

INVESTIGATION ON THE INFLUENCE OF PARAMETER UNCERTAINTIES IN THE POSITION TRACKING OF ROBOT MANIPULATORS

Habib GHANBARPOUR ASL, Kerim Youde HAN

Abstract: This paper presents a novel trajectory tracking method for robot arms with uncertainties in parameters. The new controller applies the robust output feedback linearization method and is designed so that it is robust to the variation of parameters. Robustness of the algorithm is evaluated when the parameters of the system are floating over 10 percent up and down. An Unscented Kalman Filter (UKF) is applied for state and parameter estimation purposes. As the considered system has 8 unknown parameters while only 5 of them are independent parameters, UKF is applied only to the augmented system with independent parameters. Three types of simulations are applied depending on sensor groups – first with both position and joint sensors, second with only position sensors and third with only joint sensors. The observation of parameters in these groups is discussed. Simulation results show that when both position sensors and joint sensors are used, all the parameters and states are observable and good tracking performances are obtained. When only position sensors are used, the accuracy of the estimated parameters is reduced, and low tracking performances are revealed. Finally, when only joint sensors are applied, the lengths of robot arms are unobservable, but other parameters related to the dynamic system are observable, and poor tracking performances are given.

Keywords: path following; parameter uncertainty; robot control; robust control; sensor fusion; unscented Kalman filter

1 INTRODUCTION

The robot arm (also called Robot manipulator) is a type of mechanical device which is usually programmable, and it usually has similar functions to a human arm. Traditionally, many Proportional-Derivative (PD) or Proportional-Integral-Derivative (PID) controllers were designed for controlling robot arms. Takegaki et al. [1] and Arimoto [2] designed simple PI and PID feedback controllers. A PD controller with stability robustness in the presence of parameter uncertainty in the gravitational torques vector was presented by Hsia in [3]. However, these controllers are difficult to use in determining the appropriate PID gains in the cases of nonlinear and unknown controlled plants. The fixed PID parameters in these controllers may often deteriorate control performances, and while these controllers are enough for the general control, they usually result in weak robustness and poor performances due to the nonlinearity characteristics of robot arms. Research works on nonlinear PD or PID controllers of robot arms such as [4-6] were proposed for these purposes. Huang et al. [6] proposed a nonlinear PD controller with gravity compensation that is globally asymptotically stable in position control and a comparison was made between their proposed controller and the conventional PD controller, which showed that a faster response velocity and higher position accuracy were obtained by the former. Davoud et al. [7] proposed fractional order PID controllers by applying evolutionary algorithms (particle swarm optimization (PSO), the genetic algorithm and estimations of the distribution algorithm) and better tracking results were obtained compared to the normal PID controllers. Nevertheless, the nonlinearity of the kinematics and the dynamics of robot arms is inherited in the robot itself, which means that even if a well precisely calibrated model-based controller may give good tracking performance for a given robot model [8], the difficulty of having an exact model of a robot arm makes the calibrated controllers unable to

adapt to any changes and uncertainties in its model and environment. Under these circumstances, every time the robot arm picks up some tools of different dimensions, unknown orientations or gripping points, the overall kinematics and dynamics of the robot arm will also change, which requires the derivation of a new robot arm model, as well as the designing of a controller. It is also not possible for the robot arm to grip the tool at the same grasping point and orientation site, even if the same tool is used again. To solve these problems, many controllers with stronger robustness have been developed, such as back-stepping control [9], neural network control [10-12], fuzzy logic control [13], adaptive control [14, 15], sliding mode control [16, 17] and robust control [18, 19]. These controllers can achieve more accurate trajectory tracking results with faster convergent speed even under presence of various disturbances. Among these controllers, the focus of this paper will be concentrated on the adaptive and robust controller design of the robot arm. An important point in adaptive control is that the tracking error will converge regardless of whether the trajectory persistently exists or not [20, 21]. Many adaptive controllers [22-25] were designed for robot arms. However, these adaptive controllers all assumed that the kinematics of the robot arm was known in advance, and no uncertainties in kinematics and dynamics of robot arms were considered. To deal with uncertainty issues, Cheah et al. [26] developed an adaptive controller that can give concurrent adaptation to both kinematics and dynamic uncertainties of the robot arm by measuring its end-effector position, joint angles and joint velocities. Torres et al. [27] developed a controller that added an adaptive scheme to the standard robust controller in order to improve its performance, especially when unknown parameters or unknown variable loading exist. Yin et al. [28] applied the dynamic linearization technique and designed a controller that used only the input/output data of robotic manipulator systems. This paper will present a new adaptive control-based tracking method of a robot arm when there are

uncertainties in robot arm parameters. The output linearization method will also be applied so that the proposed controller will be robust to the variation of the parameters. In order to estimate the states and parameters, UKF is applied. The standard UKF was introduced in [29], and the implementation of this filter was proposed numerically with robustness in [30, 31]. Different combinations of sensors are to be used for validating the effectiveness of the proposed adaptive and robust controller. The rest of this paper is organized as follows: in Section 2, the problem is defined and the adaptive robust inverse controller is designed. UKF is used in Section 3 to estimate the states and parameters of the robot arm. Section 4 gives a numerical example to validate the effectiveness of the proposed controller. Conclusions and future research are given in Section 5.

2 PROBLEM DEFINITION AND CONTROLLER DESIGN

In general, the dynamics of a robot manipulator is given in the form as follows:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = u \quad (1)$$

Where $\theta \in \mathcal{R}_n$ denotes the set of configuration variables of the robot arm, $u \in \mathcal{R}_n$ is the torques applied at the joints. $M(\theta)$ is the moment of the inertia matrix, $C(\theta, \dot{\theta})$ is the Coriolis/centripetal vector, and $G(\theta)$ is the gravity vector. The objective of the design problem here is to develop a suitable controller so that the robot arm can track the desired trajectory as close as possible.

