

A Robust Feedback Linearization Control Framework Using an Optimized Extended Kalman Filter

Ali Medjghou^{*}, Mouna Ghanai and Kheireddine Chafaa

(LAAAS) Laboratory, Electronics Department, Faculty of Technology, University of Batna 2, Batna, 05000, Algeria.

Received 8 August 2017; Accepted 10 October 2017

Abstract

A robust nonlinear controller based on an improved feedback linearization technique with state observer is developed for a class of nonlinear systems with uncertainties and external disturbances. First, by combining classical feedback linearization approach with a discontinuous control and a fuzzy logic system, we design and study a robust controller for uncertain nonlinear systems. Second, we propose an optimized extended Kalman filter (EKF) for the observation of the states. The parameters to be optimized are the covariance matrices Q and R , which play an important role in the EKF performances. The particle swarm optimization algorithm insures this optimization. Lyapunov synthesis approach is used to prove the stability of the whole control loop. The proposed approach is applied on a two-link robot system under Matlab environment. Simulation results have confirmed the effectiveness of the proposed approach against uncertainties and external disturbances; and exhibited a more superior performance than the non-improved control actions.

Keywords: Feedback linearization, discontinuous control, fuzzy logic, extended Kalman filter, particle swarm optimization.

1. Introduction

The control design has played an increasingly important role in industrial applications and in advanced science i.e., mechanical engineering systems, aerospace and robotics. The main objectives for control system design are stability, good tracking and disturbance rejection [1]. Feedback linearization is an approach to nonlinear control design which has attracted a great interest of researchers in recent years [2-4]. The basic idea of the approach is to algebraically transform a nonlinear system dynamics into a fully, or partially linear one, and then linear control techniques can be used. In the classical feedback linearization [5], the presence of uncertainties can perturb the function of the feedback linearization controller which can lead to system instabilities. In the nonlinear control design, the question of how to handle the parametric uncertainty and disturbances is one of the important issues in the control theory. In this context, we find in the literature that classical feedback linearization was combined with some control approaches to solve the problem of robustness [2-4, 6-12].

In practice, the state variables of a given system are rarely available for direct measurement. In most cases, there's a real needing for reliably estimate unmeasured states; the elaboration of a control law of a given system often requires access to the value of one or more of its states. For this reason, it is necessary to design an auxiliary dynamic system; named observer, capable to deliver state estimates from the measurements provided by physical sensors and applied inputs. In the case of linear systems, the

solution to observer's synthesis problem was completely resolved by Kalman [13] and Luenberger [14]. Contrarily, in nonlinear systems, there's not a general solution to the problem of observer synthesis which prompted researchers to develop nonlinear observers. Several algorithms on this subject can be found in the literature, namely extended Luenberger observer [15,16], extended Kalman filter [17,18], sliding mode observer (SMO) [19], model reference adaptive system [20], artificial neural network observer [21] and fuzzy logic observer [22,23].

Amongst all these algorithms, EKF provides the optimal state estimator due to its ability to consider the stochastic uncertainties. EKF is a recursive algorithm based on the knowledge of the statistics of both measurement and state noises. Compared to other nonlinear observers [24], EKF algorithm has better dynamic behavior, resistance to uncertainties and noise, and it can work even in the presence of a standstill conditions. Estimation performance is the major problem associated to EKF; it strongly influences the parameter values of the system, state and measurement noise covariance matrices Q and R , respectively. Following [25], Q and R have to be acquired by taking into account the stochastic properties of the corresponding noises that is why in most cases Q and R are usually unknown matrices. However, since these are usually not known, in most cases, the covariance matrices are used as weighting factors (factors adjustment). Moreover, these matrices were first tuned manually by trial-error methods which are very tedious procedures due to large time consumption [26]. To overcome this problem and to avoid the computational complexity of trial-error method, authors in [27] have used genetic algorithms (AGs) to optimize these matrices automatically.

^{*}E-mail address: medjghou.ali@gmail.com

Two contributions will be proposed in this investigation:

- (1) First, a new form of feedback linearization, called improved feedback linearization controller. This controller combines classical feedback linearization, discontinuous control and fuzzy logic. In the proposed approach, a feedback linearization control law is first designed for control purposes (stability, trajectory tracking) using pole placement. Then, a discontinuous control is added to guarantee that the state reaches the sliding mode in the presence of parameter uncertainties and external disturbances. Fuzzy logic system is employed to improve control performance and to reduce chattering phenomenon in the sliding mode.
 - (2) Second, an optimized EKF observer for system states estimation in which the optimization of EKF matrices (Q and R) is ensured by an alternative optimization method proposed in [28] which is an evolutionary algorithm inspired by social interactions, that relates to particle swarm optimization (PSO) algorithm.
- This paper is structured as follows. In Section 2, we present the problematic and a detailed explanation of the proposed method. Simulation results are conducted in Section 3. Finally, conclusions are given in Section 4.

2. Problem formulation and proposed method

2.1 Problem statement

Consider the n^{th} order nonlinear uncertain system which is described by:

$$\begin{cases} \dot{x}^{(n)} = f(X) + g(X)u + d \\ y = x \end{cases} \quad (1)$$

where $x^{(n)}$ is the n^{th} time derivative of x , $y \in \mathbb{R}$ is the output of the system, $u \in \mathbb{R}$ is the control, d represents the sum of the parametric uncertainties and external disturbances, f and g are both unknown real continuous functions assumed to be bounded. We suppose that the system state vector $X = (x_1, x_2, \dots, x_n)^T = (x, \dot{x}, \dots, x^{(n-1)})^T \in \mathbb{R}^n$ is unavailable for measurement and it will be estimated by the EKF (see Fig. 1). The control objective is to find a control law $u = u(\hat{X})$ such that even in the presence of external disturbances and modeling imprecision, the state vector X will track a given desired bounded reference trajectory $X_d = (x_d, \dot{x}_d, \dots, x_d^{(n-1)})^T$.

In respect of the dynamic system presented in Eq.1, the following assumptions will be made:

Assumption 1. The function f is unknown but the error on its estimate is bounded, i.e. $|f(\hat{X}) - f(X)| \leq F$ where $f(\hat{X})$ is an estimate of $f(X)$.

Assumption 2. The input gain g is unknown but positive and bounded, i.e. $0 < g_{\min} \leq g(X) \leq g_{\max}$.

Assumption 3. The disturbance d is unknown but bounded, i.e. $|d| \leq D$ where $D > 0$.

Assumption 4. The desired trajectory X_d is once differentiable in time. Furthermore, every element of vector X_d , as well as $\dot{x}_d^{(n)}$ is available and with known bounds.

Concerning the nonlinear control problem, we propose to use the feedback linearization approach [29]. This choice is motivated by its ability to controlling nonlinear systems and its design simplicity.

2.2 Proposed control design

Let $e = \hat{x} - x_d$ be the tracking error, therefore, the feedback linearization control law with EKF algorithm can be computed easily as follows

$$u^* = \frac{1}{g(\hat{X})} \left(-f(\hat{X}) - k_{feed}^T \underline{e} + \ddot{x}_d \right) \quad (2)$$

where $\underline{e} = (e, \dot{e})^T$ is the tracking error vector and $k_{feed} = (k_1, k_2)^T$ is chosen such that $R(p) = p^2 + k_2 p + k_1$ is a Hurwitz polynomial.

By combining control law Eq.2 and system Eq.1, we get the following error dynamics:

$$\ddot{e} + k_2 \dot{e} + k_1 e = 0 \quad (3)$$

in which the main objective will be $\lim_{t \rightarrow \infty} e(t) = 0$.

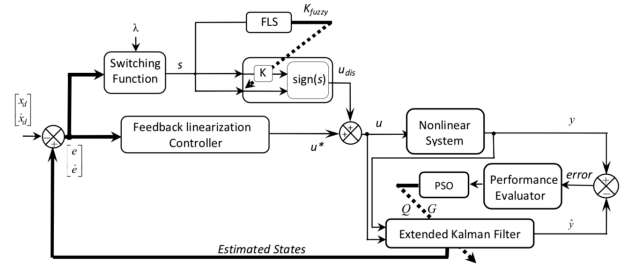


Fig. 1. Schematic diagram of EKF optimization based proposed control approach

The presence of uncertainties and disturbances can perturb the feedback linearization controller working; therefore, system dynamics may lead to instabilities like static errors (see [5]). To overcome this problem, we propose to improve this control by adding a discontinuous control as shown in Fig. 1. Discontinuous control can be found in sliding mode control. This choice is motivated by its high robustness against uncertainties and disturbances. So, the whole control law will be constituted of two terms: feedback linearization control term (u^*) and a discontinuous control term (u_{dis}) as follows:

$$u = u^* + u_{dis} = \frac{1}{g(\hat{X})} \left(-f(\hat{X}) - k_{feed}^T \underline{e} + \ddot{x}_d \right) - K \text{sign}(s) \quad (4)$$

where

s is defined by a sliding surface and described by the state space equations $s(e) = 0$, such that:

$$s(e) = \left(\frac{\partial}{\partial t} + \lambda \right)^{(r-1)} e \quad (5)$$

with $s : R^n \rightarrow R$ [29].

In the two dimensional case ($r = 2$), Eq.5 becomes

$$s = \dot{e} + \lambda e \quad (6)$$

So, the time derivative of s will be

$$\dot{s} = \ddot{e} + \lambda \dot{e} \quad (7)$$

The use of the discontinuous sign function will excite an undesired phenomenon called chatter caused by the discontinuous switching function. In this context high switching gain K of u_{dis} in Eq.4 will lead to an increase in oscillations of the control input signal, and therefore an excitation of high frequency dynamics, consequently, a chattering phenomenon will be created. Moreover, a low switching gain K can reduce the chattering phenomenon and improve the tracking performance despite uncertainties and external disturbances. However, increasing the gain causes an increase of the oscillations in input control around the sliding surface. To achieve more appropriate performance, this gain must be adjusted. This adjustment is based on the distance between the system states and the sliding surface. That is to say, the gain should be high when the state trajectory is far from the sliding surface, and when the distance decreases, it should be reduced. This idea can be realized by combining fuzzy logic with discontinuous control to adjust the gain K adaptively (see Fig. 1) according to some appropriate fuzzy rules.

For this reason, a one-input one-output FLS is designed with an input s which reflects the distance of the error trajectory to the sliding surface. The output of the fuzzy system is denoted by K_{fuzzy} .

