

BBO optimization of an EKF for interval type-2 fuzzy sliding mode control

Ali Medjghou¹, Mouna Ghanai¹, and Kheireddine Chafaa¹

¹ LAAAS, Electronics Department, University Batna 2 Mostefa BenBoulaid,
53, Constantine Road, Fesdis, Batna 05078, Algeria

E-mail: a.medjghou@univ-batna2.dz;

m.ghanai@univ-batna2.dz; k.chafaa@univ-batna2.dz

Received 8 August 2017

Accepted 27 February 2018

Abstract

In this study, an optimized extended Kalman filter (EKF), and an interval type-2 fuzzy sliding mode control (IT2FSMC) in presence of uncertainties and disturbances are presented for robotic manipulators. The main contribution is the proposal of a novel application of Biogeography-Based Optimization (BBO) to optimize the EKF in order to achieve high performance estimation of states. The parameters to be optimized are the covariance matrices Q and R , which play an important role in the performances of EKF. The interval type-2 fuzzy logic system is used to avoid chattering phenomenon in the sliding mode control (SMC). Lyapunov theorem is used to prove the stability of control system. The suggested control approach is demonstrated using a computer simulation of two-link manipulator. Finally, simulations results show the robustness and effectiveness of the proposed scheme, and exhibit a more superior performance than its conventional counterpart.

Keywords: Biogeography-Based Optimization; Interval Type-2 Fuzzy Logic System; Sliding Mode Control; Extended Kalman Filter; Two-Link Manipulator.

1. Introduction

One of the best classical nonlinear controllers is the sliding mode controller (SMC), which is based on the theory of variable structure system; it was first introduced by Utkin in 1977. The SMC is known by its robustness against uncertainties and external disturbances. During the last two decades, fuzzy logic systems (FLS) have been a dominant topic in intelligent control systems. Many FLS schemes have been developed for handling nonlinear systems to improve the performance of SMC, e.g., Kapoor and Ohri¹, proposed fuzzy sliding mode controller with global stabilization using a saturation function for the trajectory control; they used fuzzy logic systems for tuning the switching gain of the SMC and linear saturation boundary layer function for solving

the problem of chattering phenomenon (high frequency vibrations). Soltanpour *et al.*², combined the feedback linearization control with the fuzzy sliding mode controller using Takagi-Sugeno fuzzy model, and the obtained result was free of undesirable chattering phenomenon. Morover, Baklouti *et al.*³, combined the SMC, proportional integral (PI) controller and adaptive fuzzy systems for a robot manipulator; fuzzy system is used to approximate the unknown nonlinear functions and PI action is used to reduce the chattering phenomenon. However, Naoual *et al.*⁴, designed a fuzzy sliding mode controller for a two-link manipulator, in which fuzzy logic is used to approximate only the unknown dynamic parts of the system. Chen *et al.*⁵, combined the SMC and function approximation techniques for the DC motor control system, the function approximation tech-

nique is used to transform the uncertain term into finite linear combinations of orthogonal basis function. Note that in all this cited works the observation problem is not considered. Moreover, Medjghou *et al.*⁶, designed a type-1 fuzzy sliding mode controller based on an EKF for a two-link manipulator, in which fuzzy logic is used to approximate the switching gain of the SMC.

Type-2 fuzzy logic is a generalization of type-1 fuzzy logic (conventional fuzzy logic), in which the values of membership functions are themselves^{7,8}. The most commonly used type of Type-2 fuzzy logic system is the interval type-2 fuzzy logic system (IT2FLS), which uses interval membership degrees⁹. Controllers based on IT2FLS are able to maintain performance in the presence of high level noise and nonlinearity^{9,10,11}. Consequently, by integrating IT2FLS in conventional SMC, a hybrid intelligent tracking controller with robustness against measurement noise can be achieved. Although the SMC performs well in the nonlinear systems, it has a major drawback, the so-called chattering phenomenon, which is caused by inappropriate selection of the switching gain. However, several researches were devoted to avoid this problem^{12,13,14,15}.

The robust nonlinear control of a given system requires the knowledge of state variables, which are rarely available for direct measurement. In most cases, there is a real need for reliably estimated 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 that is capable to deliver estimated states 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 Kalman¹⁶ and Luenberger¹⁷, and functional observer¹⁸. Contrarily, in nonlinear systems, there is not a general solution to the problem of observer synthesis, which prompted researchers to develop nonlinear observers. On this subject, several algorithms exist in literature, namely extended Luenberger observer^{19,20}, extended Kalman filter^{21,22},

sliding mode observer (SMO)^{23,24}, model reference adaptive system²⁵, neural network observer²⁶ and fuzzy logic observer^{27,28}.

Amongst all these algorithms, EKF provides the suboptimal state estimator for 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 with other nonlinear observers^{29,30,31,32}, 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. Following³³, 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 consumption³⁴. To overcome this problem and to avoid the computational complexity of trial-error method, Shi *et al.*³⁵, have used genetic algorithms (AGs), downhill simplex algorithm³⁶, and particle swarm optimization (PSO)³⁷, which were used to optimize matrices Q and R .

This paper presents an interval type-2 fuzzy sliding mode control combined with an optimized EKF observer in the presence of parametric uncertainties and external disturbances for robotic manipulators. The main contribution is the proposal of a novel application of BBO approach introduced by Simon³⁸, which is an evolutionary algorithm inspired by mathematical models of biogeography to optimize the parameters of EKF in order to achieve high performance estimation of states, and it is then compared to PSO technique. The parameters to be optimized are the covariance matrices entries Q and R , which play an important role in the performances of EKF observer. SMC is used for trajectory tracking, then an IT2FLS is used to adaptively tune the switching gain of SMC for smoothing purposes to

reduce the chattering phenomenon.