As a starting point, we apply the input-output linearization method, where the model of the system is summarized in its state space representation form as follows [32]:

$$\begin{aligned} \dot{\theta} &= \omega \\ \dot{\omega} &= M^{-1}(u - f) \\ z &= g(\theta) \end{aligned} \quad (2)$$

where $f = C(\theta, \dot{\theta}) + G(\theta)$ is the summation of Coriolis/centrifugal and gravity effect, and z is the output of the system representing the position of the end-effector (forward kinematic). By taking the derivative of the output functional equation in terms of the time t , it results in

$$\dot{z} = \frac{\partial g(\theta)}{\partial \theta} \dot{\theta} = J\dot{\theta} \quad (3)$$

Here, J is the Jacobian matrix of the robot arm. Substitute $\dot{\theta}$ in the Eq. (2) into Eq. (3) and take the derivative of \dot{z} in terms of time t again, which results in the following representation:

$$\ddot{z} = J\ddot{\theta} + \dot{J}\dot{\theta} = J\ddot{\omega} + JM^{-1}(u - f) \quad (4)$$

Meanwhile, let us define the output error dynamics as:

$$\dot{e} = z_{ref} - z \quad (5)$$

where z_{ref} is the desired reference path that is to be tracked. By doing so, the output error will include an integrator, which will increase the type of the system and let the error converge to zero in a steady state condition. Differentiate (5) twice and substitute (3) into (5), which will lead to

$$\ddot{e} = \ddot{z}_{ref} - \dot{J}\omega - JM^{-1}(u - f) \quad (6)$$

Let us suppose that the desired error dynamic equation is given by

$$\ddot{e} + K_2\dot{e} + K_1\dot{e} + K_0e = 0_{n \times n} \quad (7)$$

Here, K_0, K_1 and K_2 are assumed to be $n \times n$ matrices, so the characteristic equation is $Is^3 + K_2s^2 + K_1s + K_0 = 0$, where I is a $n \times n$ unit matrix. Substitute (6) into (7), it results in

$$u = MJ^{-1}(\ddot{z}_{ref} - \dot{J}\omega + K_2\dot{e} + K_1\dot{e} + K_0e) + f \quad (8)$$

By selecting proper values for K_0, K_1 and K_2 , the error dynamics become stable, and the controller can be defined in terms of (8). In real situations, however, the error dynamics will not be of the form of (7) due to parameter and state errors. Thus, input function can be defined as:

$$u = \tilde{M}\tilde{J}^{-1}(\ddot{z}_{ref} - \dot{\tilde{J}}\tilde{\omega} + K_2\dot{\tilde{e}} + K_1\dot{\tilde{e}} + K_0\tilde{e}) + \tilde{f} \quad (9)$$

Where \tilde{M} , \tilde{J} , $\dot{\tilde{J}}$, and \tilde{f} are the approximated or estimated values of M , J , \dot{J} and f , and $\tilde{e} = e + \Delta e$, $\tilde{\omega} = \omega + \Delta\omega$, Δe and $\Delta\omega$ are the errors of the output and angular velocity estimation. Next, substitute (9) into (6), it results in

$$\ddot{e} = \ddot{z}_{ref} - \dot{J}\omega - JM^{-1}[\tilde{M}\tilde{J}^{-1}(\ddot{z}_{ref} - \dot{\tilde{J}}\tilde{\omega} + K_2\dot{\tilde{e}} + K_1\dot{\tilde{e}} + K_0\tilde{e}) + \tilde{f} - f] \quad (10)$$

Rearranging (10) results in

$$\begin{aligned} \ddot{e} + \tilde{I}K_2\dot{\tilde{e}} + \tilde{I}K_1\dot{\tilde{e}} + \tilde{I}K_0\tilde{e} &= \Delta u \\ \Delta u &= (I - \tilde{I})\ddot{z}_{ref} + (\dot{\tilde{J}} - \dot{J})\omega + \tilde{I}\begin{pmatrix} \dot{J}\Delta\omega - K_2\Delta\dot{e} \\ +K_1\Delta\dot{e} + K_0\Delta e \end{pmatrix} - JM^{-1}(\tilde{f} - f) \end{aligned} \quad (11)$$

where $\tilde{I} = JM^{-1}\tilde{M}\tilde{J}^{-1}$. If there is not any error in the parameters or states of the system, the matrix \tilde{I} should be a unit matrix I , and Δu will be zero. The stability of error dynamics is affected only by the matrix \tilde{I} . When errors appear in the parameters and states, the input of error equation will be Δu (non-zero function). Therefore, for the robustness of the proposed controller Eq. (9), we need an estimator to estimate the parameters and states of the system so that the tracking error will go to zero.

The values of K_0, K_1 and K_2 can be selected for system stability when parameters are varied in the range. By doing so, we have a stable control system, and errors will arise in the steady state due to an unknown input error. In order to reduce the steady state errors, in Section 3, an unscented Kalman filter will be applied to estimate the states and parameters of the robot arm system.

At this stage, for calculating the controller gains that are represented in the Eq. (7), first we convert that equation in the state space form.

$$\begin{bmatrix} \ddot{e} \\ \dot{e} \\ e \end{bmatrix} = A \begin{bmatrix} \dot{e} \\ e \end{bmatrix} + BU, \quad A = \begin{bmatrix} 0 & 0 & 0 \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \tilde{I} \\ 0 \\ 0 \end{bmatrix} \quad (12)$$

here

$$U = K \begin{bmatrix} \ddot{e} \\ \dot{e} \\ e \end{bmatrix} = [K_2 \quad K_1 \quad K_0] \begin{bmatrix} \ddot{e} \\ \dot{e} \\ e \end{bmatrix} \quad (13)$$

The matrix A , does not have uncertainty, but the matrix B is an uncertain matrix. The matrix B can be written as:

$$B = B_0 + \Delta B, \quad \Delta B = DFE, \quad F^T F \leq I \quad (14)$$

Here, the matrices D, E and F are the known matrices. One can determine the gain K through the solution of the linear matrix inequality equation represented in (15) [34].

$$\begin{bmatrix} (AS + BW) + (AS + BW)^T + \varepsilon DD^T & (EW)^T \\ EW & -\varepsilon I \end{bmatrix} < 0 \quad (15)$$

If the Eq. (15) has a feasible solution, in terms of variables ($\varepsilon > 0, S, W$), then the state feedback law represented by (13) is robustly stabilizing the system (12) with gain

$$K = [K_2 \quad K_1 \quad K_0] = WS^{-1} \quad (16)$$

Consequently, the beefcake (9) can guarantee the stability of the system (1), but the system will have a steady state error. For reducing steady state errors, we need to estimate the parameters of the system.