An FLS, consists of four parts: the knowledge base, the fuzzifier, the fuzzy inference engine, and the defuzzifier. The knowledge base is composed of a collection of fuzzy If-then rules whose rules can be stated in a linguistic manner as follows:

$$R^l : \text{If } s \text{ is } A^l, \text{ Then } K_{fuzzy} \text{ is } B^l, l = 1, 2, \dots, N$$

where A^l and B^l are fuzzy sets, which are associated with the fuzzy membership functions $\mu_{A^l}(s)$ and $\mu_{B^l}(K_{fuzzy})$, respectively and N is the total number of rules.

Note that the singleton fuzzifications, center average defuzzification, Mamdani implication and product inference engine are used in this paper. Therefore, the output of the fuzzy system could be described by the following equation:

$$K_{fuzzy}(s) = \frac{\sum_{l=1}^N \xi_l \left(\prod_{j=1}^n \mu_{A_j^l}(s_j) \right)}{\sum_{l=1}^N \left(\prod_{j=1}^n \mu_{A_j^l}(s_j) \right)} \quad (8)$$

where n is the number of system states, ξ_l is the centre of gravity of the membership function of K_{fuzzy} for the l^{th} rule.

Therefore, the control law Eq.4 becomes:

$$\begin{aligned} u &= u^* + K_{fuzzy} u_{dis} \\ &= \frac{1}{g(\hat{X})} \left(-f(\hat{X}) - k_{feed}^T \underline{e} + \ddot{x}_d \right) - K_T \text{sign}(s) \end{aligned} \quad (9)$$

with K_{fuzzy} is the output of the FLS as shown in Fig. 1 and therefore the final gain becomes $K_T = K_{fuzzy} \times K$.

Based on Assumptions 1– 3 and considering that the estimate $g(\hat{X})$ could be chosen according to the geometric mean $g(\hat{X}) = \sqrt{g_{\min} g_{\max}}$, the bounds of $g(X)$ may be expressed as $\beta^{-1} < g(\hat{X}) / g(X) < \beta$ where $\beta = \sqrt{g_{\max} / g_{\min}}$. Under this condition, the gain K_T should be chosen according to:

$$K_T \geq \beta g^{-1}(\hat{X}) (\eta + D + F) + g^{-1}(\hat{X}) (|\hat{u}_1| - \beta |\hat{u}_2|) \quad (10)$$

where η is a strictly positive constant related to the reaching-time.

In order to dominate the states of the system to arrive to the sliding surface $s = 0$ in a limited time and to stay there, the control law must be designed so that the sliding condition described in Eq.11 is satisfied.

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \quad (11)$$

This goal is assured by the following lemma.

Lemma:

Consider the uncertain nonlinear system Eq.1 and Assumptions 1–4. If the control input u is selected as Eq.9 and by considering K_T as Eq.10, then, the previous condition Eq.11 is satisfied, which ensures the convergence of the tracking error vector to the sliding surface s .

Proof:

Consider the Lyapunov function candidate

$$L = (1/2) s^2 \quad (12)$$

Its time derivative is given as

$$\begin{aligned} \dot{L} &= \frac{1}{2} \frac{d}{dt} s^2 = s \dot{s} = (\ddot{e} + \lambda \dot{e}) s = (\ddot{x} - \ddot{x}_d + \lambda \dot{e}) s \\ &= (f(X) + g(X)u + d - \ddot{x}_d + \lambda \dot{e}) s \\ &= \left[f(X) + g(X) g^{-1}(\hat{X}) (-f(\hat{X}) + \ddot{x}_d - k_{feed}^T \underline{e}) - g(X) K_T \text{sign}(s) + d - \ddot{x}_d + \lambda \dot{e} \right] s \end{aligned}$$

Noting that: $f(X) = f(\hat{X}) - [f(\hat{X}) - f(X)]$,

$\hat{u}_1 = -f(\hat{X}) + \ddot{x}_d - k_{feed}^T \underline{e}$ and $\hat{u}_2 = -f(\hat{X}) + \ddot{x}_d - [0, \lambda]^T \underline{e}$ one has

$$\begin{aligned} \dot{L} = & \left[-\left(f(\hat{X}) - f(X)\right) + d + g(X) g^{-1}(\hat{X}) \hat{u}_1 \right. \\ & \left. - \hat{u}_2 - g(X) K_T \text{sign}(s) \right] s \end{aligned} \quad (13)$$

Therefore, considering Assumptions 1 and 2, and defining K_T according to Eq.10, \dot{L} becomes

$$\dot{L} \leq -\eta|s| \quad (14)$$

which implies $L(t) \leq L(0)$. From the definition of s in Eq.6, it can be verified that e is bounded. Thus, assumption 4, Eqs.6 and 7 imply that s and \dot{s} are also bounded.

The finite time convergence of the tracking error vector to the sliding surface to the sliding surface can be represented as follows:

$$\dot{L} = \frac{1}{2} \frac{d}{dt} s^2 = s\dot{s} \leq -\eta|s| \quad (15)$$

Then, dividing by $|s|$ and integrating both sides over the interval $0 < t < t_s$, in which t_s is the time required to hit s , gives

$$\begin{aligned} \int_0^{t_s} \frac{\dot{s}}{|s|} s d\tau & \leq -\int_0^{t_s} \eta d\tau \\ |s(t=t_s)| - |s(t=0)| & < -\eta t_s \end{aligned}$$

In this way, considering t_{reach} as the time required to hit s and noting that $|s(t_{reach})| = 0$, one has

$$t_{reach} \leq \frac{|s(0)|}{\eta} \quad (16)$$

and, consequently, the finite time convergence to the sliding surface s .

2.3 Extended Kalman filter

Extended Kalman filter is a generalization of the Kalman filter which is a stochastic observer for nonlinear dynamical systems. In this paper, we shall attempt to find the best estimate of the state vector X_k of the system which evolves according to the following discrete-time nonlinear dynamic:

$$\begin{cases} X_{k+1} = f(X_k, u_k, w_k) \\ Z_k = h(X_k, v_k) \end{cases} \quad (17)$$

where $f(\cdot)$ represents the evolution function of the system, $h(\cdot)$ represents the relationship between the state vector and the measurement result Z_k , whereas u_k stands for the control input to the system at discrete time k , and w_k and v_k are the process and measurement white Gaussian noise

vectors with zero mean and with associated covariance matrices $Q = E[w_k, w_k^T]$ and $R = E[v_k, v_k^T]$, respectively.

To apply EKF to the nonlinearity given by Eq.17, it must be linearized by using first order Taylor approximation near a desired reference point $(\hat{X}_k, \hat{w}_k = 0, \hat{v}_k = 0)$, which gives the following approximated linear model:

$$\begin{cases} X_{k+1} \approx f(X_k, u_k, w_k) \approx f(\hat{X}_k, u_k, 0) + F_k(X_k - \hat{X}_k) + W_k(w_k - 0) \\ Z_k \approx h(X_k, v_k) \approx h(\hat{X}_k, 0) + H_k(X_k - \hat{X}_k) + V_k(v_k - 0) \end{cases} \quad (18)$$

where F_k , W_k , H_k and V_k are the Jacobean matrices given by:

$$\begin{aligned} F_k &= \left. \frac{\partial f(X, 0)}{\partial X} \right|_{X=\hat{X}}, \quad W_k = \left. \frac{\partial f(\hat{X}, w)}{\partial w} \right|_{w=0}, \quad H_k = \left. \frac{\partial h(X, 0)}{\partial X} \right|_{X=\hat{X}} \\ \text{and } V_k &= \left. \frac{\partial h(\hat{X}, v)}{\partial v} \right|_{v=0} \end{aligned} \quad (19)$$

The EKF is a recursive algorithm that is used for estimating state vector of the nonlinear dynamical systems. It consists of two parts, namely, the prediction and measurement correction. It can be described as follows:

$$\text{Prediction: } \begin{cases} \hat{X}_{k+1/k} = f(\hat{X}_{k/k}, u_k, 0) \\ P_{k+1/k} = F_k P_{k/k} F_k^T + W_k Q W_k^T \end{cases} \quad (20)$$

Computation of the Kalman Gain K_k as

$$K_k = P_{k+1/k} H_k^T (H_k P_{k+1/k} H_k^T + V_k R V_k^T)^{-1} \quad (21)$$

Then update the state estimate and predict the state covariance as

Correction:

$$\begin{cases} \hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_k (Z_k - h(\hat{X}_{k+1/k}, 0)) \\ P_{k+1/k+1} = P_{k+1/k} - K_k H_k P_{k+1/k} \end{cases} \quad (22)$$

where $\hat{X}_{k+1/k}$ denotes the priori state prediction vector, $\hat{X}_{k+1/k+1}$ is the posteriori state prediction vector, $P_{k+1/k}$ denotes the priori prediction error covariance matrix, $\hat{X}_{k+1/k}$ is the posteriori prediction error covariance matrix. Therefore, the functional representation of EKF algorithm is depicted in Fig. 1.

Determination of matrices Q and R is a difficult task, especially when the corresponding noises have unknown stochastic properties. In order to avoid this problem, these matrices will be considered as free parameters to be adjusted. In the literature, Bolognani S, and al. (1999) were the first who adjusted these matrices manually with trial-error method [26]. Unfortunately, this method is a tedious task. Therefore, to overcome this difficulty and to avoid trial-error method, authors in [27] have used genetic

algorithms to optimize these matrices automatically. In our work, we suggest to use a recently proposed method for the adjusting and optimization of covariance matrices Q and R by using the PSO algorithm [28].

2.4 Particle Swarm Optimization algorithm

Particle Swarm Optimization (PSO) was developed by Kennedy and Eberhart in 1995. The main idea of PSO algorithm was based on the simulation of simplified social models such as bird flocking or fish schooling. The PSO is used in a wide range of applications such as optimization; a framework for the optimization algorithm has been developed based on PSO algorithm.

In this work, the main PSO task is depicted in Fig. 1, where we running it in an offline way with EKF in order to efficiently find the optimal covariance matrices Q and R , we will call this combination PSO-EKF. Mean square error (MSE) criterion defined in Eq.23 is used in this paper as fitness/objective function between the actual output and the estimated one according to a certain number of iterations N to be performed for each step of estimation.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (23)$$

where \hat{y} is an estimate of the output y and N denotes the number of data samples.