The rest of this paper is organized as follows. In Section 2, some basics on the extended Kalman filter. Then, the principle of biogeography-based optimization approach is given in Section 3. In Section 4, the problem description and formulation is presented and the detail of proposed scheme is display. To validate the robustness and performance of the proposed method, simulation results and their discussion are presented in Section 5. Finally, conclusions are given in Section 6.

2. Extended Kalman Filter

The Kalman filter was developed by R.E. Kalman¹⁵. The EKF 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.

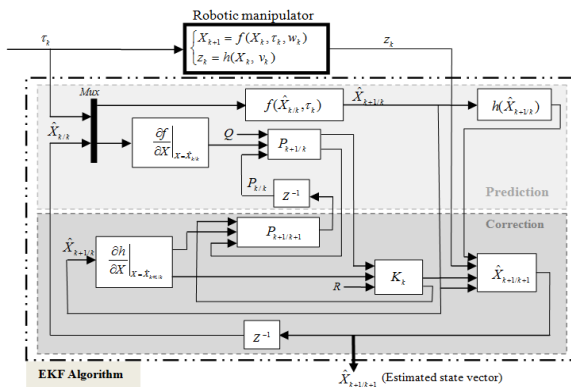


Fig. 1. Extended Kalman filter structure

The discrete state-space model describing a nonlinear process is given by:

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

where τ_k and z_k are the control input and output vectors at time instant k . $f(\cdot)$ represents the evolution function of the system, whereas $h(\cdot)$ represents the relationship between the state vector and the measurement result z_k . 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, Eq. (1) must be linearized by using the first order Taylor approximation around the desired reference point ($\hat{X}_k, \hat{w}_k = 0, \hat{v}_k = 0$), which gives us the following approximated linear model:

$$\begin{cases} X_{k+1} \approx f(X_k, \tau_k, w_k) \approx \\ f(\hat{X}_k, \tau_k, 0) + F_k(X_k - \hat{X}_k) + W_k(w_k - 0) \\ z_k \approx h(X_k, v_k) \approx \\ h(\hat{X}_k, 0) + H_k(X_k - \hat{X}_k) + V_k(v_k - 0) \end{cases} \quad (2)$$

where the Jacobean matrices of f and h are given as follows:

$$\begin{aligned} F_k &= \left. \frac{\partial f(X, 0)}{\partial X} \right|_{X=\hat{X}}, & W_k &= \left. \frac{\partial f(\hat{X}_k, w)}{\partial w} \right|_{w=0}, \\ H_k &= \left. \frac{\partial h(X, 0)}{\partial X} \right|_{X=\hat{X}} & \text{and} & V_k = \left. \frac{\partial h(\hat{X}_k, v)}{\partial v} \right|_{v=0} \end{aligned} \quad (3)$$

The EKF is a recursive algorithm that is used for estimating state vector of the nonlinear systems, given the measurement z_k by filtering out the noises. This is carried out using the Prediction and Correction. It can be described as follows

Prediction:

$$\begin{aligned} \hat{X}_{k+1/k} &= f(\hat{X}_{k/k}, \tau_k, 0) \\ P_{k+1/k} &= F_k P_{k/k} F_k^T + W_k Q W_k^T \end{aligned} \quad (4)$$

Kalman filter gain matrix

$$K_k = P_{k+1/k} H_k^T (H_k P_{k+1/k} H_k^T + V_k R V_k^T)^{-1} \quad (5)$$

Correcction:

$$\begin{aligned} \hat{X}_{k+1/k+1} &= \hat{X}_{k+1/k} + K_k(z_k - h(\hat{X}_{k+1/k}, 0)) \\ P_{k+1/k+1} &= P_{k+1/k} - K_k H_k P_{k+1/k} \end{aligned} \quad (6)$$

where $\hat{X}_{k+1/k+1}$ denotes the posteriori state prediction vector, $\hat{X}_{k+1/k}$ is the priori state prediction vector, $P_{k+1/k+1}$ denotes the posteriori prediction error covariance matrix, $P_{k+1/k}$ is the priori prediction error covariance matrix. Extended Kalman filter framework is presented in Figure 1.

3. Biogeography-Based Optimization

BBO is a new evolutionary algorithm inspired by biogeography, which is developed by Dan Simon in 2008. It is a population-based stochastic search algorithm. Similar to other evolutionary computation

algorithms, such as PSO, BBO is a search method that exploits the theory of island biogeography^{38,39}, it is designed based on the migration strategy of (animals, fish, birds, or insects) to solve the problem of optimization. In BBO, the population represents a number of habitats (or islands), each habitat represents a possible solution for the problem at hand, and each feature of the habitat is called a suitability index variable (SIV). A quantitative performance index, called habitat suitability index (HSI), is used as a measure of the quality of a solution; which is analogous to fitness in other optimization algorithms. High- HSI habitat represents a good solution and low-HSI habitat represents a poor solution. Solution features emigrate from high-HSI habitats (emigrating habitat) to low-HSI habitats (immigrating habitat). BBO algorithm works on the basis of two concepts: migration and mutation

The migration operators, which are emigration and immigration, are used to improve and evolve a solution to the optimization problem. Migration involves two main processes immigration (λ_s) and emigration (μ_s). These parameters are affected by the number of species (s) in a habitat and they are used to probabilistically share information between habitats. Figure 2 shows the relationship between immigration rate, emigration rate. It has two main operators, which are migration (including emigration and immigration) and mutation. The immigration rate λ_s and emigration rate μ_s and the number of species (s) can be modeled as Figure 2.

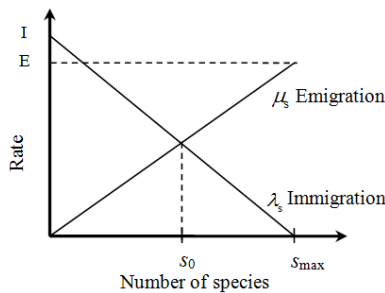


Fig. 2. A Linear model of immigration and emigration rates³⁸

As is clear from Figure 2 that I and E are the maximum possible immigration and emigration rates, respectively. s_0 is the equilibrium number of species and s_{max} is the maximum species number.

