between the reference trajectory and the actual response traced by the finger joint, caused by friction, modeling errors, or exogenous disturbances. Moreover, it improves the convergence-rate of the system in case of large transient. The PID controllers are widely used in process control industry and robotics

(Bequette, 2003; Bhatti et al., 2015; Shauri et al., 2014). It is a model-free control mechanism that depends on the weighted sum of the error, its rate-of-change and recent sum. However, the PD controllers have better asymptotic convergence capability (Shang et al., 2009). Therefore, in this research, the trajectory tracking control of the robotic finger is implemented via a dynamically compensated PD controller, where in the dynamic changes in the reference trajectory is provided by the feed-forward controller.

The contact-force controllers play an important role in improving the tracking control mechanism of robotic manipulators. These controllers enable the robotic fingers to firmly grasp an object without de-shaping it or disturbing the dynamic performance of the robotic fingers. The conventional force controllers tend to compare the contact-force with a set-point value and generate a control command to alleviate the error. However, this methodology is not practically viable, because it is very cumbersome to pre-define the set-point values of force before every grasping application. The PD compliance controller offers a better alternative solution (Song and Hsu, 2011). In this technique, the instantaneous magnitude of the contact-force being applied is used to alter the reference trajectory directly. The modified trajectory input allows the dynamic tracking controller to apply a reasonable amount of torque to the joint-motors so that the robotic fingers can appropriately penetrate the surface of the object and grasp it firmly. Thus, no set-point of force is needed.

The linear combination of the error-dynamics in PD controllers offers a robust and efficient control strategy; if and only if the weightages of the error-dynamics are tuned properly. These weightages are denoted as the PD gains. The conventional techniques used to tune the controller gains do not always yield the best solution. The meta-heuristic search and optimization algorithms offer a better solution (Vijayakumar and Manigandan, 2016). However, in some practical cases, even the most optimally tuned fixed gain controllers are unable to control the complex dynamical systems. This is majorly because they lack the ability to handle the abrupt variations incurred in the system's states. In order to quickly suppress the affects of bounded exogenous disturbances and parametric uncertainties, the PD gains of the tracking and the compliance controllers must be adaptively adjusted. Several intelligent self-tuning techniques have been discussed in the literature. The FIS is a popular model-free self-tuning technique. The system relies upon a heuristically synthesized fuzzy rule-base to dynamically adjust the PD gains (Baroud et al., 2015; Karray and Feki, 2015). Another effective technique to adaptively tune the PD gains is the usage of ILA (Xu and Huang, 2007). The ILA observes the performance of the system in response to the correctional efforts provided by the algorithm itself in the previous iterations. It then uses this knowledge to improve the effort in the present iteration. The iterative process continues until the desired correctional effort is achieved (Xu et al., 2009). In this research, the tracking and compliance controllers are separately self-tuned via FIS and ILA. Their contributions in optimizing the tracking and compliance control mechanisms are tested turn-by-turn on the index finger of an FFRH prototype via real-time 'hardware-in-the-loop' experiments. The corresponding results are comparatively analyzed in order to choose a superior self-tuning technique for the synthesis of a robust tracking controller.

The rest of the paper is formatted as follows: Section 2 presents the experimental setup of the robotic finger. The dynamics of the robotic finger are discussed in section 3. The control system design is explained in section 4. Section 5 presents the real-time experiments and their results to evaluate the performance of the proposed control scheme. Section 5 concludes the paper.

2. EXPERIMENTAL SETUP

The experimental setup used to test the proposed control mechanism is described in the following sub-sections.

2.1 Electronic hardware setup

The robotic hand contains five fingers. Each finger is actuated with the aid of a mini permanent magnet direct-current (PMDC) geared motor. Optical encoders are used to acquire the information regarding the instantaneous position of the motor shaft, and hence the MCP joint-angles (θ_r) of each finger. Capacitive tactile force sensors are appropriately placed at interior side finger tips to measure the contact-force (f_R) applied by the robotic-fingers upon grasping an object. All the sensor readings are passed through a low pass filter to remove high-frequency noise and outlier data from the acquired readings. The filtered data is then differentiated to attain the rate-of-change of joint-angles ($\dot{\theta}_r$) and the rate-of-change of applied contact-force ($\dot{f}_R$). The sensor readings and their derivatives are then fed to tracking as well as compliance controller, respectively. The motor torque control commands generated by the controller are applied to the joint-motor via a dedicated motor driver circuit.

2.2 Software architecture

A LabVIEW based graphical user interface (GUI), running on a personal computer, is used in this research (Elmerabete et al., 2010). It is used to apply the reference input signal, visualize the corresponding joint-angle response (and other sensor readings), run the control routine, and generate appropriate control signals. In order to conduct the "hardware-in-the-loop" simulations, a device is needed to relay the information regarding the current states of the joint-motor from the sensors to the GUI, as well as to communicate the string of control signals issued by the software to the motor driver circuit. Hence, an 8-bit embedded microcontroller is interfaced with the GUI software to measure the response and accordingly control the position of the robotic finger-joint in real-time. Initially, the reference trajectory is fed to the ANFIS. The ANFIS acts as an inverse model to generate the information regarding the MCP joint-angle, angular velocity, and angular acceleration. The reference angle and angular velocity are fed as to the APD tracking controller for comparison and computation of tracking error.

The measured values of joint-angle and force, acquired by the microcontroller via the encoders and force sensor (respectively), are filtered and then serially transmitted (at

9600 bps) to the LabVIEW application. The filtered sensor data is differentiated and resulting information is compared with their counter-parts from reference input. The resulting angular-error (or tracking-error) and force dynamics are evaluated and fed to the APD based tracking and compliance control routines, respectively. The angular acceleration delivered by the ANFIS and tracking control command delivered by the APD controllers are added together. The collaborative torque control output is transmitted to the microcontroller where it is translated to generate an appropriate pulse-width-modulated (PWM) command. Eventually, this PWM command drives the joint-motor. The entire system is operated at a sampling frequency of 250 Hz.

2.3 Embodiment design

The robotic hand has five fingers with one degree-of-freedom in each finger. The motors are installed at the MCP joint of each finger. A tendon-driven (wire-pulley) mechanism is used, in conjunction with the joint-motor, to bend or stretch each finger. The motors offer sufficient driving torque at low rotational speed, have a compact size, and are light in weight. Each fingertip moves from its initial extension posture (1.55 rad.) to its maximum flexion posture (0.21 rad.) The schematic of the tendon-driven robotic finger-joint is represented in Fig. 2. The relationship between τ_1 (MCP joint-torque) and the corresponding tension (F) in the tendon is given by (1).

$$\tau_1 = rF \tag{1}$$

where, r is the radius of the joint. The tension is proportional to the length of the tendon (l), as given in (2). The stiffness-value (k_t) serves as the constant of proportionality.

$$F = k_t l \tag{2}$$

The resulting position of the tendon (y) is given by (3).

$$y = r\theta_r + l \tag{3}$$

where, θ_r is the measured angular-position of the MCP joint.

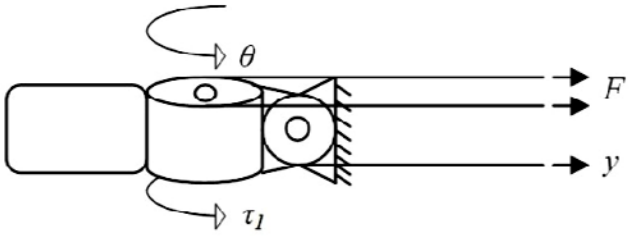

Fig. 2. Schematic of a tendon-driven robotic finger.

One complete finger, equipped with the wire pulley mechanism, is shown in Fig. 3. The entire structure is fabricated with a 5.0 mm think fiber-glass sheet because it is durable, light-weight and cheap. The fully fabricated structure of mechanical hand is shown in Fig. 4.