The control input u and the measured response y will be considered as input signals to EKF observer, where u is applied to both nonlinear system and extended Kalman filter.

The Actual output y and the estimated output \hat{y} are set to be the inputs of the performance evaluator of the PSO module through a comparator. The fitness function is calculated by the performance evaluator. Then, obtained values of MSE will be used in the PSO algorithm. Based on these values, PSO optimizer will calculate and optimize the unknown parameters of covariance matrices Q and R . After that, we get the best set of particles by updating the particles solutions according to updating Eqs.24 and 25 as follows:

$$v_i(k+1) = w.v_i(k) + c_1.r_1(k).(p_{bi}(k) - x_i(k)) + c_2.r_2(k).(p_g(k) - x_i(k)) \quad (24)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (25)$$

where $v_i(k)$ and $x_i(k)$ are the current velocity and position of particle i at time k , respectively. $r_1(k)$ and $r_2(k)$ are two independent random sequences uniformly distributed between 0 and 1. The parameters c_1 and c_2 are the cognitive and the social accelerations coefficients, respectively, with positive values. w is the inertia weight factor. The value $p_{bi}(k)$ is the best local position for particle i , and $p_g(k)$ is the best global position both at time k .

Once the velocity for each particle is calculated, Eq.24 updates the velocity to the new one. The new position is then determined by the sum of the new velocity and the previous position by Eq.24.

The new position and updated matrices Q and R are then used to adapt the EKF for the next iteration until a predefined number of iterations have been reached, and then optimal matrices Q and R are obtained. Finally, optimized Q and R are injected into EKF observer for a future online running.

Note that the PSO-EKF algorithm is executed in an offline manner for the reason that PSO algorithm requires several iterations to achieve optimal solutions. For each iteration PSO-EKF algorithm must be executed once. Therefore, PSO-EKF algorithm should be executed several times allowing the optimization of the parameters Q and R , from each measurement.

3. Simulations and discussions

In order to verify the performance of proposed optimized observer-based control scheme, let us consider two degrees of freedom planar manipulator with revolute joints shown in Fig. 2.

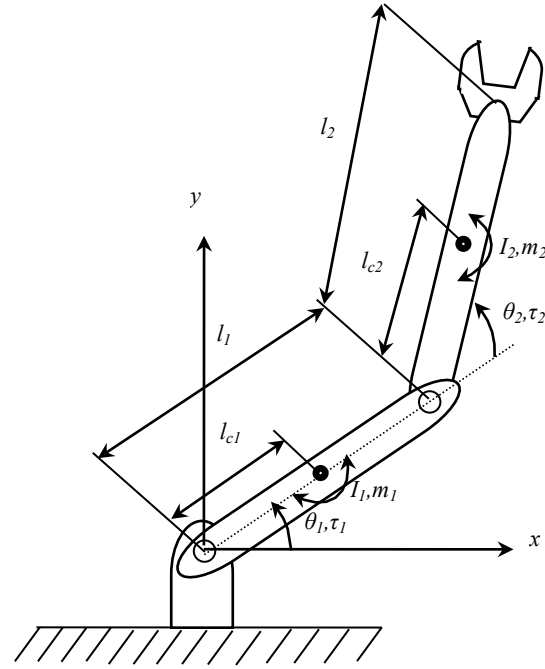


Fig. 2. Two-link robot manipulator

where l_i is the link length, m_i is the link mass, I_i is the link's moment of inertia given in center of mass, l_{ci} is the distance between the center of mass of link and the i^{th} joint.

The dynamic of the two-link robot manipulator can be described by the following differential equations [30]:

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad (26)$$

the matrix $M_0 = [m_{ij}]_{2 \times 2}$ is given by:

$$\begin{aligned} m_{11} &= m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2m_2 l_1 l_{c2} \cos(\theta_2)) + I_1 + I_2 \\ m_{12} &= m_2 (l_{c2}^2 + l_1 l_{c2} \cos(\theta_2)) + I_2 \\ m_{22} &= m_2 l_{c2}^2 + I_2 \end{aligned}$$

the matrix $C_0 = [c_{ij}]_{2 \times 2}$ is defined by:

$$c_{11} = a\dot{\theta}_2, \quad c_{12} = a\dot{\theta}_1 + a\dot{\theta}_2, \quad c_{21} = -a\dot{\theta}_1, \quad c_{22} = 0,$$

where $a = -m_2 l_1 l_{c2} \sin(\theta_2)$,

the vector $G_0 = [G_1, G_2]^T$ is given by:

$$\begin{aligned} G_1 &= (m_1 l_{c1} + m_2 l_1) g \cos(\theta_1) + m_2 l_{c2} g \cos(\theta_1 + \theta_2) \\ G_2 &= m_2 l_{c2} g \cos(\theta_1 + \theta_2) \end{aligned}$$

The vector $[d_1, d_2]^T$ represents the external load that the robot can take.

Due to modeling error (parameter variation and unknown load), it is assumed that the dynamic model of the manipulator (26) presents a certain uncertainty. Therefore, $M(\theta)$, $C(\theta, \dot{\theta})$ and $G(\theta)$ can be written as

$$M(\theta) = M_0(\theta) + \Delta M(\theta) \in R^{2 \times 2} \quad (27)$$

$$C(\theta, \dot{\theta}) = C_0(\theta, \dot{\theta}) + \Delta C(\theta, \dot{\theta}) \in R^{2 \times 2} \quad (28)$$

$$G(\theta) = G_0(\theta) + \Delta G(\theta) \in R^{2 \times 1} \quad (29)$$

where $M_0(\theta)$, $C_0(\theta, \dot{\theta})$ and $G_0(\theta)$ are nominal parts, whereas $\Delta M(\theta)$, $\Delta C(\theta, \dot{\theta})$ and $\Delta G(\theta)$ are the parameters uncertainties.

The dynamic model of the robotic manipulator (26) with parameters uncertainties and disturbance can be rewritten as following:

$$M_0(\theta)\ddot{\theta} + C_0(\theta, \dot{\theta})\dot{\theta} + G_0(\theta) + d(t) = \tau \quad (30)$$

where $d(t) = \Delta M(\theta) + \Delta C(\theta, \dot{\theta}) + \Delta G(\theta) + \delta(t) \in R^2$ represents the sum of the parametric uncertainties and external disturbances ($\delta(t)$).

To control the robot system, the state variable vector is chosen to be $X = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]^T = [x_1, x_2, x_3, x_4]^T$.

Choosing as output the position $y = [\theta_1, \theta_2]^T = [x_1, x_3]^T$, The dynamic equations are given as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(X) + g_1(X) \tau_1 + \delta_1 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(X) + g_2(X) \tau_2 + \delta_2 \\ y_1 = x_1 \\ y_2 = x_3 \end{cases} \quad (31)$$

where the dynamics $f(X)$ and $g(X)$ are given as follows

$$\begin{aligned} f &= \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = M_0^{-1} (-C_0[\dot{x}_1, \dot{x}_2] - G_0), \\ g &= \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = M_0^{-1} \end{aligned} \quad (32)$$

Since Kalman filter is a discrete algorithm, then discretization of the model is needed. This discretization will be done using the forward Euler method which provides an acceptable approximation of the systems dynamics for a short sampling period.

The resulting global discrete form will be given by the following discrete nonlinear representation:

$$\begin{cases} x_1(k+1) = x_1(k) + \Delta t \cdot x_2(k) + w_1(k) \\ x_2(k+1) = x_2(k) + \Delta t [f_1(X, k) + g_1(X, k) \tau_1(k) + \delta_1(k)] + w_2(k) \\ x_3(k+1) = x_3(k) + \Delta t \cdot x_4(k) + w_3(k) \\ x_4(k+1) = x_4(k) + \Delta t [f_2(X, k) + g_2(X, k) \tau_2(k) + \delta_2(k)] + w_4(k) \\ z_1(k) = x_1(k) + v_1(k) \\ z_2(k) = x_3(k) + v_2(k) \end{cases} \quad (33)$$

where Δt is the sampling period, k is the discrete-time point and $w(k) = [w_1(k) \ w_2(k) \ w_3(k) \ w_4(k)]$ and $v(k) = [v_1(k) \ v_2(k)]$ are the process and measurement white Gaussian noise vectors with zero mean and with associated covariance matrices

$Q = E[w_k, w_k^T]$ and $R = E[v_k, v_k^T]$, respectively.

In this simulation, the nominal parameters of the robot are given as

$$\begin{aligned} m_1 = m_2 &= 1 \text{ Kg}, & l_1 = l_2 &= 0.5 \text{ m}, & l_{c1} = l_{c2} &= 0.25 \text{ m}, \\ I_1 = I_2 &= 0.1 \text{ Kg.m}^2, & g &= 0.81 \text{ m/s}^2. \end{aligned}$$

In what follows, the proposed algorithm will be applied on the above two-link robot manipulator under PC simulation using Matlab software environment to show its efficiency. A total of $N = 4000$ measurement data are simulated on a time interval from 0 to 4 seconds with step size $\Delta t = 0.001$ s. Note that all codes are written in Matlab Language in M-files.

The desired reference trajectories for x_1 and x_3 are chosen to be $x_{1d}(t) = 70^\circ$ and $x_{3d}(t) = 90^\circ$. The initial values

of the system were selected as $x_1(0)=0$, $x_2(0)=0$, $x_3(0)=0$ and $x_4(0)=0$.

Three types of uncertainties are injected in the structure to verify the robustness of the controller: (1) Parameters uncertainties (+10% over the values of nominal model parameters)

(2) Random external disturbances which are chosen to be uniformly distributed as follows: $d_1 = d_2 = \text{rand}$ with $D=1$. Note that both disturbances sum to δ and they will be applied at $t > 2s$.

(3) Random Gaussian noises for the states and for the measurements both with zero mean values and with covariance's

$q = 10^{-2}$ and $r = 10^{-5}$, respectively.