The immigration and emigration rates are given as

$$\lambda_s = I[1 - (s/s_{max})] \quad (7)$$

$$\mu_s = E(s/s_{max}) \quad (8)$$

Migration operator is used to modify existing islands by mixing features within the population.

Algorithm 1 Migration

```

1: for  $i = 1$  to  $NP$  do
2:   Use  $\lambda_{si}$  to probabilistically decide whether to
     immigrate to  $X_i$ 
3:   if  $\text{rand}(0, 1) < \lambda_{si}$  then
4:     for  $j = 1$  to  $NP$  do
5:       Select the emigrating island  $X_j$  with
         probability  $\alpha \mu_{sj}$ 
6:       if  $\text{rand}(0, 1) < \mu_{sj}$  then
7:         Replace a randomly selected decision
           variable (SIV) of  $X_i$  with its
           corresponding variable in  $X_j$ 
8:       end if
9:     end for
10:   end if
11: end for

```

where NP denote population size, $\text{rand}(0,1)$ is a uniformly distributed random number in the interval $[0, 1]$ and X_{ij} is the j^{th} SIV of the solution X_i .

Mutation operator is used to changing SIV within a habitat itself, and thus probably increase diversity of the population. For each habitat, a species number probability $P_s = \mu_s / \sum_{i=1}^n \mu_{si}$ indicates the probability that habitat H_b is expected a priori as a solution to the problem. In this context, very high HSI habitats and very low HSI habitats are both equally improbable, and medium HSI habitats are relatively probable. The mutation $m(s)$ is inversely proportional to the probability P_s of the solution:

$$m(s) = m_{max} \left(\frac{1 - P_s}{P_{max}} \right) \quad (9)$$

where m_{max} is a user-defined parameter, and $P_{max} = \arg\max_x P_s$, and $s = 1, \dots, s_{max}$.

Biogeography-based optimization algorithm main steps are illustrated by the flowchart in Figure 3.

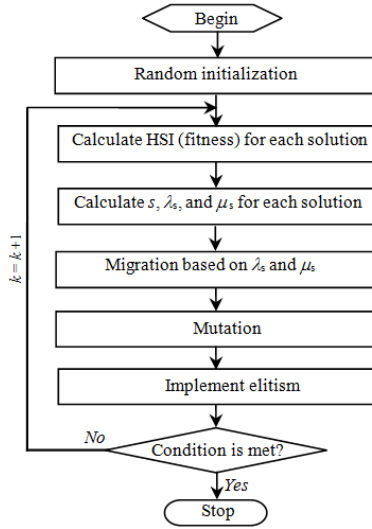


Fig. 3. Algorithm flowchart of BBO

4. Problem formulation and proposed method

4.1. Problem formulation

The dynamic model of a robotic manipulator in the standard form can be represented by ⁴⁰:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau \quad (10)$$

where $\theta, \dot{\theta}, \ddot{\theta} \in \mathbb{R}^n$ are the joint position, velocity and acceleration, respectively. $M(\theta) = M_0(\theta) + \Delta M(\theta) \in \mathbb{R}^{n \times n}$ is a symmetric positive definite inertia matrix. $C(\theta, \dot{\theta}) = C_0(\theta, \dot{\theta}) + \Delta C(\theta, \dot{\theta}) \in \mathbb{R}^{n \times n}$ comprises Coriolis and centrifugal forces. $G(\theta) = G_0(\theta) + \Delta G(\theta) \in \mathbb{R}^n$ is the vector of gravitational forces. τ is the control torque vector, in which $M_0(\theta), C_0(\theta, \dot{\theta})$ and $G_0(\theta)$ are nominal terms, whereas $\Delta M(\theta), \Delta C(\theta, \dot{\theta})$ and $\Delta G(\theta)$ are the parameters uncertainties.

The dynamic model of a robotic manipulator (10) with uncertainties and disturbances can be rewritten as following:

$$M_0(\theta)\ddot{\theta} + C_0(\theta, \dot{\theta})\dot{\theta} + G_0(\theta) = \tau + d(\theta, \dot{\theta}, \ddot{\theta}) \quad (11)$$

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

Our objective is to design a bounded control law for the input τ such that all signals are bounded and

the actual state trajectories $X = [\theta_1, \theta_2]^T$, converge to the desired trajectories $X_d = [\theta_{1d}, \theta_{2d}]^T$ as closely as possible, for all time interval $T = [0, t_f]$ when t tends to infinity despite the presence of parametric uncertainties and external disturbance. Two conventional properties of the robot manipulators are considered.

Property 1 $M(\theta)$ is symmetric and positive definite, $M^T = M$

Property 2 $(\dot{M} - 2C)$ is skew-symmetric, i.e. for any vector X , we have $M^T(\dot{M} - 2C)M = 0$

In respect of the dynamic system presented in (11), the following assumptions will be made:

Assumption 1 The joint positions and the joint speeds are unavailable.

Assumption 2 The disturbances d is unknown but bounded, i.e. $|d| \leq D$.

Assumption 3 The desired trajectory $\theta_d, \dot{\theta}_d$ and $\ddot{\theta}_d$ are available and with known bounds.

Assumption 4 The allowable values of the control input $\tau(t)$ are limited between an upper and lower bounds $\underline{\tau}$ and $\bar{\tau}$ such that $\underline{\tau} \leq \tau(t) \leq \bar{\tau}$.

The robustness of closed loop system against parametric uncertainties and external disturbances needs a robust control law. To attain this objective, we apply the sliding mode control approach⁴¹, this choice is motivated by its design simplicity and its high robustness against uncertainties and external disturbances.