3. DYNAMICS OF ROBOTIC FINGER

In this section, the motion of the proposed robotic-finger mechanism is analyzed. These dynamics are explained in terms of the mathematical model of the motor, the forward kinematics of the finger-joint, and the inverse kinematic model used for generating a trajectory in the joint-space.

3.1 Actuator Model

The fingers in the robotic hand are being driven by metal-geared PMDC servo motors (Widhiada et al., 2015). The transfer function of the angular-displacement of this motor with respect to excitation voltage applied to it is presented in (4).

Fig. 3. Structure of robotic finger.

Fig. 4. Five-fingered robotic hand.

$$\frac{\theta_r(s)}{V_a(s)} = \frac{nK_a}{s\left(s^2 PL_a + (L_a v + PR_a)s + R_a f_v + K_a K_b\right)} \tag{4}$$

where,

θ_r = Angular displacement of motor shaft, (rad.)

V_a = Armature voltage, (V)

P = Moment of inertia referred to motor shaft (J_m) and load shaft (J_L) = 0.02675 kgm^2

L_a = Armature inductance = 2.7 μH

R_a = Armature resistance = 2.0 Ω

f_v = Viscous friction coefficient referred to motor shaft (f_m) and load shaft (f_L) = 0.42 Nms

K_a = Motor torque constant = 6.28 kg.cm/A

K_b = Speed proportionality constant = 13.2 rpm/V

n = Gear ratio = 16

3.2 Finger kinematics

The study of kinematics deals with the geometrical interpretation of the motion of dynamical system. This interpretation includes the position and orientation (and their higher-order derivatives) of the mechanical links and joints. The schematic of a three link index finger is shown in Fig. 5. The lengths of the link 1, 2, and 3 are given by l_1, l_2, and l_3, respectively. Similarly, the θ_r, θ_p, and θ_d, are the angles of the joints 1 (MCP), 2 (PIP), and 3 (DIP) of the robotic finger, respectively. The fingertip coordinates (X, Y) and the orientation (ϑ) of the index finger is derived via the Devenit-Hartenberg (DH) method (Chen et al., 2011). The expressions are given in (5), (6), and (7).

$$X = l_1 \cos(\theta_r) + l_2 \cos(\theta_r+\theta_p) + l_3 \cos(\theta_r+\theta_p+\theta_d) \tag{5}$$

$$Y = l_1 \sin(\theta_r) + l_2 \sin(\theta_r+\theta_p) + l_3 \sin(\theta_r+\theta_p+\theta_d) \tag{6}$$

$$\vartheta = \theta_r+\theta_p+\theta_d \tag{7}$$

Since a wire-pulley mechanism is used in this research, therefore, the joint angles, θ_p and θ_d, are dependent on and can be derived from the MCP joint-angle (θ_r).

3.3 Inverse kinematics

The inverse kinematics provides the required joint-angles needed to move the finger from one position (or orientation) to another in the three-dimensional space.

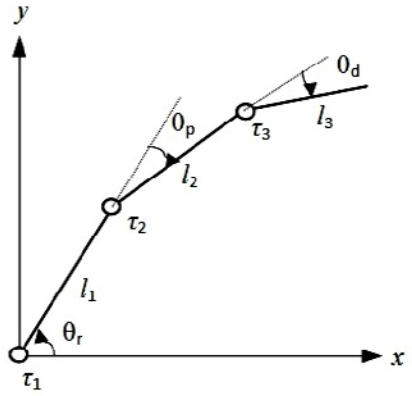

Fig. 5. Three-link finger.

This is very helpful in planning and generating a trajectory pattern for the manipulator. Generally, the reference trajectories are provided in the Cartesian space in the form of cubic polynomial. For a multi-joint manipulator, an accurately designed inverse kinematic model is required to translate the starting and destination coordinates of the end-effector into equivalent joint-angles. The analytical solution of this technique is difficult to compute as the number of links in the robotic increases. However, a well-trained ANFIS can be used to accurately formulate a strategy to convert Cartesian-space (inputs) to joint-space (outputs). The fuzzy logic systems use logical reasoning based on a set of heuristically fabricated set of "if-then-else" rules, however, they lack the ability to dynamically learn. The artificial neural network can learn and develop a model from the training data presented to them, but they lack a knowledge base. The ANFIS combines the best features of the artificial neural network with Sugeno-fuzzy logic structure (Asgari and Ardestani, 2013). The Cartesian coordinates and corresponding joint-angles are recorded to train the ANFIS. The ANFIS optimizes the parameters of the fuzzy inference system using neural network based learning method to create a model that can accurately map the input data to the output data. The hybrid learning algorithm is used to train the ANFIS structure (Angulo and Godo, 2007). It consists of the least-squares method (LSM) and the back-propagation (gradient-descent) algorithm (BPA). The LSM identifies consequent parameters in the forward path. Keeping the consequent parameters fixed, the BPA optimizes the premise parameters iteratively in the backward path, until the desired response is achieved. The governing rules of the said model are as follows:

Rule 1: If g is A_1 and h is B_1 then $z_1 = p_1x + q_1y + r_1$

Rule 2: If g is A_2 and h is B_2 then $z_2 = p_2x + q_2y + r_2$

where, g and h are the input variables, A_j and B_j are the linguistic variables of the fuzzy rule-base, v is the output function, p_j, q_j, r_j are the consequent parameters, and j is the number of rule being considered (j = 1 or 2). The ANFIS uses a five-layered structure, as shown in Fig. 6.

Layer 1: The first layer fuzzifies the input data (g and h) using the triangular membership functions (MF), given by (8). The layer evaluates the degree (μ) of input variables in the fuzzy set. The output of this layer is $\mu_{Aj}(g)$ or $\mu_{Bj}(h)$.

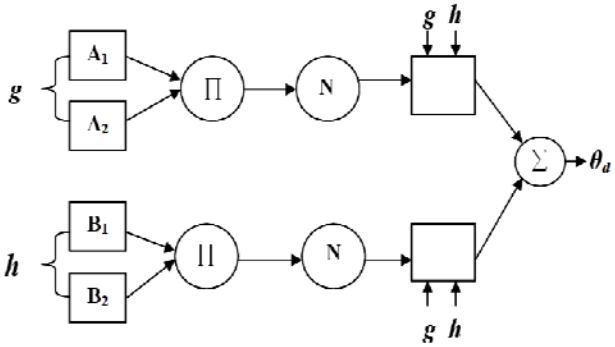

Fig. 6. The ANFIS structure.

$$\mu_j(s) = \begin{cases} 1+\dfrac{s-d_j}{b_j^-}, & -b_j^- \le s-c_j \le 0 \\ 1-\dfrac{s-d_j}{b_j^+}, & 0 \le s-c_j \le b_j^+ \\ 0, \ otherwise \end{cases} \tag{8}$$

where, $\mu(s)$ is the j^{th} triangular MF having b_j^-, d_j and b_j^+ as the left-half width, center and right-half width respectively.

Layer 2: The second layer is the product layer. It commences fuzzy inference in order to produce the dynamic states of the joint using the fuzzy rule base. Each node in this layer corresponds to a fuzzy antecedent rule (the "if" part). The product T-norm aggregation operator is used with each fuzzy

neuron. The output of this layer is expressed via a weightage (w), also referred to as the "firing strength", according to (9).

$$w_j = \mu_{Aj}(g) \times \mu_{Bj}(h) \tag{9}$$

Layer 3: The third layer acts as a normalization layer. This layer takes the ratio of the firing strength of the j^{th} fuzzy rule to sum of the firing strengths of all the rules. The normalized weightage is given by the (10).

$$\widehat{w}_j = \frac{\sum w_j}{\sum_j w_j} \tag{10}$$

Layer 4: The fourth layer performs de-fuzzification of the consequent rules. It takes product of the normalized weightages from layer 3 and the output function (v). This linear combination transforms the individual fuzzy classifications into a crisp output, as given by (11).

$$s_j = \widehat{w}_j\,(p_j g + q_j h + r_j) \tag{11}$$

Layer 5: The fifth layer performs the summation of all the incoming signals from layer 4. The aggregated output is given by (12).

$$O_j = \sum_j s_j = \frac{\sum_j w_j\,(p_j g + q_j h + r_j)}{\sum_j w_j} \tag{12}$$

The ANFIS acts as an inverse model to plan and actuate the joint-motor according to the input trajectory pattern (Bakırcıoğlu, 2016). Additionally it serves as a supervisory controller to feed forward the dynamic changes in the reference trajectory (angular acceleration).