3 STATE AND PARAMETER ESTIMATION

In this section, a two degree of freedom robot arm is taken into consideration, the schematic diagram of which is given in Fig. 1. The manipulator has 8 unknown constant parameters, which are $L_1, L_2, m_1, m_2, r_1, r_2, I_1$ and I_2 , representing lengths, masses, radius and the moment of inertia of the first and second link, respectively. r_1 and r_2 are the positions of the center of mass, m_1 and m_2 are the masses of links. The kinetic and potential energy equations are given by the following equation

$$T = \frac{1}{2} [\dot{\theta}_1 \quad \dot{\theta}_2] \begin{bmatrix} \alpha + 2\beta c_2 & \delta + \beta c_2 \\ \delta + \beta c_2 & \delta \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}, \quad V = 0, \quad L = T - V \quad (17)$$

T, V and L represent the kinetic energy, potential energy and Lagrangian of the manipulator, and $c_i = \cos\theta_i, i = 1, 2$. For simplicity, in this paper, time-based parameters will only be written with the parameters themselves, but omitting time t in notation. For instance, $\theta_i = \theta_i(t)$. Meanwhile, α, β and δ are represented by [33]:

$$\begin{aligned} \alpha &= I_1 + I_2 + m_1 r_1^2 + m_2 (L_1^2 + r_2^2) \\ \beta &= m_2 L_1 r_2 \\ \delta &= I_2 + m_2 r_2^2 \end{aligned} \quad (18)$$

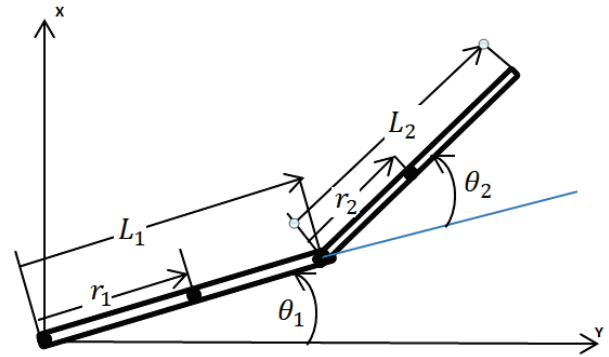


Figure 1 Robot arm parameters

By substituting the Lagrangian $L = T$ into Lagrange's equations, matrices M, C and G will become the function of $p^s = [\alpha \quad \beta \quad \delta]^T$. By discretizing the dynamic Eq. (2) by the 4th order of the Runge Kutta method, the system equation is then given as

$$\theta_k = f(\theta_{k-1}, u_{k-1}, p_{k-1}^s), \quad p_k^s = p_{k-1}^s \quad (19)$$

Let us suppose that $x_k = [\theta_k^T \quad (p_k^s)^T]^T$ is the state of the augmented system. The model of the augmented system (14) is then considered as:

$$x_k = f(x_{k-1}, u_{k-1}) \quad (20)$$

For identification and control purposes, two types of sensors are considered. One of them is the joint sensor and the other one is the position sensor. The modeling of these sensors is as follows:

$$z_k = [I_{2 \times 2} \quad 0_{2 \times 3}] x_k = g(x_k) \quad (21-1)$$

$$y_k = \begin{bmatrix} L_1 c_1 + L_2 c_{12} \\ L_1 s_1 + L_2 s_{12} \end{bmatrix} = h(x_k, p^m) \quad (21-2)$$

where $c_i = \cos\theta_i, s_i = \sin\theta_i, c_{12} = \cos(\theta_1 + \theta_2)$ and $s_{12} = \sin(\theta_1 + \theta_2)$. z_k and y_k are the output of joint sensors and position sensors respectively; $p^m = [L_1 \quad L_2]^T$ is the unknown parameter that appears in the position sensor output.

If only two sensors are used for measuring the angles of links, (20) will be the system equation and (21-1) will be the measurement equation. Therefore, in the best condition, p^s will be identifiable (when the inputs are suitable) and p^m will not appear in any measurement of the system equation. Consequently, it remains unidentifiable. As a result, tracking errors will arise in the position control of the end-effector due to the errors which arise from L_1 and L_2 , and that makes it impossible to get zero tracking errors without applying external sensors or measuring the end-effector's position.

When the position sensor is available, (20) acts as a system dynamic and (21-1) and (21-2) are the measurement equations, in which case both p^s and p^m will be active, and they can both be identified in the best condition. Consequently, when both the joint and position sensors are available, tracking errors will be reduced. Let us put two unknown parameter vectors of p^s and p^m into a vector form $p = [(p^s)^T (p^m)^T]^T$, and consider a random walk model [31] for the unknown parameter vector. Then the augmented system model is

$$\begin{aligned} \theta_k &= f(\theta_{k-1}, u_{k-1}, p_{k-1}) \\ p_k &= p_{k-1} + w_k \\ z_k &= g(x_k) + v_k \\ y_k &= h(x_k, p) + \nu_k \end{aligned} \quad (22)$$

Where w_k and v_k and ν_k are zero mean white noise signals with the covariances of $E[w_k(w_k)^T] = Q_k$, $E[v_k(v_k)^T] = R_v$ and $E[\nu_k(\nu_k)^T] = R_\nu$, respectively. $X_k = [\theta_k^T p_k^T]^T$, as an augmented state space representation of the system in (22), can be re-written as:

$$\begin{aligned} X_k &= F(X_{k-1}, u_{k-1}) + Gw_k \\ z_k &= HX_k + v_k \\ y_k &= h(X_k) + \nu_k \end{aligned} \quad (23)$$

where $H = [I_{2 \times 2} \quad 0_{2 \times 7}]$, and $G = \begin{bmatrix} 0_{4 \times 5} \\ I_{5 \times 5} \end{bmatrix}$. Due to the fact that the filter falls off reducing the covariance of the system, some noises are added here for holding the robustness of the system, which will increase the covariance of unknown parameters.

The following method will provide an augmented UKF method for parameter and state estimation.

Table 1 Unscented Kalman Filter

<ul style="list-style-type: none"> Initialization: $\hat{X}_0 = E(X_0), P_0 = E[(X - X_0)(X - X_0)^T]$ for $k = 1 \dots \infty$ set $t = k - 1$ calculate sigma points $x_t = [\hat{X}_t \quad \hat{X}_t + \gamma S_t \quad \hat{X}_t - \gamma S_t]$ time update equations $x_{i,k t} = F(x_{i,t}, u_{k-1})$ $\hat{X}_k^- = w_i^m x_{i,k t}$ $P_k^- = \sum_{i=1}^{2l} w_i^c [x_{i,k t} - \hat{X}_k^-] [x_{i,k t} - \hat{X}_k^-]^T + GQ_kG^T$ $y_{i,k t} = h(x_{i,k t})$ $y_k^- = \sum_{i=1}^{2l} w_i^m y_{i,k t}$ When position sensor data is available:

$$\begin{aligned} P_{\hat{y}_k \hat{y}_k} &= \sum_{i=1}^{2l} w_i^c [y_{i,k|t} - y_k^-] [y_{i,k|t} - y_k^-]^T + R_\nu \\ P_{\hat{x}_k \hat{y}_k} &= \sum_{i=1}^{2l} w_i^c [x_{i,k|t} - \hat{X}_k^-] [y_{i,k|t} - y_k^-]^T \\ K_k &= P_{\hat{x}_k \hat{y}_k} (P_{\hat{y}_k \hat{y}_k})^{-1} \\ \hat{X}_k &= \hat{X}_k^- + (y_k - y_k^-) K_k \\ P_k &= P_k^- - K_k P_{\hat{y}_k \hat{y}_k} K_k^T \\ \hat{X}_k^- &= \hat{X}_k, \quad P_k^- = P_k \end{aligned}$$