4.2. Sliding mode control based on interval type-2 fuzzy for 2-Links manipulators

In this section, the integration of interval type-2 fuzzy logic in the sliding mode control is discussed to make an intelligent controller so-called interval type-2 fuzzy Sliding mode control based on switching gain.

For designing the proposed control law, we let the estimated tracking error:

$$\hat{e} = \hat{\theta} - \theta_d \quad (12)$$

where $\hat{\theta}$ is the estimated position and θ_d is the desired one.

The first step of SMC is to design the sliding surface S , which can be defined as⁴²:

$$S(\hat{X}, t) = \hat{e} + \lambda \hat{e} \quad (13)$$

where λ is diagonal positive definite matrix.

The time derivative of (14) is given as:

$$\dot{S}(\hat{X}, t) = \ddot{\hat{e}} + \lambda \dot{\hat{e}} \quad (14)$$

The Second step of SMC is to design the control law. In the traditional SMC, the discontinuous control law is selected as:

$$\tau_{dis} = -K \text{sign}(S) \quad (15)$$

where K is diagonal positive definite matrix.

The equivalent control τ_{eq} is given by the following condition⁴¹:

$$S = 0 \text{ and } \dot{S} = 0 \implies \tau(t) = \tau_{eq}(t) \quad (16)$$

In order to govern the system states to follow the desired reference trajectory just now to make $S = 0$ attractive. Therefore, $\hat{e} \rightarrow 0$ as $t \rightarrow \infty$.

A sufficient condition to achieve this behavior is to select the control strategy so that the following sliding condition^{41,43} is satisfied:

$$S\dot{S} \leq -\eta|S|, \quad \eta > 0 \quad (17)$$

Consider the control problem of nonlinear uncertain system (11), the conventional sliding mode control is given as:

$$\begin{aligned} \tau &= \tau_{eq} + \tau_{dis} \\ &= M_0(\ddot{\theta}_d - \lambda \dot{\hat{e}}) + C_0(\dot{\theta}_d - \lambda \hat{e}) + \\ &\quad G_0 - d - K \text{sign}(S) \end{aligned} \quad (18)$$

The use of the discontinuous sign function will excite an undesired phenomenon called chatter, which is caused by the discontinuous switching function. In this context, high switching gain K of τ_{dis} in (15) will lead to an increase in oscillations of the control torque 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. In the literature, various solutions can be found to overcome this problem^{43,12,13,14,6}. In our work, interval type-2 fuzzy logic system^{44,45,46,47} has been used to realize this idea by combining IT2FLS with discontinuous control according to some appropriate fuzzy rules. A type-2 fuzzy logic system (T2FLS)^{45,46}, consists of five parts: the fuzzifier, knowledge base, fuzzy inference engine, type-reducer, and 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_i : If S is A_i , Then K_{IT2FLS} is B_i

where S and K_{IT2FLS} are an interval type-2 fuzzy logic system input and output, respectively. A_i and B_i are an interval type-2 membership functions of input and output, respectively. i denoted i^{th} rule of an IT2-Mamdani fuzzy system. Process of input-output mapping can be formulated as follows:

1. Calculate the weight interval of each rule:

$$\omega_i \in [\min(\mu_{A_i}) \quad \max(\mu_{A_i})] \quad (19)$$

2. Compute the weighted output from all rules (type reduction) based on center of sets type reducer¹¹

$$\begin{aligned} y_l(S) &= \min_{\omega_i} \left(\frac{\sum y_l(S) \cdot \omega_i}{\sum \omega_i} \right), \\ y_r(S) &= \max_{\omega_i} \left(\frac{\sum y_l(S) \cdot \omega_i}{\sum \omega_i} \right) \end{aligned} \quad (20)$$

3. Calculation of crisp output (defuzzification) based on arithmetic mean

$$K_{IT2FLS}(S) = \frac{y_l + y_r}{2} \quad (21)$$

Therefore, control law (18) becomes:

$$\begin{aligned}\tau &= \tau_{eq} + K_{IT2FLS} \tau_{dis} \\ &= M_0(\ddot{\theta}_d - \lambda \dot{\hat{e}}) + C_0(\dot{\theta}_d - \lambda \hat{e}) + G_0 - \\ &\quad d - K_{IT2FLS} K \operatorname{sign}(S) \\ &= M_0(\ddot{\theta}_d - \lambda \dot{\hat{e}}) + C_0(\dot{\theta}_d - \lambda \hat{e}) + \\ &\quad G_0 - d - \hat{K} \operatorname{sign}(S) \quad (22)\end{aligned}$$

where $\hat{K} = K_{IT2FLS} \times K$.

Therefore, it can be easily verified that (22) is sufficient to impose the sliding condition

$$S\dot{S} \leq -\hat{K}|S|, \text{ with } \hat{K} > 0 \quad (23)$$

which, in fact, ensures the finite-time convergence of the tracking error vector to the sliding surface S and, consequently, its exponential stability.

In EKF, the determination of Q and R covariance matrices is a difficult task, especially when the corresponding noises have unknown stochastic properties. In order to avoid this problem, we will consider these matrices as free parameters to be adjusted. In the literature, Vas³³ was the first who adjusted these matrices manually with trial-error method. Unfortunately, this method is a tedious task. Therefore, to overcome this difficulty and to avoid trial-error method, the authors^{34,35,36} have used genetic algorithms, downhill simplex, and particle swarm optimization, respectively, to optimize these matrices. In our work, we suggest using a novel optimization method for the adjusting these matrices by using the BBO algorithm³⁷ (see Figure 4).