4. CONTROL SYSTEM DESIGN

The proposed control scheme mainly consists of the tracking controller that suppresses the tracking error in position. The tracking controller is supported by the compliance controller that offers active force control (and alleviates the corresponding disturbances) during grasping action. Finally, the intelligent adaptation mechanism serves to self-tune these closed-loop controllers. The outputs of these controllers are beneficially combined with the dynamic contributions of the feed-forward controller using a feedback linearization technique (Chen and Naidu, 2014).

4.1. Feedback linearization

The Lagrangian approach is used to derive the dynamic equations of motion, given in (13), for the robotic finger (Abdallah et al., 2013).

$$M\ddot{\theta}_r + \rho = \tau \tag{13}$$

where,

$\boldsymbol{M}$ is the joint-inertia matrix

$\boldsymbol{\rho}$ is vector representing the sum of Coriolus, centripetal, gravitational the, and frictional forces

$\ddot{\boldsymbol{\theta}}_{\boldsymbol{r}}$ is the angular-acceleration vector of the MCP joints

$\boldsymbol{\tau}$ is the torque vector of the revolute joints

The nonlinear dynamics represented by (13) are transformed into linear state-space system. The angular position and the angular velocity are chosen as the state-variables of this tracking control system, as given by (14).

$$\boldsymbol{x} = \begin{bmatrix} \theta_r \\ \dot{\theta}_r \end{bmatrix} \tag{14}$$

Where, θ_r and $\dot{\theta}_r$ represent the angular positions and angular velocities of the MCP joints, respectively. The dynamic equation in (13) is re-arranged to give (15).

$$\ddot{\theta}_r = -M^{-1}(\rho - \tau) \tag{15}$$

The expressions given in (14) and (15) are used to derive the linear state-space system given by (16).

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & \mathrm{I} \\ 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ \mathrm{I} \end{bmatrix} \boldsymbol{u} \tag{16}$$

Such that, the control input '$\boldsymbol{u}$' is given by (17).

$$u = -M^{-1}(\rho - \tau) \tag{17}$$

At a given instant, corresponding to the reference trajectory, if θ_d, $\dot{\theta}_d$, and $\ddot{\theta}_d$ are the desired values of angular positions, angular velocities and angular accelerations of the MCP joints, respectively; then the resulting tracking errors in positions, velocities, and accelerations are computed by (18), (19), and (20), respectively.

$$e = \theta_d - \theta_r \tag{18}$$

$$\dot{e} = \dot{\theta}_d - \dot{\theta}_r \tag{19}$$

$$\ddot{e} = \ddot{\theta}_d - \ddot{\theta}_r \tag{20}$$

Substituting the expression given in (15) into (20), yields the expression given by (21).

$$\ddot{e} = \ddot{\theta}_d + M^{-1}(\rho - \tau) \tag{21}$$

Using the linear state-variable system expressed in (16), the updated form of the control input can be expressed according to the expressions in (22).

$$u = \ddot{\theta}_d + M^{-1}(\rho - \tau) \tag{22}$$

The expression in (22) is referred to as the linearized feedback control law. The function for the torque applied to the motor is given by (23).

$$\tau = M(\ddot{\theta}_d - u) + \rho \tag{23}$$

As already discussed in the previous section, the value of desired angular acceleration ($\ddot{\theta}_d$) is fed-forward by the ANFIS inverse model. The control effort (u) is provided by the self-tuning PD based tracking controller. The building-blocks are described in the following sub-sections.

4.2. Tracking controller

A classical feedback controller is used to minimize the tracking error in the angular position (θ_r) and angular velocity ($\dot{\theta}_r$) of the MCP joint-motor in the robotic finger. Apart from minimizing the tracking error in the angular-position, the controller also serves to improve the convergence-rate and dampen the oscillations. The PID controller is a popular controller in the process-control industry. It is a conventional and linear control mechanism that does not require the mathematical model of the system for its computation.

Despite having a simple structure, it offers effective control effort in bringing the system to its steady-state efficiently. In contrast to the PID controllers, the PD controllers provide better global asymptotic stability for a trivial pair of PD gains (Su and Zheng, 2010). The PD tracking control law, used in this research, is given by (24).

$$u = - k_P e - k_D \dot{e} \tag{24}$$

where, k_P and k_D are the proportional and derivative gains of the tracking controller, respectively. Initially, these gains are optimally tuned via trial-and-error method. The values of these gains are fixed at k_P = 68.24 and k_D = 8.65. However, the fixed-gain controllers are unable to handle dynamical systems in the presence bounded random perturbations. If not taken care of, these disturbances may lead to degradation of the tracking response. Hence, the fixed gains in the PD control law are replaced by self-regulated gains. These gains are adjusted online by a dedicated adaptation mechanism. Self-regulation of gains enhances the performance of the controller by enabling it to quickly alleviate the effects of bounded exogenous disturbances.

4.3. Compliance controller

The compliance control mechanism uses the force feedback to ensure a smooth interaction between the surface of the robotic finger and the object (having unknown stiffness coefficient). This phenomenon enhances the dexterity in grasping the objects. Upon contact with an object, the instantaneous value of the contact-force (f_r) experienced by the robotic fingers is measured and filtered to remove noise. The filtered data is fed to a PD force controller. The controller generates a proportional perturbation in the reference trajectory signal, as shown in (25).

$$\Delta z = k_P{}' f_r + k_D{}' \dot{f}_r \tag{25}$$

where, $k_P{}'$ and $k_D{}'$ are the proportional and derivative gains of the compliance controller, respectively. Upon contact, the compliance controller tends to nullify the effect of torque signals applied by the feed-forward controller and the tracking controller by directly modifying the desired trajectory. It makes sure that the fingers have firmly grasped the object while applying a reasonable force on it. This feature maintains the integrity of the trajectory-tracking performance of the fingers (Seraji, 1998).

The PD gains must be optimally adjusted in order to ensure asymptotic stability of the controller. Initially, the values of the compliance PD gains are fixed at $k_P{}'$ = 1.38 and $k_D{}'$ = 0.18, via trial-and-error method. Fixed-gain compliance controllers perform well only if the stiffness coefficient of the object is already known and the gains are tuned accordingly. In practice, the stiffness coefficients of every object cannot be pre-ascertained. Hence, the fixed-gain controllers tend to exhibit a slow response upon contact with a soft object, and become unstable while grabbing a hard object. Therefore, in this research, the gains of the compliance controller are also adjusted dynamically so that the finger can robustly adapt to the hardness or softness of the object while grasping it, without affecting the trajectory tracking phenomenon.

4.4. Adaptation mechanism

As discussed earlier, the adaptive tuning of the PD gains is essential in enhancing the controller's transient response, steady-state response, convergence-rate of tracking-error, and susceptibility against external disturbance. In this research, two intelligent self-tuning mechanisms are used to adaptively modulate the PD gains of the tracking and the compliance controller; namely, the FIS and the ILA. These adaptation mechanisms serve to self-regulate the PD gains in real-time with respect to the variations in respective controlled variables (or states) of the controller being tuned. The improvements rendered by these mechanisms in robustifying the tracking and compliance controllers are individually analyzed via real-time experiments.

A. Fuzzy inference system

The FIS is a model-free technique. It infers its actions according to a set of logical rules that are heuristically synthesized by the control system designer, based on his expertise. In this research, a two input-two output FIS is used for self-tuning the PD controllers, as shown in Fig. 7.