- When joint sensor data is available:

$$\begin{aligned} P_{\hat{z}_k \hat{z}_k} &= HP_k^- H^T + R_v \\ K_k &= P_k^- H^T (HP_k^- H^T + R_v)^{-1} \\ \hat{X}_k &= \hat{X}_k^- + (z_k - H\hat{X}_k^-) K_k \\ P_k &= P_k^- - P_k^- H^T P_{\hat{z}_k \hat{z}_k} H P_k^- \end{aligned}$$

End for

Where l is the summation of the lengths of states and parameters, w_i^m and w_i^c are a set of scales weights as follows:

$$w_0^m = \frac{\lambda}{\lambda + l}, w_0^c = \frac{\lambda}{\lambda + l} + (1 - \tilde{\alpha}^2 + \tilde{\beta}) \quad (24-1)$$

$$w_i^m = w_i^c = \frac{\lambda}{2(\lambda + l)}, \quad i = 1, \dots, 2l \quad (24-1)$$

where $\lambda = l(\tilde{\alpha}^2 - 1)$ and $\gamma = \sqrt{l + \lambda}$ are the scaling parameters. The constant $\tilde{\alpha}$ determines the spread of the sigma points around the estimated state and is usually set to $10^{-4} \leq \tilde{\alpha} \leq 1$, $\tilde{\beta}$ is used to add prior knowledge of the distribution of the state (for Gaussian distribution, $\tilde{\beta} = 2$ is optimal). S is the square root matrix of P (Choleski factor of P).

4 EXAMPLE APPLICATION

In this section, the MATLAB simulation will be used to demonstrate the effectiveness of the proposed controller. We will start with a two-link robot arm whose system model is given by (25).

$$\begin{bmatrix} \alpha + 2\beta c_2 & \delta + \beta c_2 \\ \delta + \beta c_2 & \delta \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -\beta s_2 \dot{\theta}_2 & -\beta s_2 (\dot{\theta}_1 + \dot{\theta}_2) \\ \beta s_2 \dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (25)$$

Let us consider the parameters of this robot arm that are given by the values in the Tab. 2.

Table 2 Robot arm parameters

Parameter	Unit	Value
L_1	m	0.5
L_2	m	0.5
r_1	m	0.2
r_2	m	0.25
m_1	kg	1.5
m_2	kg	1
I_1	kg - m ²	0.15
I_2	kg - m ²	0.1

The initial conditions of the system are given in Tab. 3.

Table 3 State parameters without errors

Parameter	Unit	Value
θ_1	rad	0
θ_2	rad	0.5π
ω_1	rad/sec	0
ω_2	rad/sec	0

The independent parameters that are calculated from Tab. 2, through (18), are represented in the Tab. 4.

Table 4 States as calculated parameters without errors

Calculated Parameters	Unit	Value
α	$kg - m^2$	0.6225
β	$kg - m^2$	0.125
δ	$kg - m^2$	0.1625

From the basic 8 parameters given in Tab. 2, three independent parameters of α , β and δ that are defined in (18) will be calculated as 0.6225, 0.1250 and 0.1625 respectively as given in Tab. 4. Let us suppose that all independent parameters have their own uncertainties, and let us suppose that the estimation of parameters and states are shown in the Tab. 5.

Table 5 Estimated parameters of the robot arm

Parameter	Unit	Error range or standard deviation	Mean Value
\hat{L}_1	m	± 0.05	0.45
\hat{L}_2	m	± 0.05	0.55
$\hat{\alpha}$	$kg - m^2$	± 0.06	0.6845
$\hat{\beta}$	$kg - m^2$	± 0.01	0.1375
$\hat{\delta}$	$kg - m^2$	± 0.01	0.1463
$\hat{\theta}_2$	rad	0.01	0.001
$\hat{\theta}_2$	rad	0.01	$0.5\pi + 0.003$
$\hat{\omega}_1$	rad/sec	0.01	0.02
$\hat{\omega}_2$	rad/sec	0.01	0.03

The Standard Deviation (STD) of the sensors' noises used in this simulation example is also given in Tab. 6.

Table 6 Standard deviation of sensors

Joint sensors	Joint 1	0.01 rad
	Joint2	0.01 rad
Position sensors	x-axis	0.05 m
	y-axis	0.05 m

To design the controller's gain, here we apply the Monte Carlo simulation method. For this purpose, we generate 10 thousand time parameters of the system with uniform distribution within the range given in Tab. 5 for determining the structured uncertain model represented by (14), and then the controller gain is calculated as

$$K = \begin{bmatrix} -3.7156 & -0.0000 & -2.3804 & -0.0000 & -0.4145 & -0.0000 \\ -0.0000 & -3.7156 & 0.0000 & -2.3804 & 0.0000 & -0.4145 \end{bmatrix}$$

In order to test the robustness of the controller, by generating the above-mentioned data, and by calculating \tilde{M} and \tilde{J} , we can find the poles of random selected systems. The poles of this system are shown in Fig. 2.

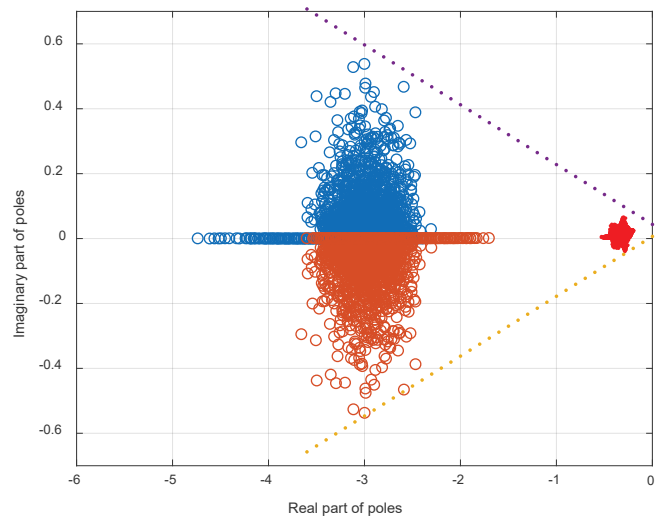


Figure 2 Poles of the uncertain system

It is seen that the nominal poles of a closed-loop system are two repeated poles at $-2.9575, -0.4154$ and -0.3235 . The slowest pole is -0.29 , the fastest pole is -4.8 and the minimum damping of the system is 0.95 for an uncertain closed-loop system. These conditions are acceptable.