4.3. Proposed optimized extended Kalman filter (BBO-EKF algorithm)

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. According to Vas³³, Q and R must be acquired taking into account the stochastic properties of the corresponding noises, for this reason they are generally unknown matrices. However, as these are not known,

in most cases are used as weighting parameters (adjustment of parameters). Moreover, these matrices were first tuned manually by trial-error methods, which are very tedious procedures due to a large time consumption³⁴. To avoid the computational complexity of this method, and when the values of these matrices are not known precisely, the improvement of the EKF performance can be assimilated to an optimization problem.

In this paper, we propose a novel alternative for the tuning and the optimization of Q and R based on BBO algorithm, which is described in section 3. In our algorithm, each habitat is considered as an individual and has its habitat suitability index (HSI) instead of fitness value to show the degree of its goodness. High- HSI habitat represents a good solution and low-HSI habitat represents a poor solution. Solution features emigrate from high-HSI habitats (emigrating habitat) to low-HSI habitats (immigrating habitat).

In our works, we suppose that we have a population size of NP , that x_k is the k – th individual in the population, that the dimension of our optimization problem is n , and that $x_k(s)$ is the s – th independent variable in x_k , where $k \in [1, NP]$ and $s \in [1, n]$. At each generation and for each solution feature in the k – th individual, there is a probability of λ_{sk} (immigration probability) that it will be replaced by (7). If a solution feature is selected to be replaced, then we select the emigrating solution with a probability that is proportional to the emigration probabilities μ_{sk} in (8). Mean square error (MSE) criterion defined in (24) is used in this paper as fitness (objective function), between the actual output and the estimated one according to a certain number of iterations T to be performed for each step of estimation.

$$MSE = \frac{1}{T} \sum_{k=1}^T (z_i(k) - \hat{z}_i(k))^2, \quad i = 1, 2 \quad (24)$$

where \hat{z} is an estimate of the output z ; and T denotes the number of data samples.

This is carried out using the two steps. As a first step, we present a BBO-EKF algorithm as in Figure 4, which is done in an offline, because BBO algorithm requires several iterations to achieve optimal solutions. For each iteration, BBO-EKF algorithm

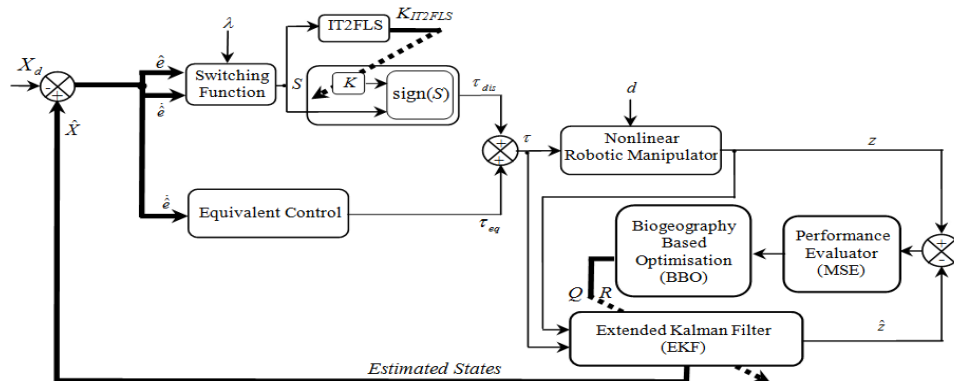


Fig. 4. Block diagram of the control system based on the proposed BBO-EKF observer

must be executed once. Therefore, BBO-EKF algorithm should be executed several times allowing the optimization of Q and R , from each measurement. The main steps of the biogeography-based optimization algorithm are illustrated in the flowchart in Figure 3, in which seven steps can be distinguished:

- 1) Initialize a set of solutions to a problem randomly
- 2) Calculate HSI (fitness) for each solution
- 3) For each habitat map the HSI to the number of species s , calculate the immigration rate λ_s , and emigration rate μ_s by using (7) and (8).
- 4) Modify habitats (Migration) based on λ_s and μ_s , see Algorithm 1
- 5) Mutation according to (9)
- 6) Implement elitism to retain the best solution in the population from one generation to the next
- 7) Go to step 2 for the next iteration if needed. This loop can be terminated after a predefined number of generations or after an acceptable problem solution has been found.

As a second step, after obtaining the optimized values Q and R in the first step, we inject them into the EKF observer running online to estimate the state variable of two-link manipulator.

The control torque input τ and the measured response z will be considered as input signals to EKF observer, where τ is applied to both robotic manipulator and extended Kalman filter (see Figure 4).

4.4. Stability of closed-loop control system

The actual output z and the estimated output \hat{z} are set to be the inputs of the performance evaluator of the BBO module through a comparator. The MSE is calculated by the performance evaluator. Then, obtained values of MSE will be used in the BBO algorithm. Based on these values, BBO optimizer will calculate and optimize the unknown parameters of covariance matrices Q and R .

The new solutions and updated matrices Q and R are then used to adapt the EKF for 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.

In order to dominate the states of system to arrive 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 (17) is satisfied. This goal is assured by following lemma.

Lemma 1. Consider the uncertain nonlinear system (11) and assumptions 1-4. Then, the controller defined by (22), and (23) ensures the convergence of tracking error vector to the sliding surface S .