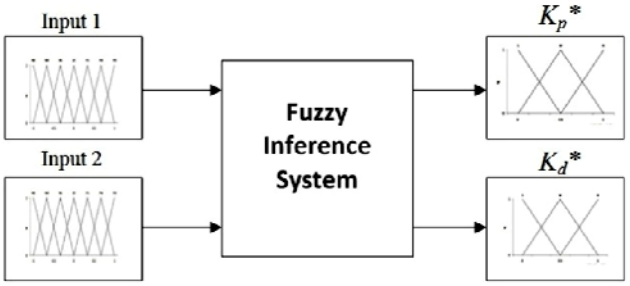

Fig. 7. Structure of Fuzzy Inference system.

A separate FIS system is used for tuning each of the two PD (tracking and compliance) controllers being considered (Xing et al., 2015). The FIS for the tracking PD controller takes e and $\dot{e}$ as the inputs, and delivers the updated values of k_P and k_D as its outputs. The FIS for the compliance PD controller takes f_r and $\dot{f}_r$ as the inputs, and delivers the updated values of $k_P{}'$ and $k_D{}'$ as its output.

The updated values of k_P and k_D are bounded within the prescribed ranges of [40, 100] and [2, 16], respectively. Similarly, the updated values of $k_P{}'$ and $k_D{}'$ are bounded in the interval [0.6, 2.2] and [0.1, 1.2], respectively. The PD gains are normalized in the range [0, 1] by using (26).

$$K_z = \frac{k_z - k_{z,min}}{k_{z,\ max} - k_{z,min}} \tag{26}$$

where, K_z is a general notation used to represent K_p or K_d for either of the two controllers. The PD gains are updated via a set of fuzzy rules of the form: *If input-1 is A_j and input-2 is B_j, then K_p* is C_j and K_d* is D_j,* where, A_j, B_j, C_j, and D_j are the fuzzy sets on the corresponding supporting sets, such that j = 1, 2,.., n. The fuzzy subsets of membership-functions (MF) of the input variables, for both controllers, are linguistically defined as; Negative Big (NB), Negative Small (NS), Zero (Z), Positive Small (PS), and Positive Big (PB). The fuzzy subsets of the output variables (K_p* and K_d*), for both controllers, are defined as; Small (S), Medium (M), Big

(B), Very Big (VB). The output value of the grade, μ_j, is given by (27).

$$\mu_j = \mu_{Aj} \times \mu_{Bj} \tag{27}$$

where, μ_{Aj} is the MF value of the fuzzy set A_j according to the value of e or f_r, and μ_{Bj} is the MF value of the fuzzy set B_j according to the value of $\dot{e}$ or $\dot{f}_r$. Based on the value of μ_j, the values of K_p* and K_d* for each rule are determined from their corresponding membership functions. The rule-base to tune the K_p* and K_d* for tracking controller and compliance controller is given in Table 1 and 2, respectively.

Table 1. Fuzzy rule base of tracking controller gains.

k_P, k_D		$\dot{e}$				
		NB	**NS**	**Z**	**PS**	**PB**
e	**NB**	V,M	B,M	B,S	B,M	V,M
	NS	B,M	M,S	M,S	M,S	B,M
	Z	B,S	M,S	S,S	M,S	B,S
	PS	B,M	M,S	M,S	M,S	B,M
	PB	V,M	B,M	B,S	B,M	V,M

Table 2. Fuzzy rule base of compliance controller gains

k_P', k_D'		$\dot{f}_r$				
		NB	**NS**	**Z**	**PS**	**PB**
f_r	**NB**	B,M	M,M	S,M	M,M	B,M
	NS	B,M	M,S	S,S	M,S	B,M
	Z	M,S	M,S	S,S	M,S	M,S
	PS	B,M	M,S	S,S	M,S	B,M
	PB	B,M	M,M	S,M	M,M	B,M

There are a total of 25 tuning rules for each gain of each controller. The triangular MF of the inputs and outputs, for both controllers, are shown in Fig. 8 and 9, respectively. The center-of-area based de-fuzzification method, in (28), provides the normalized PD coefficients.

$$K_z^* = \frac{\sum_{j=1}^{n} \mu_j K_{z,j}^*}{\sum_{j=1}^{n} \mu_j} \tag{28}$$

where $K_{z,j}$* is the value of K_z* corresponding to the grade μ_j for the j^{th} rule. The actual values of the updated PD gains are calculated via (29).

$$k_z = \left(K_{z,max} - K_{z,min}\right) K_z^* + K_{z,min} \tag{29}$$

B. Iterative learning algorithm

The ILA is also an intelligent model-free technique. Instead of developing an accurate mathematical model to represent the relationship between the PD gains and the variations in system dynamics (or states), the ILA optimally tunes the gains during each sampling interval via repeated learning (Fadil et al., 2013). This repeated sampled learning mechanism provides a new paradigm to precisely tune the PD gains. The algorithm simply memorizes the past values of PD gains and the corresponding profiles of tracking-error (or force) dynamics from the previous iterations, and uses this information to update the PD gains in the present iteration. The algorithm initiates with very little knowledge of the process and progressively improves via a well-postulated learning mechanism.

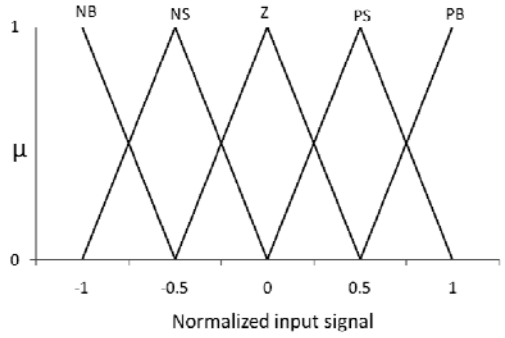

Fig. 8. Input fuzzy membership function.

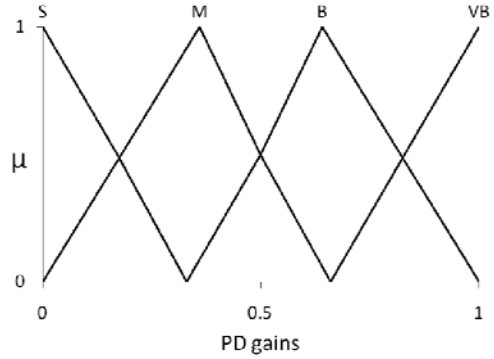

Fig. 9. Output fuzzy membership function.

The expression of PID-type ILA, expressed in (30), is derived from the original derivative-type iterative learning control scheme presented in (Arimoto et al., 1984).

$$u_{i+1} = u_i + \alpha e_i + \beta \int_0^t e_i \, d\tau + \gamma \dot{e}_i \tag{30}$$

where, u_i is the control signal at i^{th} iteration of a given time-sample, e_i is the tracking error at the i^{th} iteration of a given time-sample, and α, β, and γ are the learning gains. Generally the ILA system is demonstrated according to Fig. 10. The objective is to make $e_i \rightarrow 0$ as $i \rightarrow \infty$. In order to achieve asymptotic stability, the leaning gains are tuned using (31).

$$|1 - (\alpha + \beta + \gamma)\, b_1| < 1 \tag{31}$$

where, b_1 is the first Markov parameter. If the state-space representation of the continuous-time system is available in terms of its matrices (A, B, C, and D), then $b_1 = CB \neq 0$. In this research, the integral part in PID-type ILA is excluded because the ILA naturally takes the sum of the control action from previous iterations. Hence, a PD-type ILA based model is used to self-tune the controller(s) and deliver precisely adjusted PD gains within a finite time interval (Han and Lee, 2011). Consequently, the modified ILA used to tune the k_P and k_D of tracking controller is given by (32).

$$\begin{bmatrix} k_p \\ k_d \end{bmatrix}_{i+1} = \begin{bmatrix} k_p \\ k_d \end{bmatrix}_i + \begin{bmatrix} \alpha_p & \gamma_p \\ \alpha_d & \gamma_d \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix}_i \quad (32)$$

The expression in (32) can be interpreted according to (33).

$$\boldsymbol{k}_{z,\,i+1} = \boldsymbol{k}_{z,\,i} + \boldsymbol{\varphi} \boldsymbol{E}_i \quad (33)$$

where, $\boldsymbol{k}_{z,i}$ is the vector containing the values of k_P and k_D at i^{th} iteration of a given time-instant, $\boldsymbol{E}_i$ is the vector of error dynamics at i^{th} iteration of a given time-instant, and $\boldsymbol{\varphi}$ is the matrix of manually-tuned fixed learning gains of ILA for self-tuning of tracking controller.