EKF is implemented as in Eqs.19 to 22 where the Jacobean matrices are defined in Appendix 1. EKF will provide the state estimate vector $\hat{X} = [\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \hat{\theta}_4]^T = [\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4]^T$. The initial state and initial covariance conditions of the EKF are chosen to be $\hat{X}_{0/0} = [0, 0, 0, 0]^T$ and $P_{0/0} = \text{ones}(4, 4)$, respectively. In the simulation, error covariance matrix P is set to a 4×4

matrix. Q and R matrices have dimensions 4×4 and 2×2 , respectively, and are assumed having the following form:

For comparison purposes, the performance of EKF with diverse compositions of Q and R is evaluated by using the mean-squared-error Eq.23 of the position-estimating

response which is defined as: $MSE = \frac{1}{T} \sum_{k=1}^N [\theta_i(k) - \hat{\theta}_i(k)]^2$,

$$Q = \text{diag}(q_{x_1}, q_{x_2}, q_{x_3}, q_{x_4}) = \begin{bmatrix} q_{x_1} & 0 & 0 & 0 \\ 0 & q_{x_2} & 0 & 0 \\ 0 & 0 & q_{x_3} & 0 \\ 0 & 0 & 0 & q_{x_4} \end{bmatrix} \quad (34)$$

$$R = \text{diag}(r_1, r_2) = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \quad (35)$$

$i = 1, 2$

First, we simulate the system under the traditional feedback linearization control, in order to show its drawback in presence of parametric uncertainties and external disturbances. Applying the control law Eq.2, and after some trials, we chosen $k_{feed} = [200I_{2 \times 2}; 50I_{2 \times 2}]$ where $I_{i \times i}$ is an $i \times i$ identity matrix.

Table 1. EKF Performances for two-link robot using trial-error estimation

Case	Q and R entries						MSE_EKF	Estimation quality
	q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}	r_1	r_2		
1	1	1	1	1	1	1	1.6131	Poor
2	0.2	10^{-1}	0.2	10^{-1}	10^{-1}	1	3.3692×10^{-5}	Good
3	10^{-1}	10^{-1}	0.2	0.2	1	10^{-1}	3.2587×10^{-5}	Good
4	10^{-1}	10^{-1}	10^{-1}	10^{-1}	1	1	3.1821×10^{-6}	Very good

Table 1 shows typical EKF performance with their corresponding covariance matrices' (with entries $q_{q_1}, q_{q_1}, q_{q_2}, q_{q_2}, r_1$ and r_2) obtained by trial-error method. It is found that good estimation performance results when Q and R are equal (case 2 and 3 in Tab. 1), but a bad selection of ($q_{x_1}, q_{x_2}, q_{x_3}, q_{x_4}, r_1$ and r_2) can produce a poor estimation performance (case 1). Note that the best estimation performance is obtained with Q and R matrices ($q_{x_1} = q_{x_2} = q_{x_3} = q_{x_4} = 10^{-1}$ and $r_1 = r_2 = 1$) (case 4) which corresponds to the smallest MSE. Simulation results relative to the best case (case 4) are showed in Fig. 3 where we present in Fig. 3(a) and (b) respectively, the position of link-1 and position of link-2.

To compare the performance of EKF observer with other observers in terms of MSE, we give in Tab. 2 performances relative to sliding mode observer (SMO) [19]. The sliding gains of SMO were selected as in Tab. 2. By comparing

Tab. 1 and Tab. 2 (see MSE columns), we see clearly that both methods gave small MSE, but MSE obtained by EKF was smaller than that obtained with SMO observer.

Table 2. Performance of the SMO for two-link robot with trial-error estimation

Case	SMO gains		MSE_SMO	Estimation quality
	γ_1	γ_2		
1	10^{-5}	10^{-3}	1.3306	Poor
2	1	1	7.1961×10^{-5}	Good
3	2	10	4.3120×10^{-5}	Good
4	30	70	3.2612×10^{-6}	Very good

Note that the tracking performances are not very satisfactory especially after time $t = 2s$ when the perturbations were applied (+10% variation of parameters and external disturbance). As can be seen also, the prediction accuracy of EKF is not quite satisfactory due to the trial-error choice for EKF matrices.

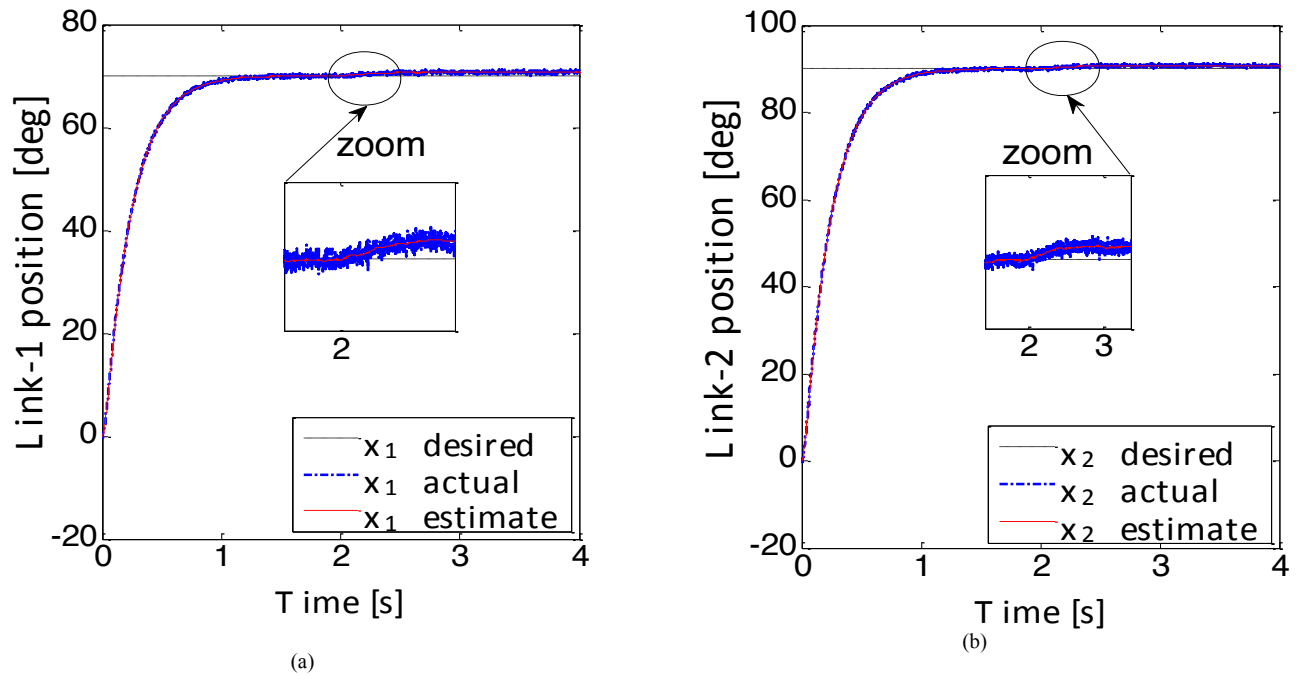


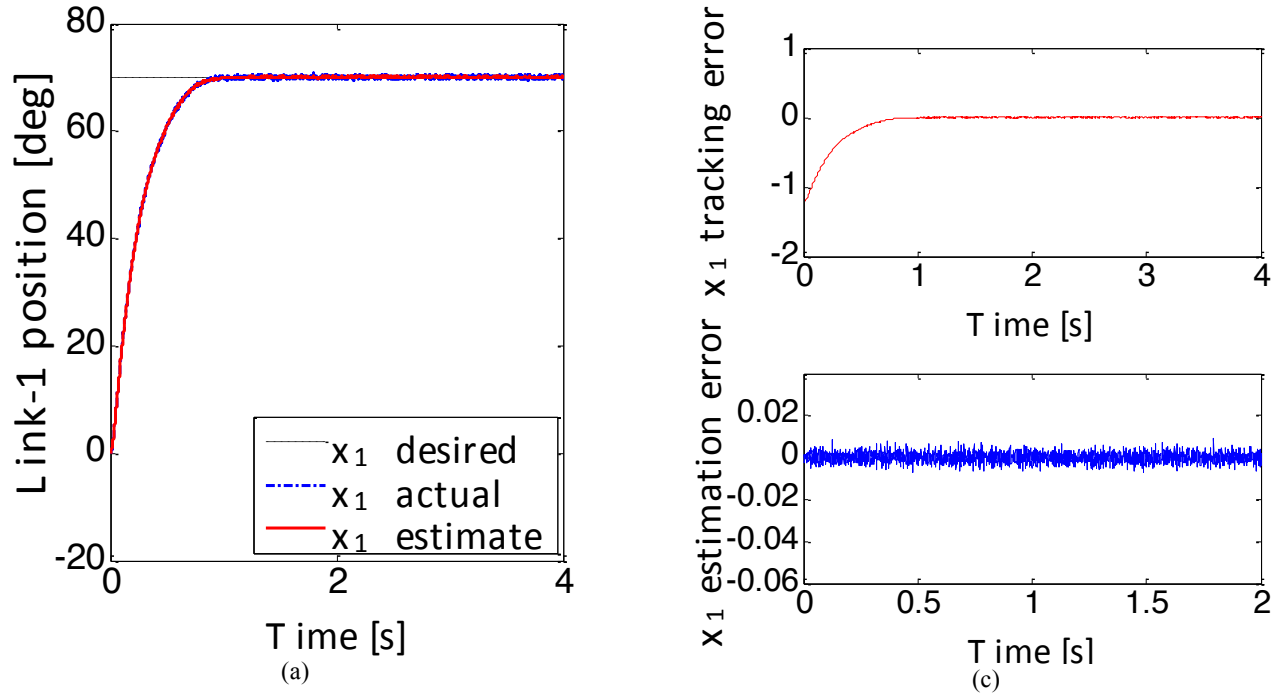
Fig. 3. Position of the joint angles using classical feedback linearization control with EKF.

In the rest of this section, the proposed method will be applied in order to resolve the above two problems.

3.1. Robustness problem

To solve the problem of robustness and acquire a better response to this system, control law given by Eq.4 is used in which the discontinuous control was added to classical

feedback linearization control. In this case and after some trials, we chosen $K=10I_{2 \times 2}$ and $\lambda=5I_{2 \times 2}$, where $I_{i \times i}$ is an $i \times i$ identity matrix. Simulation results are shown in Fig. 4.