It is worth mentioning that, when the filter identifies all parameters after some time, the poles of the system will approach the nominal poles. Hence, the best way for the robot arm to give small tracking errors is to give more time to the filter for identifying the parameters of system. When the error estimations of α , β and δ approach zero, the roots of the closed-loop system approach the nominal poles. Meanwhile, when the parameter errors of L_1 and L_2 approach zero, the tracking error will decrease.

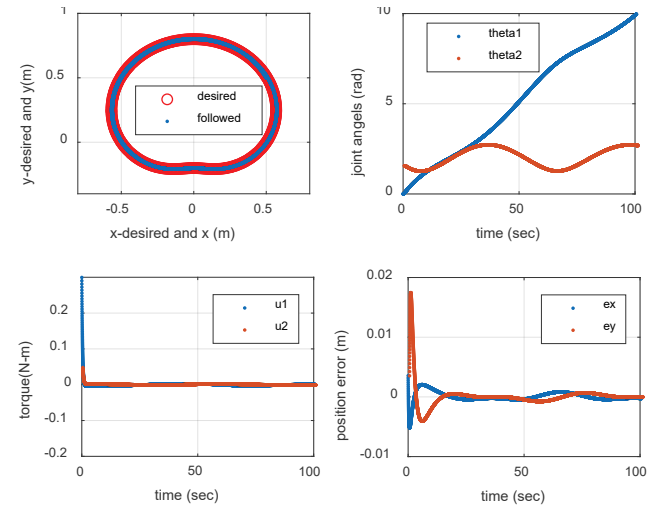


Figure 3 Trajectory Tracking of Robot Arm

Furthermore, to evaluate the power of the system in path tracking, a desired path is generated for tracking purposes. The equation of the desired path is in its polar coordinated form, which is characterized by $r = 0.5 + 0.3\sin\theta$, $\theta = \theta_0 + \omega t$, $\omega = \omega_0 + \dot{\omega}\cos(0.01t)$, $\omega_0 = 0.1\text{rad}$, $\dot{\omega} = 0.01\text{rad/sec}$, $\theta_0 = 0.7854$. When all

parameters of the robot are known, and the measurements are perfect, performances of the controller are presented in Figure 3. The desired and tracked path of the robot is given in Figure 3.a. Figure 3.b also shows that the first arm is rotated more than one time, but the second arm is oscillated between $1.28 - 2.74rad$. Figures 3.c and 3.d show that the applied torques and tracking errors of the robot are always lower than $0.3 N - m$ and $0.02m$ respectively. In all graphs, the blue and red colors represent the first and second parameters respectively.

The first analysis will be made when both joint angles and position sensors are applied. The position tracking error and position estimating error is represented in Fig. 4.a and Fig. 4.b respectively. Here, the minimum position error is $0.025m$ and finally, it reduces to a very small value that is lower than $1 \times 10^{-3} m$. The joint angles and angular velocities and those 3δ bounds are shown in Fig. 5. Errors are in 3δ bounds, which is why the filter worked well. Fig. 6 represents the estimation of unknown parameters, the STD of parameters, errors of α, β, δ and errors of L_1 and L_2 . It is shown that all parameters of the system are estimated correctly and the standard deviation of all parameters is close to zero. Therefore, in this case, good estimation and tracking are obtained.

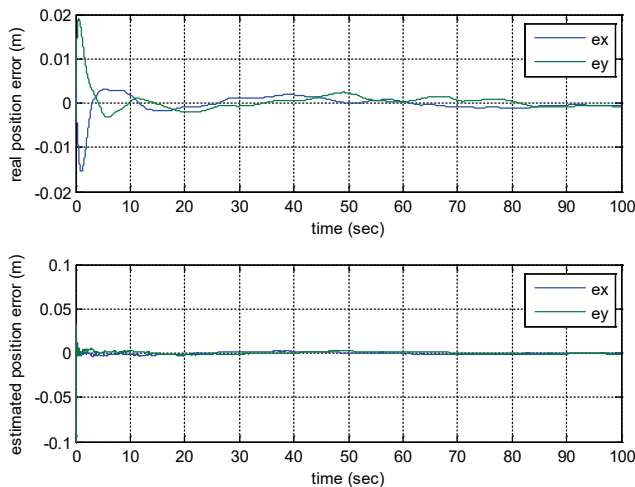


Figure 4 Errors of Position and Estimated Position

The second simulation is done when there is only a position sensor, so the robot arm cannot find two solutions of inverse dynamics (elbow up and down) without using the historical data of the position sensor. Furthermore, it needs to have small errors in the initial condition of the robot arm angles. The position errors and position estimation errors are shown in Fig. 7. Their maximum position error is near $0.02m$. The joint angles errors and its 3δ bounds and angular velocities of joint angles and its 3δ bounds are represented in Fig. 8. The errors of joint angles and angular velocities are higher than in the previous case. Nevertheless, all estimations are in the 3δ bounds. In Fig.9, the estimated parameters, STD of parameters, estimation error of α, β, δ and estimation error of L_1 and L_2 are shown. The results show that the estimation errors of parameters are larger than of the previous case, but

all errors are finally bounded, which indicates that tracking only with position sensors results in bounded errors.