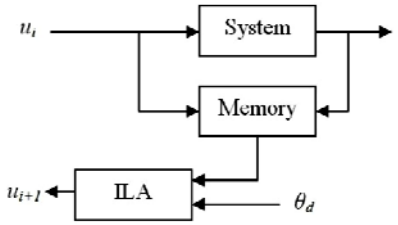

Fig. 10. Structure of iterative learning algorithm.

Similarly, the modified ILA used to tune the $k_P{}'$ and $k_D{}'$ of compliance controller is given by (34).

$$\begin{bmatrix} k'_p \\ k'_d \end{bmatrix}_{i+1} = \begin{bmatrix} k'_p \\ k'_d \end{bmatrix}_i + \begin{bmatrix} \alpha'_p & \gamma'_p \\ \alpha'_d & \gamma'_d \end{bmatrix} \begin{bmatrix} f_r \\ \dot{f}_r \end{bmatrix}_i \quad (34)$$

The expression in (34) can be simplified according to (35).

$$\boldsymbol{k}'_{z,\,i+1} = \boldsymbol{k}'_{z,\,i} + \boldsymbol{\delta} \boldsymbol{F}_i \quad (35)$$

where, $\boldsymbol{k'}_{z,i}$ is the vector containing the values of $k_P{}'$ and $k_D{}'$ at i^{th} iteration of a given time-instant, $\boldsymbol{F}_i$ is the vector of contact-force dynamics at i^{th} iteration of a given time-instant, and $\boldsymbol{\delta}$ is the matrix of manually-tuned learning gains of the ILA for compliance controller. Using the condition given in (31), the learning gains are optimally tuned via trial-and-error method to achieve best effort. The evaluated matrices of the learning gains, $\boldsymbol{\varphi}$ and $\boldsymbol{\delta}$ are shown in (36). They are kept constant through the experimentation phase.

$$\boldsymbol{\varphi} = \begin{bmatrix} 11.58 & 2.66 \\ 3.17 & 1.42 \end{bmatrix}, \boldsymbol{\delta} = \begin{bmatrix} 2.09 & 0.85 \\ 0.95 & 0.18 \end{bmatrix} \quad (36)$$

With all the individual control modules appropriately installed in the proposed control architecture, the final block diagram of the adaptive collaborative control mechanism is shown in Fig. 11.

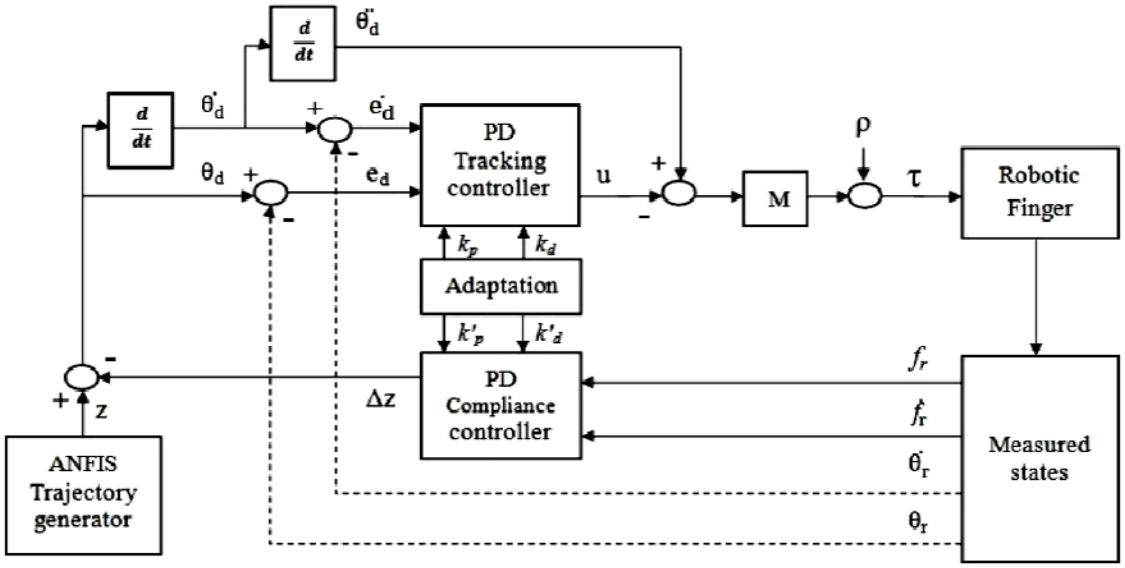

Fig. 11. Adaptive collaborative control mechanism.

5. TESTS AND RESULTS

The controller's trajectory tracking response is analyzed via three different test cases. For each test-case, the proposed control scheme is initially equipped with fixed-gain PD controller, followed by a fuzzy tuned PD (FPD) controller, and eventually with the ILA-based adaptive PD controller (IPD).

The dimensions of all the fingers are only slightly different. Hence, when the MCP joint-motor is actuated according to a given trajectory, each finger exhibits consistent closed-loop tracking response. Therefore, the performances of tracking controllers (and its variants) are experimentally tested on the index finger only.

The GUI application represents the recorded sensor data graphically for the sake of visualization and comparative assessment. In each graphical representation, the reference trajectory and the corresponding angular responses traced by the joint-motor is shown in "blue" and "red" color, respectively.

Test A: The free-space motion of the index finger is tested while tracking the reference trajectory, given by (37), that is applied to the MCP joint-motor of the finger.

$$z = 1.5cos(0.15(t - 8.6)) + 0.08sin(0.9(t - 8.6)) \quad (37)$$

The trajectory tracking response of the generic PD, FPD, and IPD controller is shown in Fig. 12, 13, and 14, respectively.

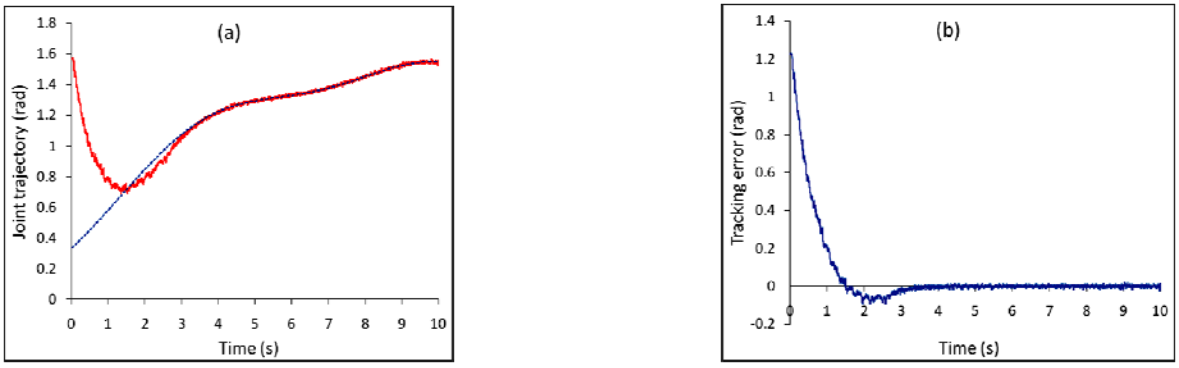

Fig. 12. (a) Trajectory tracking response with PD controller, (b) Tracking error response with PD controller.