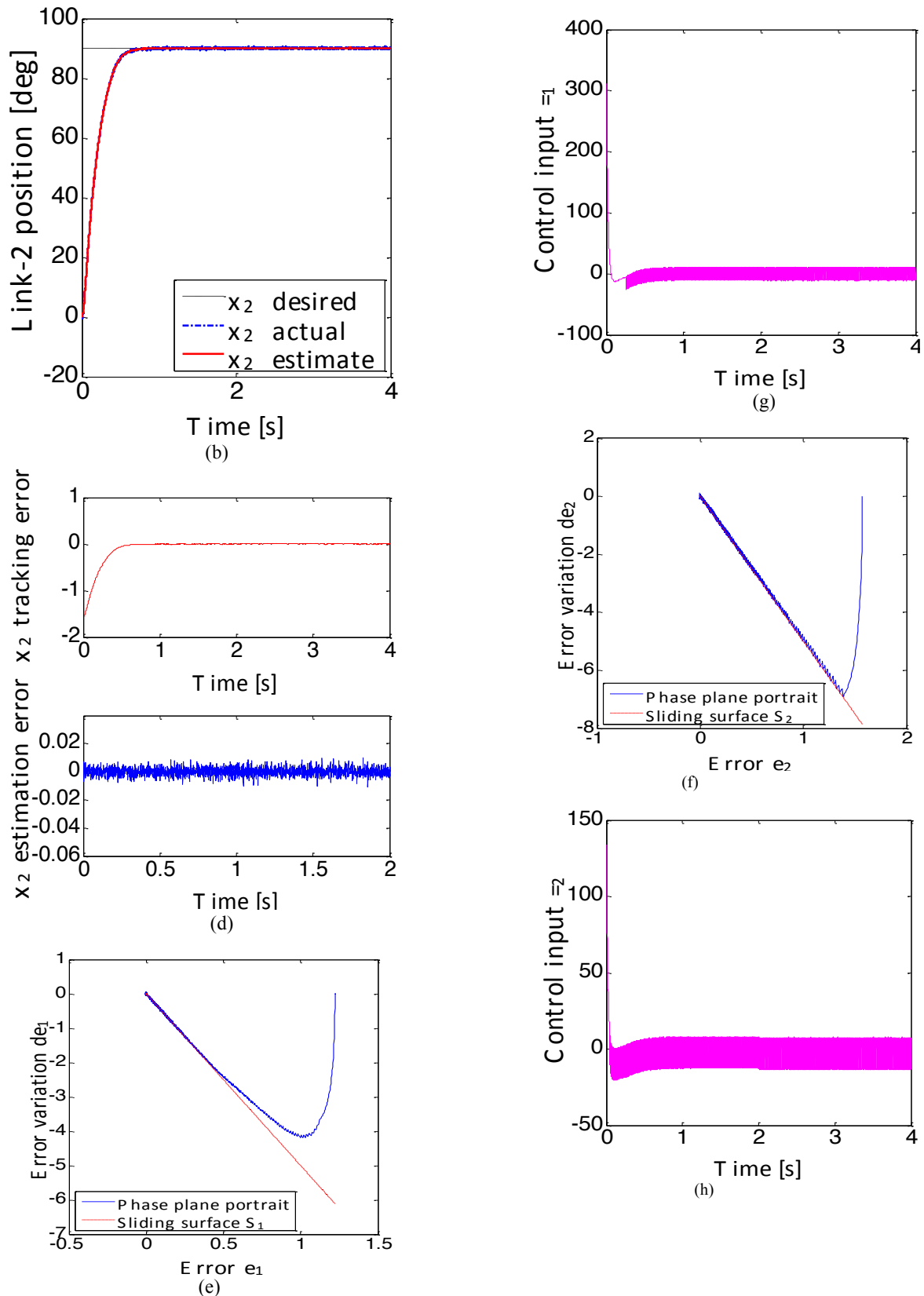


Fig. 4. Simulation results using enhancement of feedback linearization via discontinuous control (a) Position of the joint angle of link-1 (b) Position of the joint angle of link-2 (c) Link-1 position estimation and tracking errors (d) Link-2 position estimation and tracking errors (e) Phase plane portrait for the link-1 (f) Phase plane portrait for the link-2 (g) Control input applied to the link-1 (h) Control input applied to the link-2.

We present in Fig. 4(a) and (b) the position of the joint angles of link-1 and link-2 using the enhanced feedback linearization via discontinuous control. As we see the performances under the occurrence of parameter variations and external disturbance are satisfactory (see Fig. 4(c) and (d)). Contrary to classical feedback linearization alone

presented in Fig. 3(a) and (b), it appears in this case that the tracking performance of the joint angles of link-1 and link-2 are satisfactory especially after time $t=2$ when the perturbation arises. Fig. 4(e) and (f) represent the phase plane portrait of the robot, in which we can clearly see that the chattering phenomenon is appeared. Fig. 4(g) and (h)

show control inputs applied to the robot, where we note that the control performance is not satisfactory due to chattering phenomenon caused by the inappropriate selection of the switching gain.

In order to tackle this problem, the smoothing property of fuzzy logic is exploited as seen in section 2 to reduce the

chattering effect. The memberships functions of s and K_{fuzzy} are chosen as illustrated in Fig. 5(a) and (b) respectively, in which the following linguistic variables have been used: Negative (N), Zero (Z), Positive (P), Small (S) and Big (B).

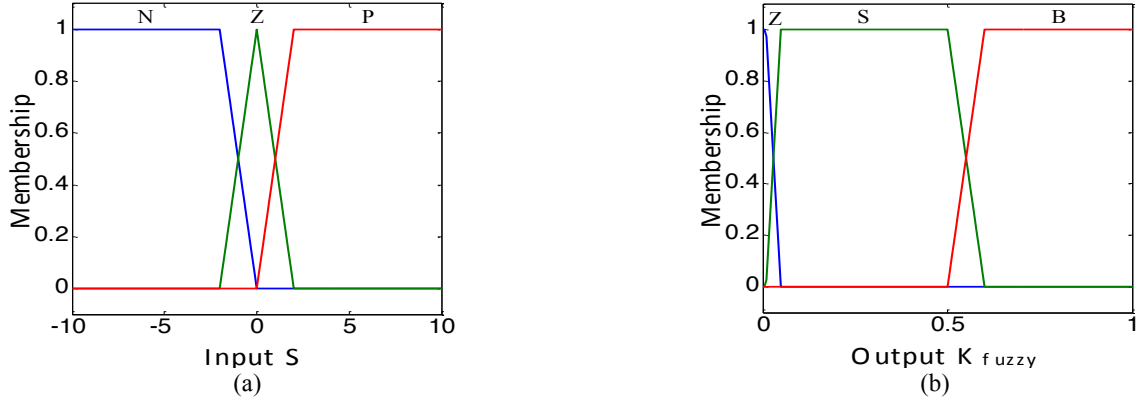


Fig. 5. (a) Input membership functions (b) Output membership functions.

The rule set of the adopted FLS contains 3 rules defined as following:

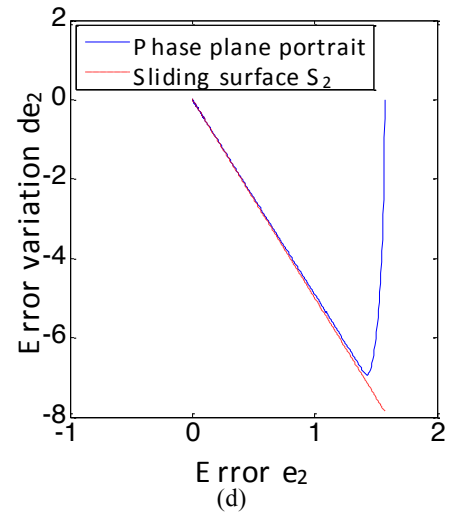
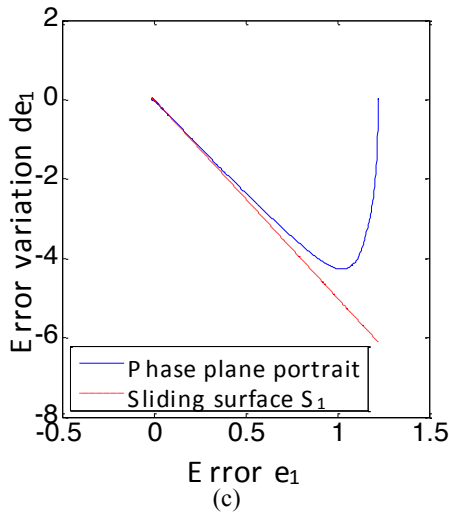
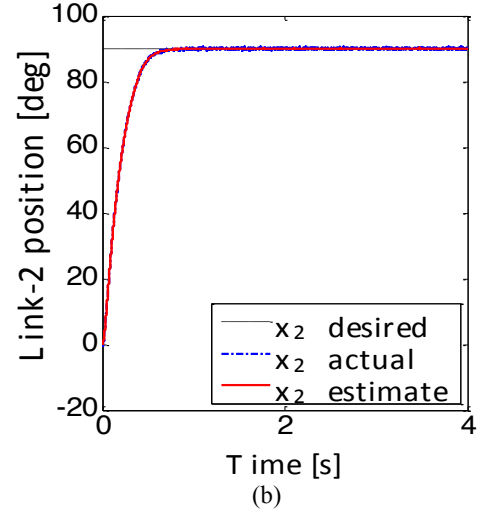
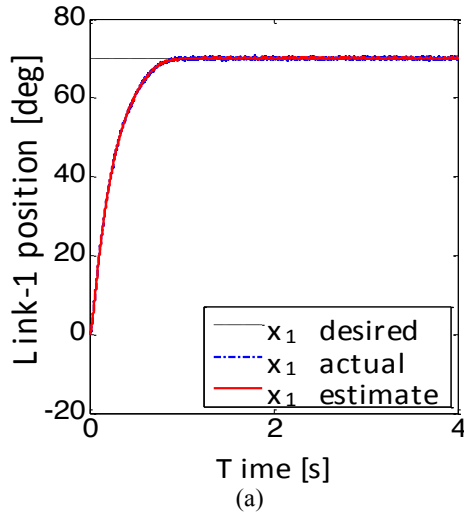
These rules govern the input-output relationship between s and K_{fuzzy} by adopting the Mamdani-type inference engine, in

Rule 1: If s is N , Then K_{fuzzy} is B

Rule 2: If s is Z , Then K_{fuzzy} is Z

Rule 3: If s is P , Then K_{fuzzy} is S

which the center of gravity method is used for defuzzification.