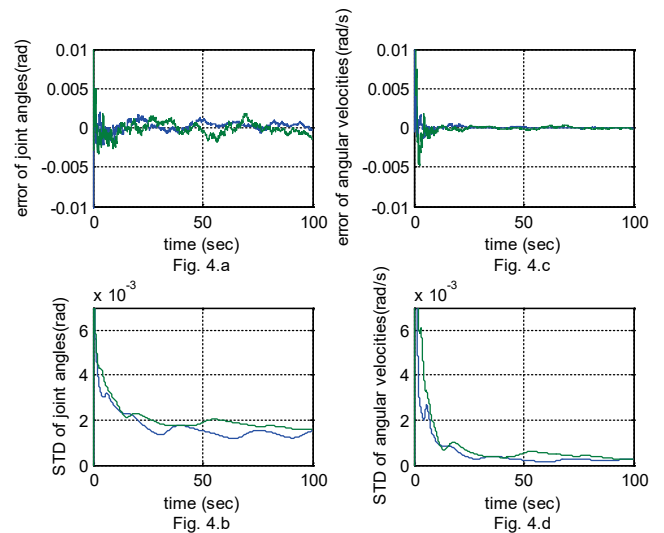


Figure 5 Joint Angle and Joint Angle Rates

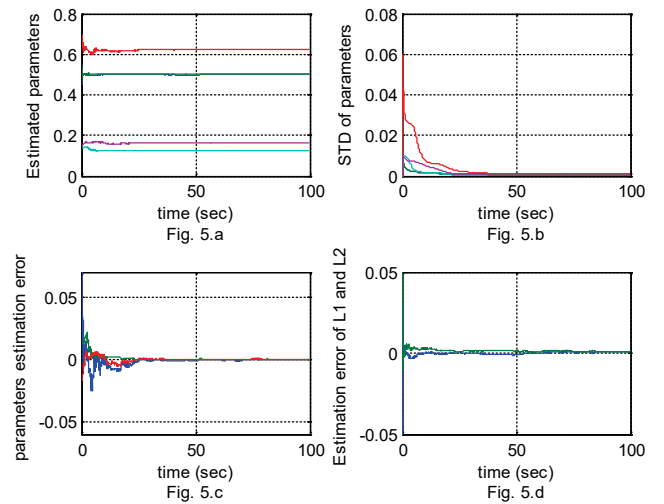


Figure 6 Estimated Parameters and Errors

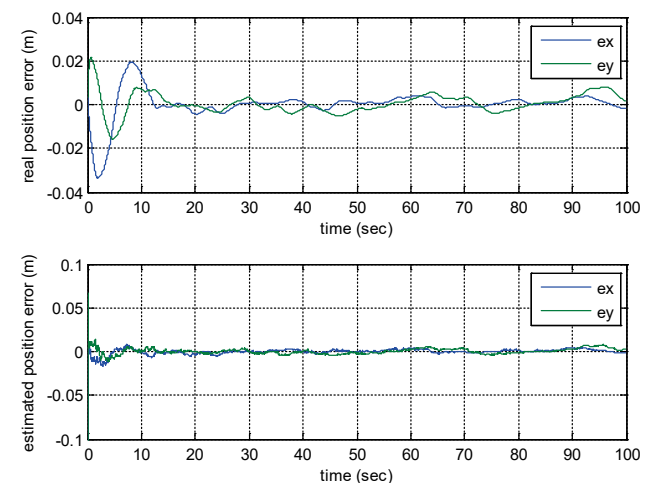


Figure 7 Errors of Position and the Estimated Position

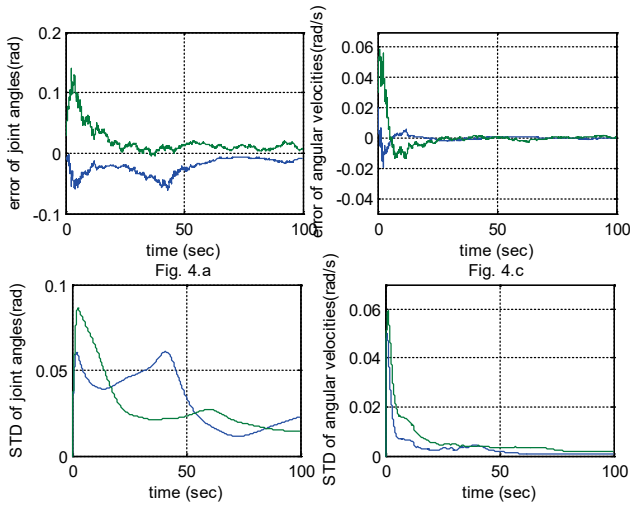


Figure 8 Joint Angle and Joint Angle Rates

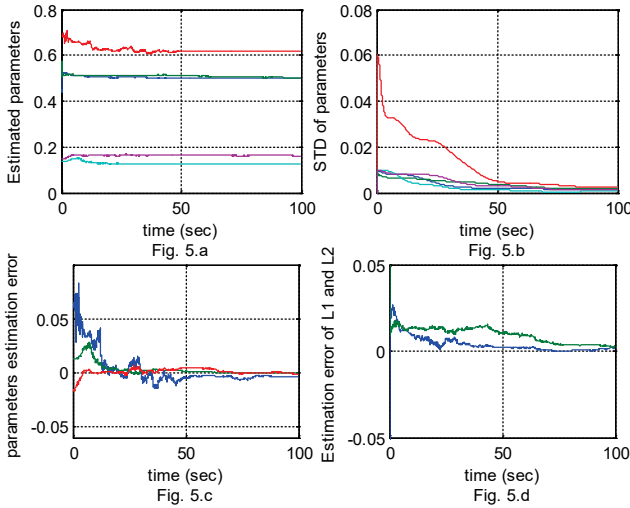


Figure 9 Estimated Parameters and Errors

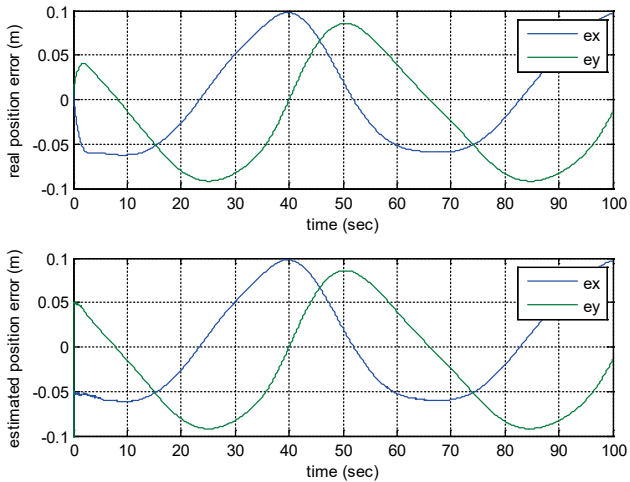


Figure 10 Errors of Position and the Estimated Position

In the third case, only joint sensors are used. The results of position tracking and estimation are shown in Fig. 10. The joint angle and angular velocities and those 3δ bounds are

represented in Fig. 11. The estimated parameters and those 3δ bounds are shown in Fig. 12. The results show that position tracking errors are large, but the errors of joint angles and joint velocities are very small and in 3δ bounds. In addition to that, the parameter errors, standard deviation of errors, estimation errors of α, β, δ and estimation errors of L_1 and L_2 are represented in Fig. 12. It shows that errors of α, β, δ are small, but the filter is not able to estimate the value of L_1 and L_2 . Therefore, large position tracking errors will always appear due to the errors of L_1 and L_2 , and then these parameters are not identifiable.