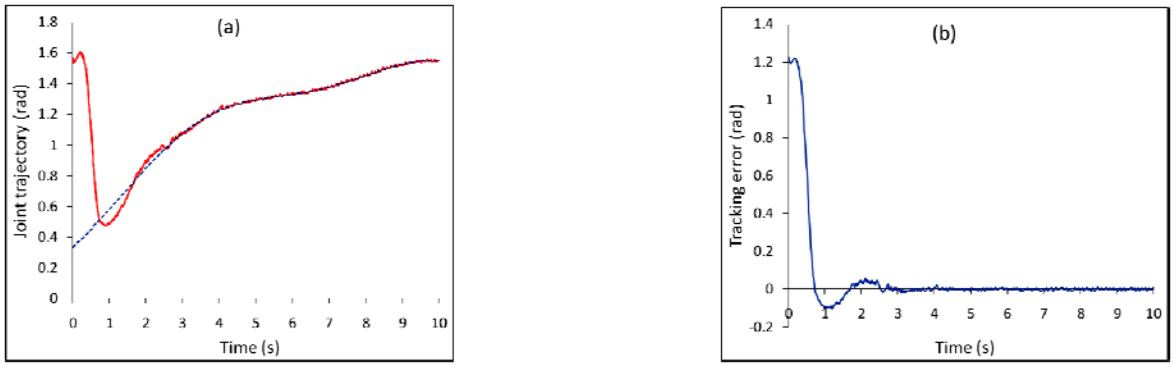

Fig. 13. (a) Trajectory tracking response with FPD controller, (b) Tracking error response with FPD controller.

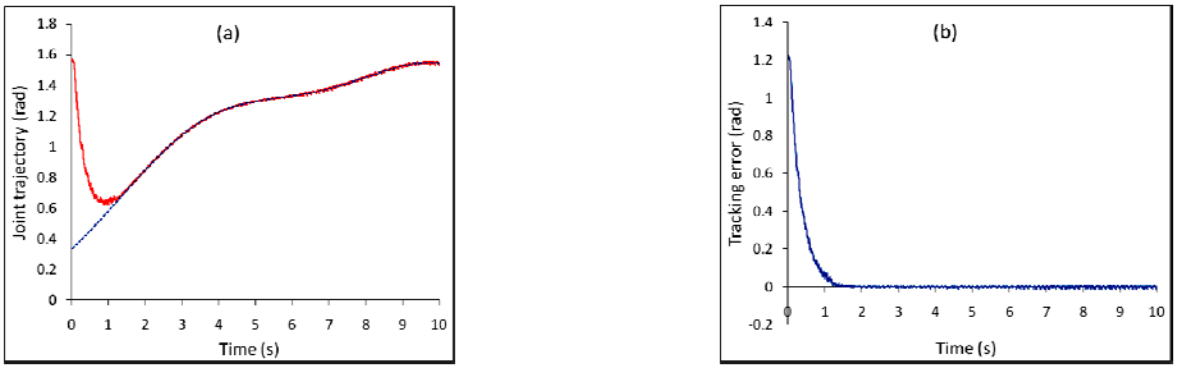

Fig. 14. (a) Trajectory tracking response with IPD controller, (b) Tracking error response with IPD controller.

Test B: The reference input is perturbed by externally applying an external disturbance torque signal, given by (38), to the joint motor while it is tracking the reference trajectory of (37). This time-varying disturbance torque signal is added directly to the finalized motor torque (τ) control commands.

$$\tau_{dist} = 0.05(\cos(2\pi t) + \sin(4\pi t)) \quad (38)$$

The disturbance torque signal is shown in Fig. 15. The trajectory tracking responses and the error variations of the generic PD controller and its adaptive variants are shown in Fig. 16, 17, and 18, respectively.

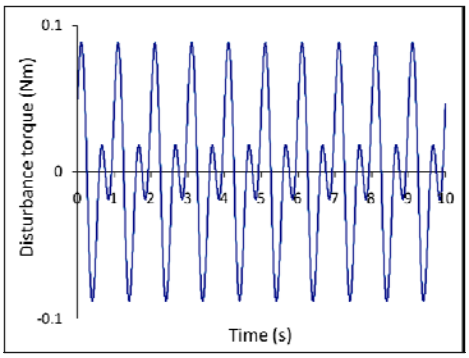

Fig. 15. Disturbance torque signal.

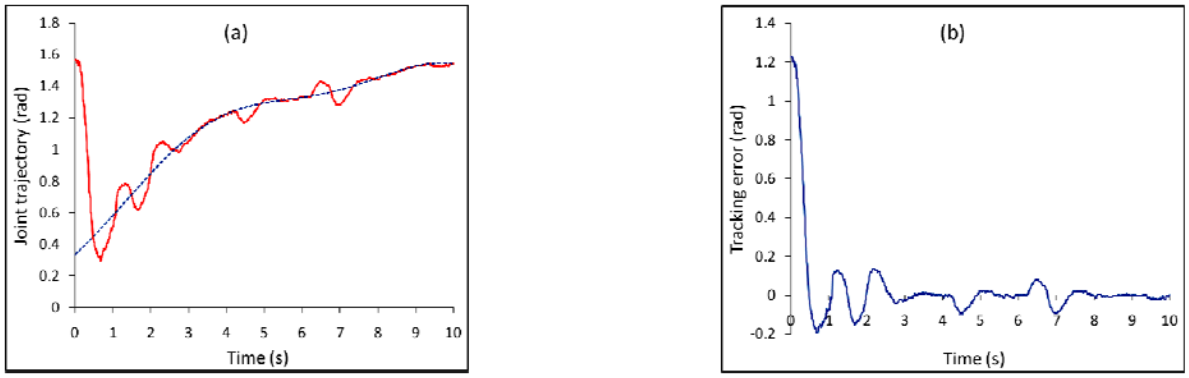

Fig. 16. (a) Trajectory tracking response with PD controller, (b) Tracking error response with PD controller.

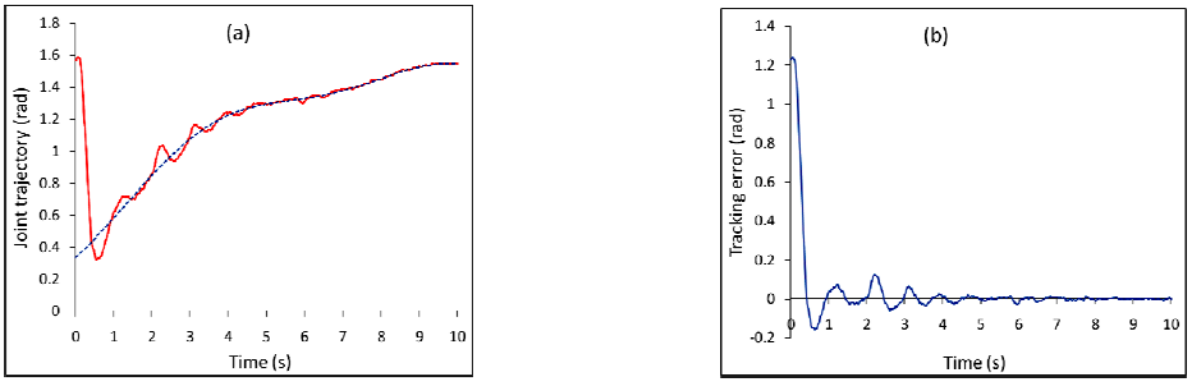

Fig. 17. (a) Trajectory tracking response with FPD controller, (b) Tracking error response with FPD controller.