Proof. Consider the Lyapunov function candidate:

$$L = 1/2 S^T M S \quad (25)$$

Its time derivative is given as:

$$\dot{L} = 1/2 S^T \dot{M} S + S^T M \dot{S} \quad (26)$$

Considering property 2, then

$$S^T (1/2 \dot{M} - C) S = 0 \quad (27)$$

Combining (25)–(27), one can get

$$\begin{aligned} \dot{L} &= 1/2 S^T (\dot{M} - 2C) S + S^T C S + S^T M \dot{S} \\ &= S^T (C S + M \dot{S}) = S^T [C S + M(\ddot{\hat{e}} + \lambda \dot{\hat{e}})] \\ &= S^T [C S + M(\ddot{\hat{\theta}} - \ddot{\theta}_d) + M \lambda (\dot{\hat{\theta}} - \dot{\theta}_d)] \\ &= S^T [C(\hat{\theta} - \theta_d) + C \lambda (\hat{\theta} - \theta_d) - M \ddot{\theta}_d - \\ &\quad C \hat{\theta} - G + \tau + M \lambda (\hat{\theta} - \theta_d)] \\ &= S^T [M(\lambda(\hat{\theta} - \theta_d) - \ddot{\theta}_d) + \\ &\quad C(\lambda(\hat{\theta} - \theta_d) - \ddot{\theta}_d) - G + \tau] \\ &= S^T [M_0(\lambda(\hat{\theta} - \theta_d) - \ddot{\theta}_d) + \\ &\quad C_0(\lambda(\hat{\theta} - \theta_d) - \ddot{\theta}_d) - G_0 + \tau + d] \end{aligned}$$

In order to obtain the stability in closed-loop, the derivative of Lyapunov function must be negative definite ($\dot{L} \leq 0$), the control law τ can be chosen as:

$$\begin{aligned} \tau &= M_0 (\ddot{\theta}_d - \lambda(\hat{\theta} - \theta_d)) + C_0 (\dot{\theta}_d - \\ &\quad \lambda(\hat{\theta} - \theta_d)) + G_0 - d - \hat{K} \text{sign}(S) \end{aligned} \quad (28)$$

Therefore, the derivative \dot{L} becomes:

$$\dot{L} = -S^T \hat{K} \text{sign}(S) = -\hat{K} |S| \leq 0 \quad (29)$$

which implies that $S(t) \rightarrow 0$ as $t \rightarrow \infty$. Therefore, $\hat{e}(t)$ and all its derivatives up to converge to zero. \square

5. Simulation and discussions

In order to verify the robustness and effectiveness of the proposed control framework, let us consider two degrees of freedom planar manipulator with revolute joints shown in Figure 5.

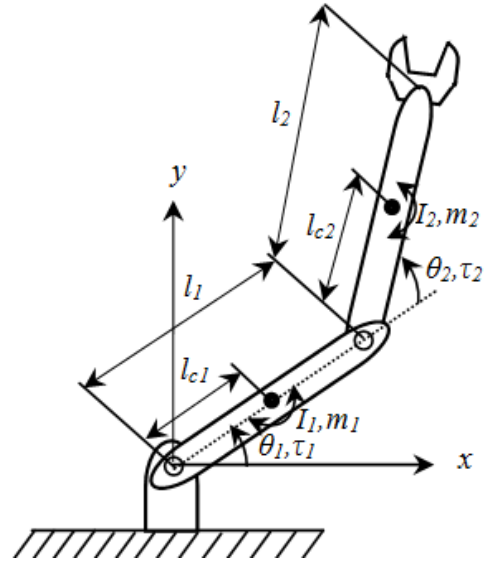


Fig. 5. Two-link robot manipulator

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

The dynamic of two-link manipulator can be described in the following differential equations³⁸:

$$\begin{aligned} \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \\ = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \end{aligned} \quad (30)$$

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

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

The matrix $C_0 = [c_{ij}]_{2 \times 2}$ is given by: $c_{11} = a\dot{\theta}_2$, $c_{12} = a\dot{\theta}_1 + a\dot{\theta}_2$, $c_{21} = -a\dot{\theta}_1$, $c_{22} = 0$, where $a = -m_2 l_1 l_{c2} \sin(\theta_2)$.

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

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

Since the 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.

Let the state vector be given by $X = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]^T = [x_1, x_2, x_3, x_4]^T$, and then the resulting global discrete form will be given by the following discrete nonlinear representation:

$$\begin{cases} x_1(k+1) = x_1(k) + \Delta t \, x_2(k+1) + w_1(k) \\ x_2(k+1) = x_2(k) + \Delta t \, [f_1(X, k) + \\ \quad g_1(X, k) \, \tau(k) + d_1(k)] + w_2(k) \\ x_3(k+1) = x_3(k) + \Delta t \, x_4(k+1) + w_3(k) \\ x_4(k+1) = x_4(k) + \Delta t \, [f_2(X, k) + \\ \quad g_2(X, k) \, \tau(k) + d_2(k)] + w_4(k) \\ z_1(k) = x_1(k) + v_1(k) \\ z_2(k) = x_3(k) + v_2(k) \end{cases} \quad (31)$$

where $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = M_0^{-1} (-C_0[x_2, x_4] - G_0)$, $\begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = M_0^{-1}$, $\tau = [\tau_1, \tau_2]^T$ is the control torque input, and Δt is the sampling period and k is the discrete-time points.

The nominal parameters of the robot used are chosen by $m_1 = m_2 = 1 \text{ Kg}$, $l_1 = l_2 = 0.5 \text{ m}$, $l_{c1} = l_{c2} = 0.25 \text{ m}$, $I_1 = I_2 = 0.1 \text{ Kg.m}^2$, $g = 9.81 \text{ m/s}^2$.

In this section, our proposed algorithm is simulated on a PC using Matlab software environment (version 8.6.0.267246). A total of $T = 2000$ measurement data are simulated on a time interval from 0 to 2s. Note that all codes are written in Matlab language in M-files with step size $\Delta t = 0.001 \text{ s}$.