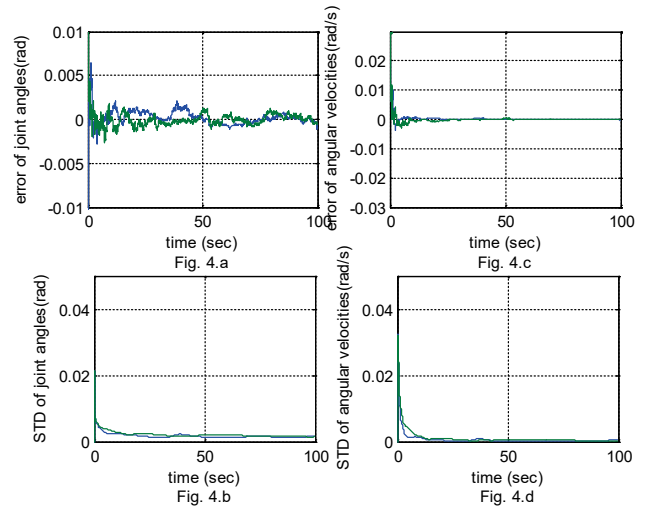


Figure 11 Joint Angle and Joint Angler Rates

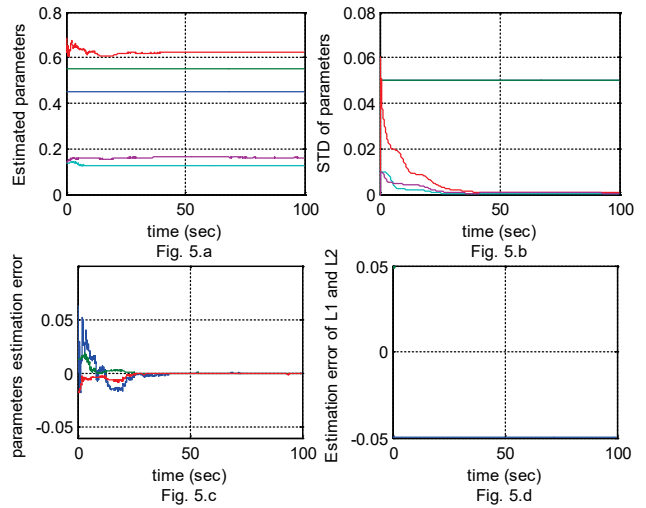


Figure 12 Estimated Parameters and Errors

4 CONCLUSIONS

In this paper, a new adaptive robust inverse control algorithm is developed for the position tracking of the end-effector of a 2 DOF robot arm. Analyses were done on this controller under uncertain conditions. It has been shown that the stability of the controller depends only on the independent parameters of the system, and under the considered range of parameters, the proposed controller is stable. To achieve better tracking results, all independent

parameters need to be identified. An augmented UKF is used for state and parameter estimation purposes. This filter is able to identify all parameters and states when joint sensors, as well as an external position sensor, are used to measure the position of the end-effector, which is how the best tracking performance is achieved. When only the external position sensor is available, all states are observable, but parameters have large errors. Nonetheless, tracking errors are getting very close to zero. Finally, when only joints sensors are present, but without any external sensors, states become observable but L_1 and L_2 are not observable and poor tracking results are obtained due to the error which arises from estimating the position of the end-effector. It is worthwhile to mention that when only a position sensor is present, errors of the initial condition will be important because two solutions will appear in this case. These two solutions are of the link up and down. When the initial condition does not have large errors, the algorithm will converge to the true condition. Future work can be carried out to apply the proposed analyzing method to robot manipulators with a higher DOF to finalize the role of position sensors in robot manipulator position tracking problems.