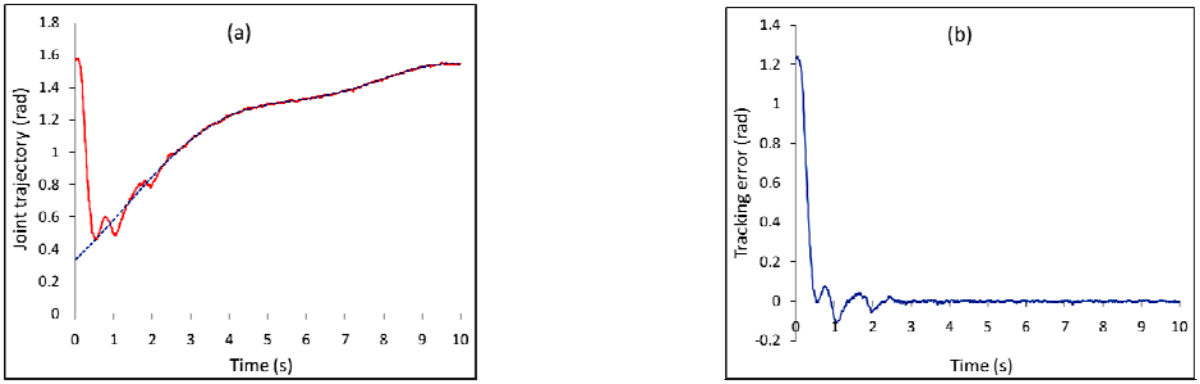

Fig. 18. (a) Trajectory tracking response with IPD controller, (b) Tracking error response with IPD controller.

The performance of the controller is analyzed by the following parameters exhibited by the response.

- *Rise-time (t_r):* The time taken by the response to rise from its initial position and converge to the reference trajectory.
- *Overshoot (OS):* The maximum absolute error incurred in the response after converging to the reference trajectory.
- *Settling time (t_s):* The time taken by the joint-angle response to converge and settle within ±2% of the reference trajectory.
- *Root-mean-square-error (RMSE):* The root of the arithmetic mean of the squared-sum of instantaneous error samples.

These performance parameters are recorded in Table 3 for each of the three control strategies, for the two testing scenarios (A and B). The experimental results clearly manifest that the ILA-based self-tuning of PD tracking controller is superior in performance than the other versions.

Test C: When the robotic hand grasps an object, the contact-force acts a disturbance to the trajectory tracking response of robotic fingers. For this purpose, a compliance controller is employed. Instead of proportionally changing the motor torque control signal, the proposed controller directly modifies the reference trajectory. This technique effectively complies with the applied contact-force (disturbance) and adjusts the tracking control signal accordingly. The compliance of the finger is tested as it makes contact with a soft rubber ball (of radius = 0.08 m), while tracking the trajectory, given by (39).

$$z = 0.5\,cos\,(0.05(t + 0.75)) + 1 \tag{39}$$

The variation in force experienced at the tip of the index finger, upon its contact with the ball, is shown in Fig. 19. The corresponding modified reference trajectories of the index finger provided by the PD, FPD, and IPD based compliance controllers are shown in Fig. 20, 21, and 22, respectively.

The graphical illustrations clearly manifest that the ILA based PD compliance controller surpasses the compliance performance of the other two PD controller variants.

Table 3. Summary of experimental results.

Test	Controller	t_r (s)	OS (rad)	t_s (s)	RMSE (rad)
A	PD	1.09	0.09	3.18	0.13
	FPD	0.66	0.10	2.79	0.09
	IPD	0.75	None	1.17	0.06
B	PD	0.44	0.21	8.07	0.25
	FPD	0.37	0.16	4.92	0.18
	IPD	0.39	0.11	2.20	0.12

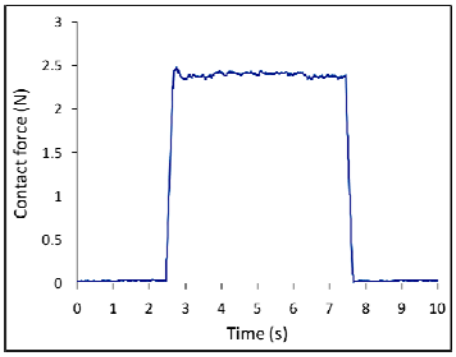

Fig. 19. Contact force applied on the tip of index finger.

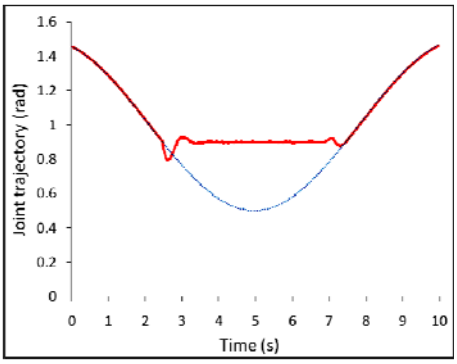

Fig. 20. Modified trajectory using PD compliance controller.

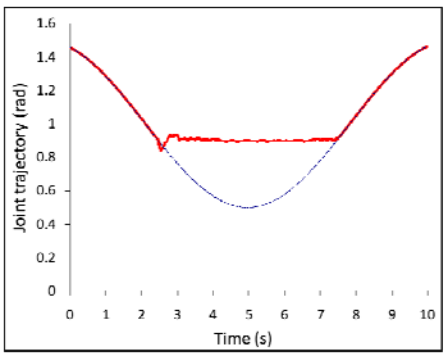

Fig. 21. Modified trajectory using FPD compliance controller.

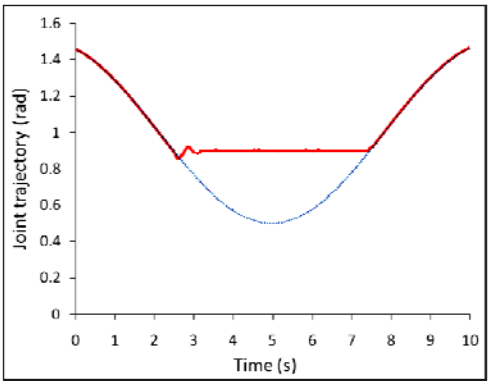

Fig. 22. Modified trajectory using IPD compliance controller.

6. CONCLUSION

The synthesis of an adaptive collaborative position and tracking controller for the finger joint of an FFRH is described in this paper. The ANFIS controller is trained to accurately track the trajectory pattern by delivering the dynamic changes occurring in the reference trajectory in real-time. The tracking error minimization and disturbance rejection is done via a dynamically compensated adaptive PD tracking controller. The individual controller outputs are fused together via a linearized feedback control law to enhance the trajectory tracking performance of the finger during free-space or grasping motion. The tracking control system is augmented by a compliance controller that alters the reference trajectory according to the disturbance caused by the application of contact-force upon grasping a given object. The PD gains of tracking and compliance controller are self-tuned via two different model-free techniques; namely FIS and ILA. In FIS systems, the accuracy of the artificially synthesized rule-base becomes the bottleneck. The heuristically developed rules-base cannot compensate for the inevitable and unprecedented nonlinearities and uncertainties existing in a servo-system. On the other hand, the ILA tuned servomechanism "learns" the PD gains needed to generate the required control action via the repeated process. This self-tuning procedure ensures the quick asymptotic convergence of the PD controllers; allowing them to achieve enhanced trajectory-tracking performance. Three tests are conducted to compare the effect of the two adaptation techniques upon the tracking accuracy exhibited by the proposed control system. The experimental results have validated the superior performance of ILA in robustifying the PD controllers. In future, nonlinear gain modulation and other intelligent tuning techniques can be explored. Model-based optimal controllers can also be analyzed and integrated with the existing PD control system to improve the tracking response. Moreover, the compliance control output can be used to appropriately modify the torque signals being applied to the motors, instead of using it to directly to modify the reference trajectory.

REFERENCES

Angulo, C., Godo, L. (2007). *Artificial intelligence research and development*. IOS Press, Amsterdam, Netherlands.

Arimoto, S., Kawamura, S., Miyazaki, F. (1984). Battering operation of robots by learning. *Journal of Robotic Systems*, 2(1), pp. 123-140.

Asgari, M., Ardestani, M. A. (2013). Trajectory tracking control of a class of lower mobility parallel manipulators using ANFIS. In: *Proceedings of 13th Iranian Conference on Fuzzy Systems*, Qazvin, Iran, pp. 1-6.