The desired reference trajectories is chosen as $X_d = [70^\circ, 90^\circ]^T$. The initial values of the robot were selected as $X_d = [0, 0, 0, 0]^T$. Three types of uncertainties will be injected in the structure to verify the robustness of controller. Firstly parameters uncertainties (+10% over the values of nominal model parameters). Secondly the external disturbances are assumed to be time-varying as follows: $\delta_1 = 0.3 \times \text{rand}$, and $\delta_2 = 0.3 \times \text{rand} \times \sin(t)$. Note that both disturbances first and second sum to d and they will be applied at $t > 1 \text{ s}$. Thirdly random Gaussian noises for the states and for the measurements

both with zero mean values and with covariances $q = 10^{-2}$ and $r = 10^{-4}$, respectively. $D = 1$

EKF is implemented as in (3) to (6). The Jacobean matrices F_k , W_k , H_k , V_k are defined in appendix A; EKF will provide the state estimate vector $\hat{X} = [\hat{\theta}_1, \hat{\dot{\theta}}_1, \hat{\theta}_2, \hat{\dot{\theta}}_2]^T$. 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 our simulation case, error covariance matrix P is set to a 4×4 matrix, and Q and R matrices with dimensions 4×4 and 2×2 , respectively, are assumed as

$$Q = \text{diag}(q_{\theta_1}, q_{\dot{\theta}_1}, q_{\theta_2}, q_{\dot{\theta}_2}) = \begin{bmatrix} q_{\theta_1} & 0 & 0 & 0 \\ 0 & q_{\dot{\theta}_1} & 0 & 0 \\ 0 & 0 & q_{\theta_2} & 0 \\ 0 & 0 & 0 & q_{\dot{\theta}_2} \end{bmatrix} \quad (32)$$

$$R = \text{diag}(r_1, r_2) = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \quad (33)$$

For comparison purposes, the performance of EKF with diverse compositions of Q and R is evaluated by using the mean-squared-error (24) of the position-estimating response, which is defined as: $MSE = \frac{1}{T} \sum_{k=1}^T [\theta_i(k) - \hat{\theta}_i(k)]^2$, $i = 1, 2$.

First, we simulate the system under control law of traditional sliding mode action in order to show the drawback of SMC taken alone. Applying the control law (18), and after some trials, we chosen $K = 20I_{2 \times 2}$ and $\lambda = 5I_{2 \times 2}$, where $I_{i \times i}$ is an $i \times i$ identity matrix.

Table 1 shows typical EKF performance with their corresponding covariance matrices' entries ($q_{\theta_1}, q_{\dot{\theta}_1}, q_{\theta_2}, q_{\dot{\theta}_2}, r_1$ and r_2) obtained by a trial-error method. It is found that good estimation performance results when Q and R are equal (case 2 and 3 in Table 1), but a bad selection of ($q_{\theta_1}, q_{\dot{\theta}_1}, q_{\theta_2}, q_{\dot{\theta}_2}, r_1$ and r_2) can produce a poor estimation performance results (case 1). Note that the best estimation performance is obtained with Q and R matrices ($q_{\theta_1} = q_{\dot{\theta}_1} = 0.01, q_{\theta_2} = 0.02, q_{\dot{\theta}_2} = 0.01, r_1 = 0.01$, and $r_2 = 0.08$) (case 4), which corresponds to the smallest MSE.

Table 1. EKF performances for a two-link manipulator using trial-error estimations.

Case	Q and R entries						MSE (rad)	Estimation quality
	q_{θ_1}	\dot{q}_{θ_1}	q_{θ_2}	\dot{q}_{θ_2}	r_1	r_2		
1	1	1	1	1	1	1	1.0881	Poor
2	0.1	0.1	10^{-3}	0.1	10^{-4}	10^{-4}	1.9894×10^{-5}	Good
3	0.1	0.01	0.1	0.1	0.1	10^{-6}	1.4567×10^{-5}	Good
4	0.01	0.01	0.02	0.01	0.01	0.08	1.4287×10^{-5}	Very good

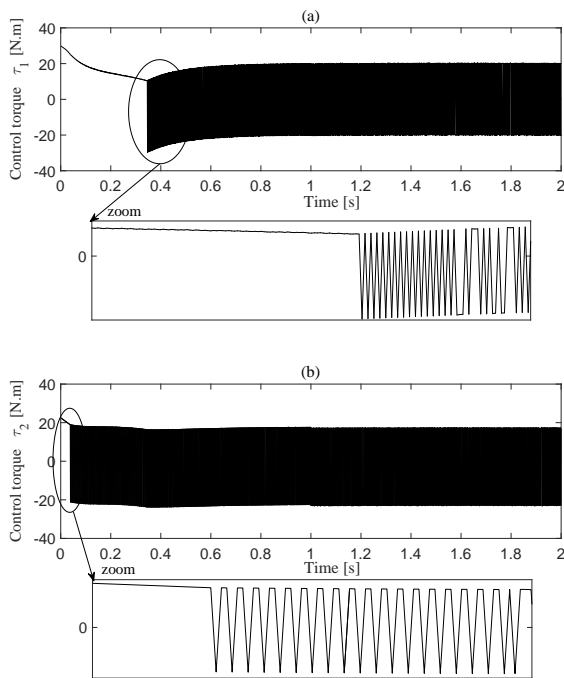


Fig. 6. Control torque inputs using traditional SMC (a) Link-1, (b) Link-2

The control torque inputs relative to the best case (case 4) are showed in Figure 6, where we see clearly that the control torque 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 interval type-2 fuzzy logic is exploited as seen in subsection 4.2 to reduce the chattering effect.

5.1. Chattering phenomenon problem

In this section, to avoid chattering phenomenon, one-input one-output IT2FLS is designed with an input S , which reflects the distance of error trajectory to the sliding surface. Output of IT2FLS is denoted by K_{IT2FLS} . The memberships functions of S are chosen as illustrated in Figure 7(a), in which the following linguistic variables have been used: negative (N), zero (Z), positive (P). The memberships functions of K_{IT2FLS} are chosen as illustrated in Figure 7(b), in which the following linguistic variables have been used: positive small (PS), positive big (PB). The rule set of the adopted IT2FLS contains 3 rules defined as following:

- Rule 1: If S is N, Then K_{IT2FLS} is PB
- Rule 2: If S is Z, Then K_{IT2FLS} is PS
- Rule 3: If S is P, Then K_{IT2FLS} is PB

These rules govern the input-output relationship between S and K_{IT2FLS} by adopting the Mamdani-type inference engine, in which the center of gravity method is used for defuzzification as in (21). By considering the assumption 4, in which $-30 \leq \tau(t) \leq 30$, the simulation results corresponding to improvement of switching gain K are presented using IT2FSMC with standard EKF.