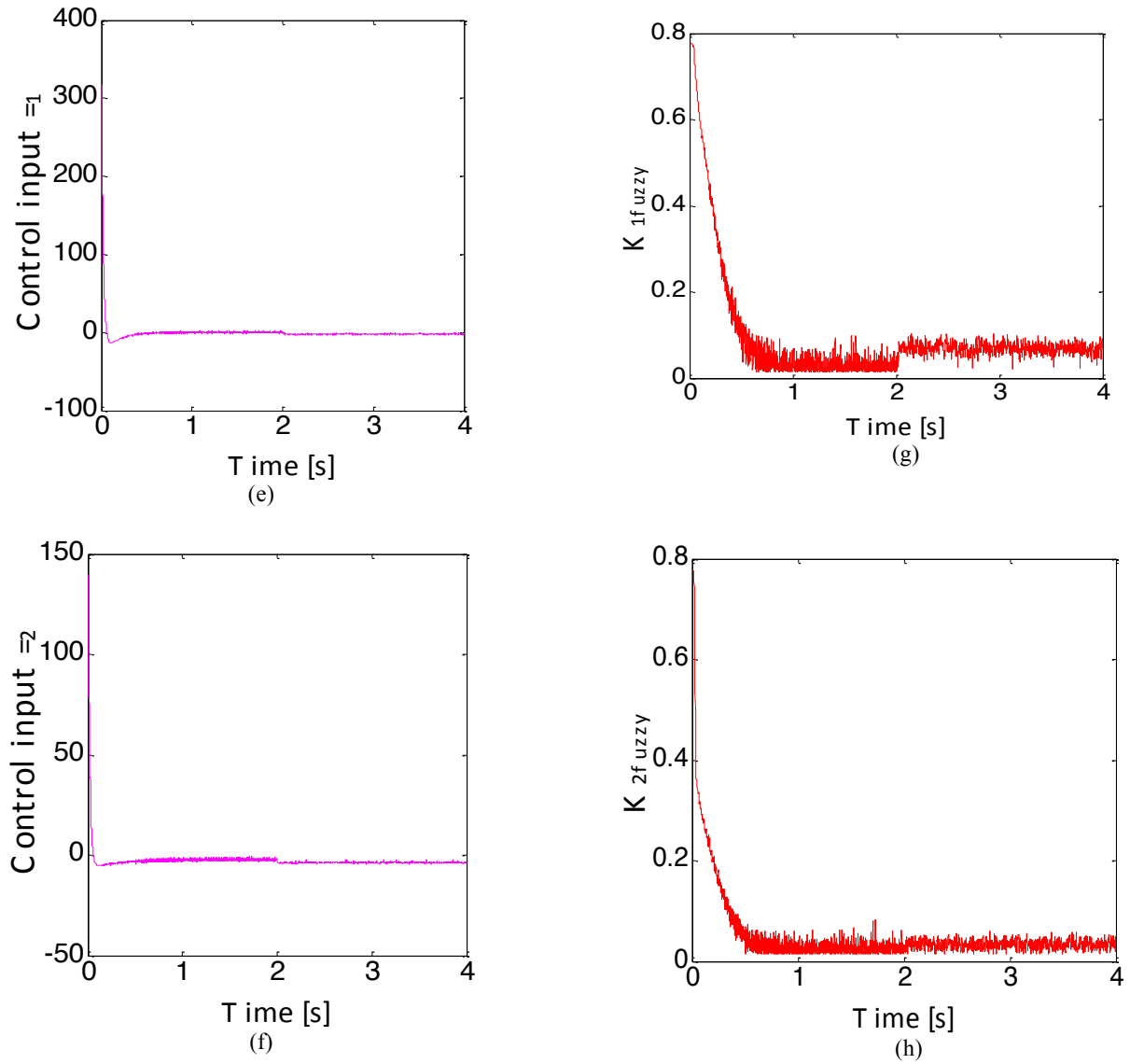


Fig. 6. Simulation results using enhancement of feedback linearization via discontinuous control and FLS (a) Position of the joint angle of link-1 (b) Position of the joint angle of link-2 (c) Phase plane portrait for the link-1 (d) Phase plane portrait for the link-2 (e) Control input applied to the link-1 (f) Control input applied to the link-2 (g) Evolution of fuzzy gain K_{1fuzzy} (h) Evolution of fuzzy gain K_{2fuzzy} .

We show in Fig. 6 the simulation results corresponding to improvement of switching gain, where we present in Fig. 6(a) and (b) respectively, the position of the joint angles. From comparing the new obtained Phase plane portrait in Fig. 6(c) and (d), with the old one Fig. 4(e) and (f), we can clearly see that the chattering phenomenon is almost disappeared. And comparing the associated control inputs presented in Fig. 6(e) and (f), with the old one Fig. 4(g) and (h), it is noted that the discontinuities amplitudes are reduced. The estimated fuzzy gains are depicted in Fig. 6(g) and (h).

3.2. Prediction problem

Note that in all above simulations, the EKF covariance matrices were adjusted by using the trial-error method which is simple to achieve but takes a very longtime. To get more satisfactory performance, the adjustment will be done automatically by PSO algorithm discussed above in section 2.4.

3.2.1. PSO-EKF method

We suggest searching simultaneously the optimal combination of six variances $q_{x_1}, q_{x_2}, q_{x_3}, q_{x_4}, r_1$ and r_2 using Eqs.32 and 33 to find the optimal covariance matrices Q and R of the EKF, which will allow to obtain better estimates with higher precision than the trial-error method.

After running the PSO-EKF, the optimized covariance matrices Q and R and their corresponding performances MSEs are given in Tab. 3. It should be noted that the convergence of the PSO method to the optimal solution depends on the parameters c_1 , c_2 and w in which self-recognition coefficient $c_1 = 1.5$, social coefficient $c_2 = 2$ and Inertia weight varies between $w = 0.3$ to 1 . Since we have six parameters to be optimized, therefore the dimension of the simulation will be 6. Note also that the simulated swarm has a size of 20 with a maximal number of generations equal to 100.

Table 3. Optimized EKF Performances using PSO

Swarm size	Iterations	Q and R entries						MSE_PSO
		q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}	r_1	r_2	
20	5	10^{-4}	0.0316	10^{-4}	0.0534	0.0685	0.0745	1.5401×10^{-6}
	10	10^{-4}	0.0432	10^{-4}	0.0416	0.0515	0.0621	1.3712×10^{-6}
	50	10^{-4}	0.0005	10^{-3}	0.0001	0.0400	0.0601	1.2049×10^{-6}
	100	10^{-5}	10^{-5}	10^{-5}	10^{-5}	0.4800	0.8637	1.1637×10^{-6}

Tab. 3 illustrates the convergence of PSO-EKF algorithm, where the MSE is decreased to 1.1637×10^{-6} after 100 iterations, which is less than the value obtained by trial-error method ($MSE_{trial-error} = 3.1821 \times 10^{-6}$) which confirms the effectiveness of this method.

3.2.2. Genetic algorithms method

For comparison purposes, we will present in what follows EKF optimization by using GAs. Note that we used genetic

algorithms with the following parameters: *Population size*=20, *Maximal number of generation*=100, *Dimension*=6, *Crossover probability*=0.8 and *Mutation probability*=0.01.

Optimized covariance matrices using GA algorithms are given in Tab. 4 where we see that the MSE is decreased to 2.4959×10^{-6} after 100 iterations. Note that this MSE is close to the trial-error MSE which is equal to 3.1821×10^{-6} .

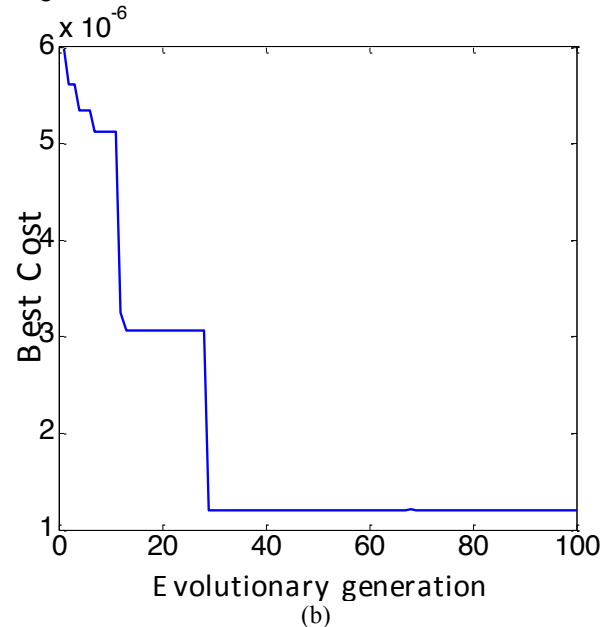
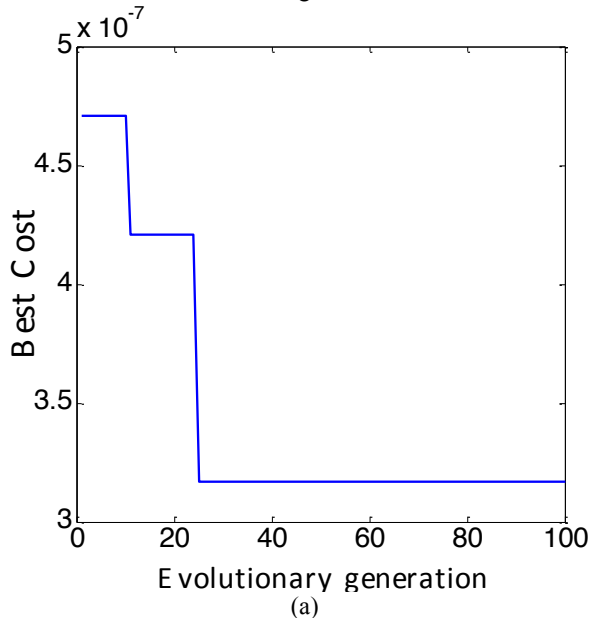
Table 4. Optimized EKF Performances using GA

Swarm size	Iterations	Q and R entries						MSE_GAs
		q_{x_1}	q_{x_2}	q_{x_3}	q_{x_4}	r_1	r_2	
20	5	0.0153	0.0416	10^{-4}	0.0456	0.0772	0.0753	7.4531×10^{-6}
	10	0.0106	0.0112	10^{-4}	0.0324	0.0456	0.0568	6.2235×10^{-6}
	50	0.0081	0.0153	0.001	0.0248	0.0440	0.0654	2.7500×10^{-6}
	100	0.0010	0.1000	10^{-4}	0.0010	0.5000	0.4001	2.4959×10^{-6}

From the obtained results showed in Tables 3 and 4, comparison of PSO-EKF and GAs approaches shows that both methods are able to find the optimum design covariance matrices Q and R . It can be easily seen that PSO-EKF gives more precise results than GAs approach when the number of iteration (generation) increases. Therefore, it can be confirmed that PSO-EKF can give better estimates than GAs

approach. Note that the comparison was done under the same conditions (generation number, population size, initial population).