4 REFERENCES

- [1] Takegaki, M. & Arimoto, S. (1981). A new feedback method for dynamic control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2), 119-125. <https://doi.org/10.1115/1.3139651>
- [2] Youcef-Toumi, K. & Wu, S. T. (1992, June). Robustness and stability analysis of time delay control. In *American Control Conference, 1992* (pp. 2691-2695). IEEE. <https://doi.org/10.23919/ACC.1992.4792631>
- [3] Hsia, T. C. (1994). Robustness analysis of a PD controller with approximate gravity compensation for robot manipulator control. *Journal of Robotic Systems*, 11(6), 517-521. <https://doi.org/10.1002/rob.4620110606>
- [4] Kelly, R. & Carelli, R. (1996). A class of nonlinear PD-type controllers for robot manipulators. *Journal of Robotic Systems*, 13(12), 793-802. [https://doi.org/10.1002/\(SICI\)1097-4563\(199612\)13:12<793::AID-ROB2>3.0.CO;2-Q](https://doi.org/10.1002/(SICI)1097-4563(199612)13:12<793::AID-ROB2>3.0.CO;2-Q)
- [5] Seraji, H. (1998). A new class of nonlinear PID controllers with robotic applications. *Journal of Robotic Systems*, 15(3), 161-181. [https://doi.org/10.1002/\(SICI\)1097-4563\(199803\)15:3<161::AID-ROB4>3.0.CO;2-O](https://doi.org/10.1002/(SICI)1097-4563(199803)15:3<161::AID-ROB4>3.0.CO;2-O)
- [6] Huang, J., Yang, C., & Ye, J. (2014). Nonlinear pd controllers with gravity compensation for robot manipulators. *Cybernetics and Information Technologies*, 14(1), 141-150. <https://doi.org/10.2478/cait-2014-0011>
- [7] Fani, D. & Shahraki, E. (2016). Two-link robot manipulator using fractional order PID controllers optimized by evolutionary algorithms. *Biosciences Biotechnology Research Asia*, 13(1), 589-598. <https://doi.org/10.13005/bbra/2075>
- [8] Craig, J. J. (1989). Introduction to robotics. Addison-Wesley, New York.
- [9] Oh, J. H. & Lee, J. S. (1999). Control of flexible joint robot system by backstepping design approach. *Intelligent Automation & Soft Computing*, 5(4), 267-278. <https://doi.org/10.1080/10798587.1999.10750607>
- [10] Shuzhi, S. G., Hang, C. C., & Woon, L. C. (1997). Adaptive neural network control of robot manipulators in task space. *IEEE transactions on industrial electronics*, 44(6), 746-752. <https://doi.org/10.1109/41.649934>
- [11] Cheng, L., Hou, Z. G., & Tan, M. (2009). Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model. *Automatica*, 45(10), 2312-2318. <https://doi.org/10.1016/j.automatica.2009.06.007>
- [12] Al Ashi, M. M. (2014). Trajectory Tracking Control of A 2-DOF Robot Arm Using Neural Networks. *MS. c, Islamic University of Gaza, Electrical Engineering Department*.
- [13] Goléa, N., Goléa, A., Barra, K., & Bouktir, T. (2008). Observer-based adaptive control of robot manipulators: Fuzzy systems approach. *Applied Soft Computing*, 8(1), 778-787. <https://doi.org/10.1016/j.asoc.2007.05.011>
- [14] Seo, D. & Akella, M. R. (2009). Non-certainty equivalent adaptive control for robot manipulator systems. *Systems & Control Letters*, 58(4), 304-308. <https://doi.org/10.1016/j.sysconle.2008.11.008>
- [15] Craig, J. J., Hsu, P., & Sastry, S. S. (1987). Adaptive control of mechanical manipulators. *The International Journal of Robotics Research*, 6(2), 16-28. <https://doi.org/10.1177/027836498700600202>
- [16] Zeinali, M. & Notash, L. (2010). Adaptive sliding mode control with uncertainty estimator for robot manipulators. *Mechanism and Machine Theory*, 45(1), 80-90. <https://doi.org/10.1016/j.mechmachtheory.2009.08.003>
- [17] Jin, M., Lee, J., Chang, P. H., & Choi, C. (2009). Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control. *IEEE Transactions on Industrial Electronics*, 56(9), 3593-3601. <https://doi.org/10.1109/TIE.2009.2024097>
- [18] Bascetta, L. & Rocco, P. (2010). Revising the robust-control design for rigid robot manipulators. *IEEE Transactions on Robotics*, 26(1), 180-187. <https://doi.org/10.1109/TRO.2009.2033957>
- [19] Kolhe, J. P., Shaheed, M., Chandar, T. S., & Talole, S. E. (2013). Robust control of robot manipulators based on uncertainty and disturbance estimation. *International Journal of Robust and Nonlinear Control*, 23(1), 104-122. <https://doi.org/10.1002/rnc.1823>
- [20] Arimoto, S. (1996). *Control Theory of Non-Linear Mechanical Systems: A Passivity-Based and Circuit-Theoretic Approach (Oxford Engineering Science Series)*. Oxford University Press.
- [21] Slotine, J. J. E. & Li, W. (1987). On the adaptive control of robot manipulators. *The international journal of robotics research*, 6(3), 49-59. <https://doi.org/10.1177/027836498700600303>
- [22] Ortega, R. & Spong, M. W. (1989). Adaptive motion control of rigid robots: A tutorial. *Automatica*, 25(6), 877-888. [https://doi.org/10.1016/0005-1098\(89\)90054-X](https://doi.org/10.1016/0005-1098(89)90054-X)
- [23] Sadegh, N. & Horowitz, R. (1990). Stability and robustness analysis of a class of adaptive controllers for robotic manipulators. *The International Journal of Robotics Research*, 9(3), 74-92. <https://doi.org/10.1177/027836499000900305>
- [24] Whitcomb, L., Arimoto, S., Naniwa, T., & Ozaki, F. (1996). Experiments in adaptive model-based force control. *IEEE Control Systems*, 16(1), 49-57. <https://doi.org/10.1109/37.482150>
- [25] Lee, K. W. & Khalil, H. K. (1997). Adaptive output feedback control of robot manipulators using high-gain observer. *International Journal of Control*, 67(6), 869-886. <https://doi.org/10.1080/002071797223839>
- [26] Cheah, C. C., Liu, C., & Slotine, J. J. E. (2006). Adaptive tracking control for robots with unknown kinematic and

- dynamic properties. *The International Journal of Robotics Research*, 25(3), 283-296.
<https://doi.org/10.1177/0278364906063830>
- [27] Torres, S., Méndez, J. A., Acosta, L., & Becerra, V. M. (2006). Adaptive robust controller for robot manipulators: experiments on a PUMA 560 robot. *IFAC Proceedings Volumes*, 39(15), 437-442. <https://doi.org/10.3182/20060906-3-IT-2910.00074>
- [28] Yanling, Y. (2015). Model free adaptive control for robotic manipulator trajectory tracking. *The Open Automation and Control Systems Journal*, 7(1).
<https://doi.org/10.2174/1874444301507010358>
- [29] Julier, S., Uhlmann, J., & Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on automatic control*, 45(3), 477-482.
<https://doi.org/10.1109/9.847726>
- [30] Van Der Merwe, R. & Wan, E. A. (2001). The square-root unscented Kalman filter for state and parameter-estimation. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings (ICASSP'01). 2001 IEEE International Conference on*, Vol. 6, 3461-3464. <https://doi.org/10.1109/ICASSP.2001.940586>
- [31] Asl, H. G. & Pourtakdoust, S. H. (2007). UD covariance factorization for unscented Kalman filter using sequential measurements update. *World Academy of Science, Engineering and Technology*, 34, 368-376.
- [32] Khalil, H. K. & Grizzle, J. W. (2002). *Nonlinear systems* (Vol. 3). Upper Saddle River, NJ: Prentice hall.
- [33] Murray, R. M. (2017). *A mathematical introduction to robotic manipulation*. CRC press.
<https://doi.org/10.1201/9781315136370>
- [34] Obaiah, M. C. (2014). *Multiobjective output feedback controller compare with IMC-based PID controller* (Doctoral dissertation).

Authors' contact:

Habib GHANBARPOUR ASL, Assistant Professor
University of Turkish Aeronautical Association
Department of Mechatronics
Bahçekapı Mahallesi, Okul Sk. No:11, 06790 Etimesgut/Ankara, Turkey
Tel: +905060214718,
E-mail: hghanbarpourasl@thk.edu.tr, habib.ghanbarpour@gmail.com

Kerim Youde HAN, Assistant Professor
Çankaya University
Department of Mechatronics
Yukarıyurtçu Mahallesi Eskişehir Yolu 29. Km, Mimar Sinan Caddesi No: 4, 06790
Etimesgut, Turkey
Tel: +905422047928,
E-mail: kerimyoudehan@gmail.com