Abdallah, M.E., Platt Jr., R., and Wampler, C.W. (2013). Decoupled torque control of tendon-driven fingers with tension management. *The International Journal of Robotics Research*, 32(2), pp. 247-258.

Bakırcıoğlu, V., Sen, M.A., Kalyoncu, M. (2016). Adaptive Neural-Network Based Fuzzy Logic (ANFIS) Based Trajectory Controller Design for One Leg of a Quadruped Robot. In: *Proceedings of the 5th International Conference on Mechatronics and Control Engineering*, Venice, Italy, pp. 82-85.

Baroud, Z., Benmiloud, M., and Benalia, A. (2015). Fuzzy self-tuning PID controller for air supply on a PEM fuel cell system. In: *Proceedings of 4th International Conference on Electrical Engineering*, Boumerdes, Algeria, pp. 1-4.

Bequette, B.W. (2003). *Process Control: Modeling, Design, and Simulation,* pp. 195-209. Prentice Hall, New Jersey, USA.

Bhatti, O.S., Mehmood-ul-Hassan, K., and Imtiaz, M.A. (2015). Attitude Control and Stabilization of a Two-Wheeled Self-Balancing Robot. *Control Engineering and Applied Informatics*, 17(3), pp. 98-104.

Carbone, G., and Ceccarelli, M. (2008). DESIGN OF LARM HAND: PROBLEMS AND SOLUTIONS. *Control Engineering and Applied Informatics*, 10(2), pp. 39-46.

Carbureanu, M. (2014). The Development of a Neuro-Fuzzy Expert System for Wastewater pH Control. *Control Engineering and Applied Informatics*, 16(4), pp. 30-41

Chen, C.H., and Naidu, D.S. (2011). Fusion of fuzzy logic and PD control for a five-fingered smart prosthetic hand. In: *Proceedings of 2011 IEEE International Conference on Fuzzy Systems*, Taipei, Taiwan, pp. 2108-2115.

Chen, C.H., Naidu, D.S., and Schoen, M.P. (2011). Adaptive control for a five-fingered prosthetic hand with unknown mass and inertia. *WSEAS Transactions on Systems*, 10(5), pp. 148-161.

Chen, C.H., and Naidu D.S. (2013). Hybrid control strategies for a five-finger robotic hand. *Biomedical Signal Processing and Control*, 8(4), pp. 382-390.

Chen, C.H., and Naidu D.S. (2014). A modified optimal control strategy for five-finger robotic hand. *International Journal of Robotics and Automation Technology*, 1, pp. 3-10.

Chen, C.H., Naidu, D.S., and Schoen, M.P. (2014). An adaptive control strategy for a five-fingered prosthetic hand. In: *Proceedings of 18th International Conference on System*, Santorini Island, Greece, pp. 405-410.

Chen, C.H., Ze, Y., and Li, C. (2012). Mechatronics design of multi-finger robot hand. In: *Proceedings of 12th International Conference on Control, Automation and Systems*, Jeju Island, South Korea, pp. 1491 -1496.

Duka, A.V. (2015). ANFIS Based Solution to the Inverse Kinematics of a 3DOF Planar Manipulator. *Procedia Technology*, 19, pp. 526-533

Elmerabete, J., Li, H., and Rafi, Y. (2010). Application of Real-Time Control System Using LabVIEW in Distance-Learning. In: *Proceedings of 2010 International Conference on Measuring Technology and Mechatronics Automation*, Shanghai, China, pp. 663-666.

Fadil, M.A., Jalil, N.A., and Darus, I.Z.M. (2013). Intelligent PID controller using iterative learning algorithm for active vibration controller of flexible beam. In: *Proceedings of 2013 IEEE Symposium of Computers and Informatics*, Langkawi, Malaysia, pp. 80-85.

Karray, A., and Feki, M. (2015). Tracking control of a mobile manipulator with Fuzzy PD controller. In: *Proceedings of 2015 World Congress on Information Technology and Computer Applications Congress*, Hammamet, Tunisia, pp. 1-5.

Niehues, T., Badger, J., Diftler, M., and Ashish, D.D. (2014). Cartesian-space control and dextrous manipulation for multi-fingered tendon-driven hand. In: *Proceedings of 2014 IEEE International Conference on Robotics and Automation*, Hong Kong, China, pp. 6777-6783.

Potluri, C., Kumar, P., Anugolu, M., Chiu, S., Urfer, A., Schoen, M.P., and Naidu, D.S. (2010). sEMG based fuzzy control strategy with ANFIS path planning for prosthetic hand. In: *Proceedings of 2010 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, Tokyo, Japan, pp. 413-418.

Rezaeeian, A., Shasti, B., Doosthoseini, A., Yousefi-Koma, A. (2008). ANFIS modeling and feed forward control of shape memory alloy actuators. In: *Proceedings of the 10th WSEAS International Conference on Automatic Control, Modelling & Simulation*, Istanbul, Turkey, pp. 243-248.

Saini, S., and Rani, S. (2012). Temperature Control Using Intelligent Techniques. In: *Proceedings of 2012 Second International Conference on Advanced Computing & Communication Technologies*, Rohtak, India, pp. 138-145.

Seraji, H. (1998). Nonlinear and Adaptive Control of Force and Compliance in Manipulators. *The International Journal of Robotics Research*, 17(5), pp. 467-484.

Shang, W. W., Cong, S., Li, Z. X., and Jinag, S. L. (2009). Augmented nonlinear PD controller for a redundantly actuated parallel manipulator. *Advanced Robotics*, 23, pp. 1725-1742.

Shauri, R.L.A., Salleh, N.M., and Hadi, A.K.A. (2014). PID position control of 7-DOF three-fingered robotic hand for grasping task. In: *Proceedings of 2014 IEEE International Conference on Control System, Computing and Engineering*, Penang, Malaysia, pp. 70-74.

Song, K. T., Hsu, C. H. (2011). A compliance control design for safe motion of a robotic manipulator. In: *Proceedings of 9th World Congress on Intelligent Control and Automation*, Taipei, Taiwan, pp. 920-925.

Su, Y.X., and Zheng. C.H. (2010). Global asymptotic tracking of robot manipulators with a simple decentralised non-linear PD-like controller. *IET Control Theory and Applications*, 4(9), pp. 1605-1611.

Vijayakumar, K., and Manigandan, T. (2016). Nonlinear PID Controller Parameter Optimization using Enhanced Genetic Algorithm for Nonlinear Control System.

Control Engineering and Applied Informatics, 18(2), pp. 3-10.

Widhiada, W., Nindhia, T.G.T., and Budiarsa, N. (2015). Robust Control for the Motion Five Fingered Robot Gripper. *International Journal of Mechanical Engineering and Robotics Research*. 4(3), pp. 226-232.

Xu, J.X., and Huang, D. (2007). Optimal tuning of PID parameters using iterative learning approach. In: *Proceedings of 22nd IEEE International Symposium on Intelligent Control*, Sinagpore, pp. 226-231.

Xu, J.X., Panda, S.K., and Lee, T.H. (2009). *Real Time Iterative Learning Control: Design and Applications*. Springer-Verlag, Londan, England.

Xu, Z., Kumar, V., Matsuoka, Y., and Todorov, E. (2012). Design of an anthropomorphic robotic finger system with biomimetic artificial joints. In: *Proceedings of 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, Rome, Italy, pp. 568 - 574.

Xing, G., Lifeng, W., Zhijun, M. (2015). Fuzzy self-tuning PID controller for an unmanned helicopter. In: *Proceedings of 34th Chinese Control Conference*, Hangzhou, China, pp. 3635-3641.

Zaki, A.M., Soliman, A.M., Mahgoub, O.A., and El-Shafei, A.M. (2010). Design and implementation of efficient intelligent robotic gripper. In: *Proceedings of 2010 International Conference on Modelling, Identification and Control*, Okayama City, Japan, pp. 710-716.