Furthermore, Fig. 7 shows the evolution of the fitness function for PSO and GAs methods, respectively; where we notice that the convergence of PSO is faster than the convergence of GAs.


Fig. 7. Evolution of fitness/objective function versus 100 iterations. (a) PSO relative to Tab. 3, (b) GA relative to Tab. 4.

In what follows we will present the final simulation results relative to the improved feedback linearization control with the best optimal values of EKF parameters (see

Fig.8)

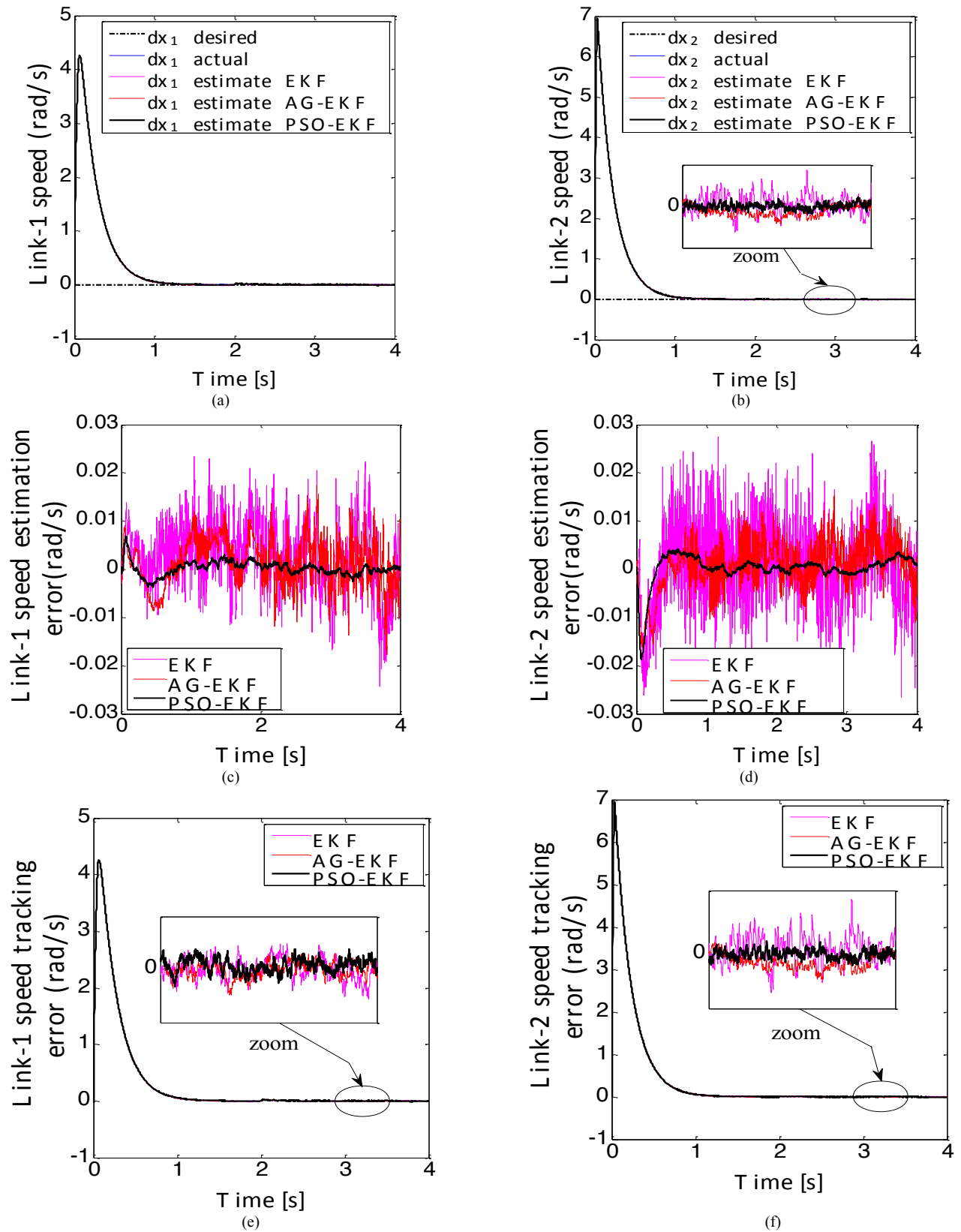


Fig. 8. Simulation results using proposed control for different optimization algorithms (a) Speed of the joint angle of link-1, (b) Speed of the joint angle of link-2, (c) Speed estimation errors for the link-1, (d) Speed of the joint angle of link-2, (e) Speed tracking errors for the link-1, (f) Speed tracking errors for the link-2.

In Fig. 8(a) and (b), we present respectively the Link-1 and Link-2 speed responses with the optimal values of EKF covariance matrices given in Tables 1, 3 and 4 for trial-error, PSO and GAs optimizations. The corresponding speed estimation errors are presented in Fig. 8(c) and (d), respectively. Also, the corresponding speed tracking errors

are presented in Fig. 8(e) and (f), respectively.

In all these figures, we see that best results are obtained with the proposed PSO-EKF method where it can be seen that PSO-EKF fits the true state variables with higher accuracy for two-link robot.

4. Conclusions

In this paper, we have proposed an improved and optimized feedback linearization controller. The approach combines classical feedback linearization, discontinuous control and fuzzy systems. The introduced improvement on classical feedback linearization was guaranteed by a discontinuous control action which itself was also enhanced by a fuzzy system. We assumed that not all states are measured; therefore an EKF system to observe the hidden states was introduced. The performances of EKF has been efficiently improved by adjusting the covariance matrices Q and R via PSO technique, and the obtained results was validated by a short comparative study with GA. The effectiveness of EKF

observer is also validated by comparing it with SMO. The stability of the proposed approach was guaranteed by Lyapunov stability criterion. Simulation results confirmed the ability of proposed approach to ensure an acceptable robustness and yields superior control performances for nonlinear system control against uncertainties and external disturbance simultaneously.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence



References

1. M.W.Spong, M.Vidyasagar, Robot Dynamics and Control. Wiley-India Edition, New York, (2009).
2. M.H.Khooban, Design an intelligent proportional derivative pd feedback linearization control for nonholonomic-wheeled mobile robot. Journal of Intelligent & Fuzzy Systems **26**(4), 1833–1843 (2014). DOI: 10.3233/IFS-130863.
3. H.A.Abbas, B.Belkheiri, B.Zegnini, Feedback Linearization Control for Highly Uncertain Nonlinear Systems Augmented by Single-Hidden-Layer Neural Networks. Journal of Engineering Science and Technology Review, **8**(2) (2015).
4. S.Şahin, Learning feedback linearization using artificial neural networks. Springer, Int. J. Neural Process. Lett. **44**(3), 625–637 (2016). DOI: 10.1007/s11063-015-9484-8.
5. A.Isidori, Nonlinear Control Systems - An Introduction. Springer-Verlag, (1989).
6. A.L.D.Franco, et al., Robust nonlinear control associating robust feedback linearization and h_∞ control. IEEE Transactions on Automatic Control **51**(7), 1200–1207 (2006).
7. C-J.Huang, J.Ohri, Fuzzy feedback linearization control for mimo nonlinear system and its application to full-vehicle suspension system. Springer, J. Circ. Syst. Sig. Proc.; **28**, 959–991 (2009).
8. L.Wang, The feedback linearization based on backstepping technique. IEEE International Conference on Intelligent Computing and Intelligent Systems, pp. 282–286 (2009).
9. J.O.Pedro and O.A.Dahunsu, Neural based feedback linearization control of a servo-hydraulic vehicle suspension system. Int. J. Appl. Math. Comput. Sci. **21**(1), 137–147 (2011).
10. C-C.Fuh, H-H.Tsai and W-H.Yao, Combining a feedback linearization controller with a disturbance observer to control a chaotic system under external excitation. Elsevier, Communications in Nonlinear Science and Numerical Simulation, pp. 1423–1429 (2012).
11. F.Piltan, et al., Design novel nonlinear controller applied to robot manipulator: Design new feedback linearization fuzzy controller with minimum rule base tuning method. Int. J. Rob. Auto. (IJRA) **3**, 1–12 (2012).
12. M.C.Tanaka, J.M.M.Fernandes and W.M.Bessa, Feedback linearization with fuzzy compensation for uncertain nonlinear systems. Int. J. Comput. Commun. **8**(5), 736–743 (2013). ISSN 1841-9836.
13. R.E.Kalman, A new approach to linear filtering and prediction problems. Transactions ASME, Journal Basic Engineering **82**, 35–45 (1960).
14. D.Luenberger, Observing the state of linear system. IEEE Trans. Military Electron. **8**, 74–90 (1964).
15. T.O.Kowalska, Application of extended Luenberger observer for flux and rotor time-constant estimation in induction motor drives. IEE Proc. Control Theory **36**, 324–330 (1989).
16. T.Du, et al., Application of Kalman Filters and Extended Luenberger Observers to Induction Motor Drives. Proc. Intelligent Motion conference, pp. 369–386 (1994).
17. K.L.Shi, et al., Speed estimation of an induction motor drive using an optimized extended Kalman filter. IEEE Transactions on Industrial Electronics **49**, 124–133 (2002).
18. M.Barut, S.Bogosyan and M.Gokasan, Speed-sensorless estimation for induction motors using extended Kalman filters. IEEE Trans. Ind. Electron. **54**, 272–280 (2007).
19. Y.Xiong and M.Saif, Sliding mode observer for nonlinear uncertain systems. IEEE Trans. on Automatic Control **46**(12), 2012–2017 (2001). DOI: 10.1109/9.975511
20. Y.Agrebi, et al., Rotor speed estimation for indirect stator flux oriented induction motor drive based on MRAS scheme. Journal. Electr. Syst. **3**, 131–143 (2007).
21. B.Karanayil, M.F.Rahman and C.Grantham, Online stator and rotor resistance estimation scheme using artificial neural networks for vector controlled speed sensorless induction motor drive. IEEE Trans. Indus. Electron. **54**, 167–176 (2007).
22. R.Shahnazi, Mohammad-R.Akbarzadeh-T, PI Adaptive Fuzzy Control With Large and Fast Disturbance Rejection for a Class of Uncertain Nonlinear Systems. IEEE Trans. on Fuzzy Systems **16**(1), 187–197 (2008).
23. X-J.Ma, Z.Sun and Y-Y.He, Analysis and design of fuzzy controller and fuzzy observer. IEEE 12th symposium on computational intelligence and informatics (CINTI), Budapest, Hungary, p p. 45–50 (2011).
24. C.Manes, F.Parasiliti and M.Tursin, A comparative study of rotor flux estimation in induction motors with a nonlinear observer and the extended Kalman filter. Proc. IEEE IECON'94, pp. 2149–2154 (1994).
25. P.Vas, Sensorless Vector and Direct Torque Control. Oxford University Press. London: ISBN 978-0198564652, 0198564651, (1998).
26. S.Bolognani, R.Oboe and M.Ziglotto, Sensorless full-digital PMSM drive with EKF estimation of speed and rotor position. IEEE Trans. Ind. Electron. **46**, 184–191 (1999).
27. K.L.Shi, et al., Speed estimation of an induction motor drive using an optimized extended Kalman filter. IEEE Trans. on Industrial Electronics **49**(1), 124–133 (2002).
28. Y.Laamari, K.Chafaa, B.Athamena, Particle swarm optimization of an extended Kalman filter for speed and rotor flux estimation of an induction motor drive. Electrical Engineering, Archiv für Elektrotechnik, Springer-Verlag Berlin Heidelberg **97**, 129–138 (2014).
29. J-JE.Slotine, W.Li, Applied Nonlinear Control. London: Prentice-Hall, Inc. ISBN: 0130408905, (1991).
30. A. Tayebi, Adaptive iterative learning control for robot manipulators, Automatica, Elsevier **40** (7) (2004) 1195–1203.

ppendix 1: Jacobian matrices for the two-link robot

$$F_k = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{bmatrix}, \quad W_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } V_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

where $f_{11}=1$, $f_{12}=\Delta t$, $f_{13}=0$, $f_{14}=0$

$$f_{21} = \Delta t(((m_2 l_{c2}^2 + I_2)(g S_1(l_1 m_2 + l_{c1} m_1) + g l_{c2} m_2 S_{12}))/((I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 \times l_{c2}^2 m_2^2 C_2^2)) - (g l_{c2} m_2 S_{12}(m_2 l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2))/((I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 \times l_{c2}^2 m_2^2 C_2^2))).$$

$$f_{22} = \Delta t[(2 l_1 l_{c2} m_2 \dot{q}_2 S_2(m_2 l_{c2}^2 + I_2))/((I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)) + (2 l_1 \times l_{c2} m_2 \dot{q}_1 S_2(m_2 l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2))/((I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2))] + 1.$$

$$f_{23} = (\Delta t l_1 l_{c2} m_2 \tau_2 S_2) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2) - \Delta t(((-l_1 l_{c2} m_2 \times C_2 \dot{q}_1^2 + g l_{c2} m_2 S_{12})(m_2 l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)) - ((m_2 l_{c2}^2 + I_2)(\dot{q}_2(\dot{q}_1 l_1 l_{c2} m_2 C_2 + \dot{q}_2 l_1 l_{c2} m_2 C_2) + g l_{c2} m_2 S_{12} + \dot{q}_1 \dot{q}_2 l_1 l_{c2} m_2 C_2))/((I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 \times m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)) + (l_1 l_{c2} m_2 S_2(l_1 l_{c2} m_2 S_2 x_2^2 + g l_{c2} m_2 C_{12}))/((I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)) + (2 l_1^2 l_{c2}^2 m_2^2 C_2 S_2(l_1 l_{c2} m_2 S_2 \dot{q}_1^2 + g l_{c2} m_2 C_{12}))(m_2 l_{c2}^2 + l_1 m_2 C_2 \times l_{c2} + I_2)) / ((-l_1^2 l_{c2}^2 m_2^2 C_2^2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + m_1 l_{c1}^2 l_{c2}^2 m_2 + I_2 m_1 l_{c1}^2 + I_1 l_{c2}^2 m_2 + I_1 I_2)^2 + (2 l_1^2 l_{c2}^2 m_2^2 C_2 S_2(m_2 \times l_{c2}^2 I_2)(\dot{q}_2(\dot{q}_1 l_1 l_{c2} m_2 S_2 + \dot{q}_2 l_1 l_{c2} m_2 S_2) - g C_1(l_1 m_2 l_{c1} + m_1) - g l_{c2} m_2 C_{12} + \dot{q}_1 \dot{q}_2 l_1 l_{c2} m_2 S_2)) / ((-l_1^2 l_{c2}^2 m_2^2 C_2^2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + m_1 l_{c1}^2 l_{c2}^2 m_2 + I_2 m_1 l_{c1}^2 + I_1 l_{c2}^2 m_2 + I_1 I_2)^2) - (2 \Delta t l_1^2 l_{c2}^2 m_2^2 \tau_1 C_2 S_2(m_2 l_{c2}^2 + I_2)) / ((-l_1^2 l_{c2}^2 m_2^2 C_2^2 + l_1^2 l_{c2}^2 \times m_2^2 + I_2 l_1^2 m_2 + m_1 l_{c1}^2 l_{c2}^2 m_2 + I_2 m_1 l_{c1}^2 + I_1 l_{c2}^2 m_2 + I_1 I_2)^2 + (2 \Delta t l_1^2 l_{c2}^2 m_2^2 \tau_2 C_2 S_2(m_2 l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2)) / ((-l_1^2 l_{c2}^2 \times m_2^2 C_2^2 + l_1^2 l_{c2}^2 m_2^2 + I_2 \times l_1^2 \times m_2 + m_1 l_{c1}^2 l_{c2}^2 m_2 + I_2 m_1 l_{c1}^2 + I_1 l_{c2}^2 m_2 + I_1 I_2)^2).$$

$$f_{24} = (\Delta t(m_2 l_{c2}^2 + I_2)(2 l_1 l_{c2} m_2 \dot{q}_1 S_2 + 2 l_1 l_{c2} m_2 \dot{q}_2 S_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2).$$

$$f_{31} = 0, \quad f_{32} = 0, \quad f_{33} = 1, \quad f_{34} = \Delta t$$

$$f_{41} = -\Delta t(((g S_1(l_1 m_2 + l_{c1} m_1) + g l_{c2} m_2 S_{12})(m_2 l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 \times m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)) - (g l_{c2} m_2 S_{12}(m_2 l_{c2}^2 + 2 m_2 C_2 l_1 l_{c2} + m_1 l_{c1}^2 + m_2 l_{c2}^2 + I_1 + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2))).$$

$$f_{42} = -\Delta t((2 \dot{q}_2 l_1 l_{c2} m_2 S_2(m_2 l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)) + (2 \dot{q}_1 l_1 l_{c2} m_2 S_2(m_2 l_{c2}^2 + 2 m_2 C_2 l_1 l_{c2} + m_1 l_{c1}^2 + m_2 l_{c2}^2 + I_1 + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 \times l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 \times l_{c2}^2 m_2^2 C_2^2))).$$

$$f_{43} = \Delta t(((g l_{c2} m_2 S_{12} - l_{c2} m_2 C_2 \dot{q}_1^2)(m_2 l_1^2 + 2 m_2 C_2 l_{c2} + m_1 l_{c1}^2 + m_2 l_{c2}^2 + I_1 + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 \times m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 c_2^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2) - ((\dot{q}_2 (\dot{q}_1 l_{c2} m_2 C_2 + \dot{q}_2 l_{c2} m_2 C_2) + g l_{c2} m_2 S_{12} + \dot{q}_1 \dot{q}_2 l_{c2} m_2 C_2) (m_2 \times l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2) + (l_{c2} m_2 S_2 \times (\dot{q}_2 (\dot{q}_1 l_{c2} m_2 S_2 + \dot{q}_2 l_{c2} m_2 S_2) - g C_1 (l_1 m_2 + l_{c1} m_1) - g l_{c2} m_2 C_{12} + \dot{q}_1 \dot{q}_2 l_{c2} m_2 S_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2) + (2 l_{c2} m_2 S_2 (l_{c2} m_2 S_2 \dot{q}_1^2 + g l_{c2} m_2 C_{12})) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2) + (2 l_1^2 l_{c2}^2 m_2^2 C_2 S_2 (m_2 l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2) (\dot{q}_2 (\dot{q}_1 l_{c2} m_2 \times S_2 + \dot{q}_2 l_{c2} m_2 S_2) - g C_1 (l_1 m_2 + l_{c1} m_1) - g l_{c2} m_2 C_{12} + \dot{q}_1 \dot{q}_2 l_{c2} m_2 S_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 \times m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)^2 + (2 l_1^2 l_{c2}^2 m_2^2 C_2 S_2 (l_{c2} m_2 S_2 \dot{q}_1^2 + g l_{c2} m_2 C_{12})) (m_2 l_1^2 + 2 m_2 C_2 l_{c2} + m_1 l_{c1}^2 + m_2 l_{c2}^2 + I_1 + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)^2) + (\Delta t l_{c2} m_2 \tau_1 S_2) / (I_1 \times I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2) - (2 \Delta t l_{c2} m_2 \tau_2 S_2) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 \times l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2) - (2 \Delta t l_1^2 l_{c2}^2 m_2^2 \tau_2 C_2 S_2 (m_2 l_1^2 + 2 m_2 C_2 l_{c2} + m_1 l_{c1}^2 + m_2 l_{c2}^2 + I_1 + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)^2 + (2 \Delta t l_1^2 l_{c2}^2 m_2^2 \tau_1 C_2 S_2 (m_2 \times l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)^2).$$

$$f_{44} = 1 - (\Delta t (2 \dot{q}_1 l_{c2} m_2 S_2 + 2 \dot{q}_2 l_{c2} m_2 S_2) (m_2 l_{c2}^2 + l_1 m_2 C_2 l_{c2} + I_2)) / (I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 \times m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)$$

$$S_{12} = \sin(q_1 + q_2), \quad S_1 = \sin(q_1), \quad S_2 = \sin(q_2), \quad C_{12} = \cos(q_1 + q_2), \quad C_1 = \cos(q_1), \quad C_2 = \cos(q_2).$$