Figure 8(a) shows desired, actual and estimated position of Link-1, and as we see the performance under occurrence parameter variations and external disturbance are satisfactory (see also Figure 8(b)), which represents the position tracking error of Link-1.

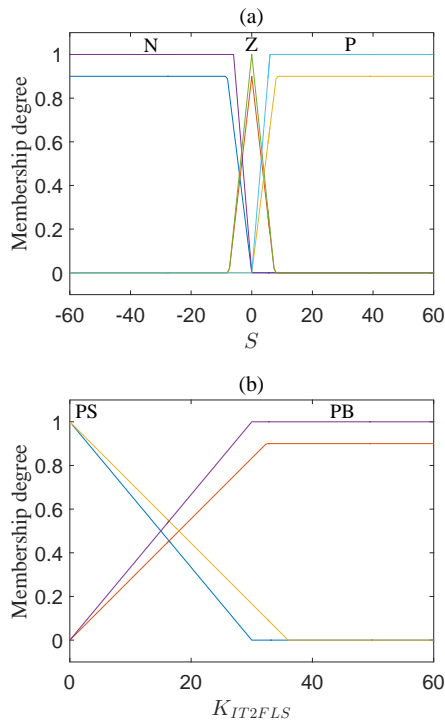


Fig. 7. (a) Input membership functions (b) Output membership functions

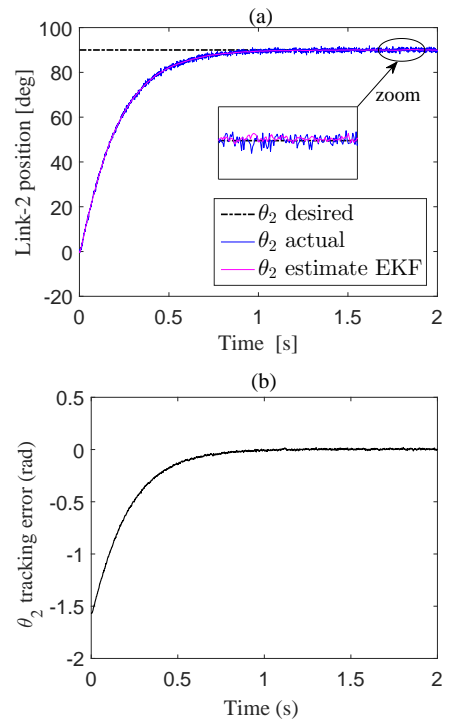


Fig. 9. (a) Desired, actual and estimated position of Link-2 (b) Position tracking error of Link-2, using IT2FSMC with standard EKF

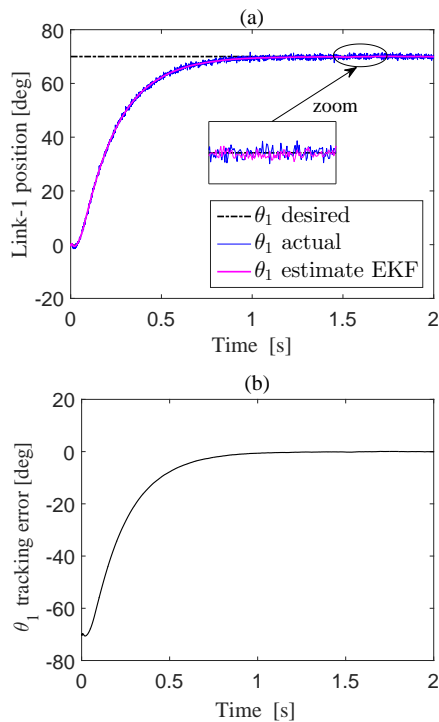


Fig. 8. (a) Desired, actual and estimated position of Link-1 (b) Position tracking error of Link-1, using IT2FSMC with standard EKF

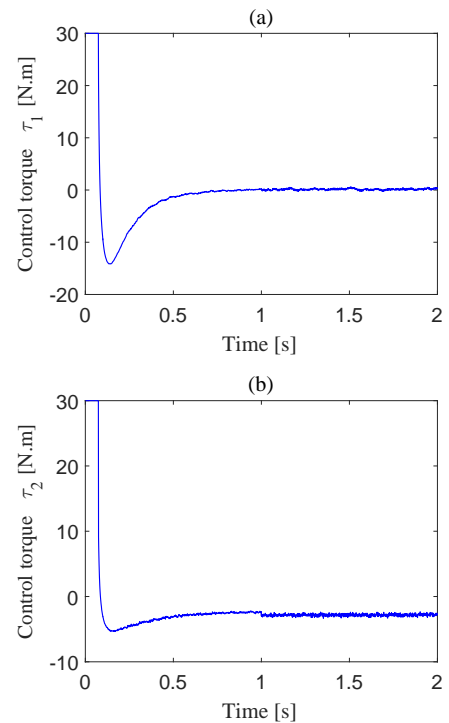


Fig. 10. Control torque inputs using IT2FSMC with standard EKF for (a) Link-1, (b) Link-2

Figure 9(a) shows desired, actual and estimated position of Link-2, and as we see the performance under occurrence parameter variations and external disturbance are satisfactory (see also Figure 9(b)), which represents the position tracking error of Link-2. From comparing the new obtained control torque inputs in Figure 10 with the old one (Figure 6), we can clearly see that the chattering phenomenon is disappeared. The fuzzy gain K_{IT2FLS} is depicted in Figure 11.

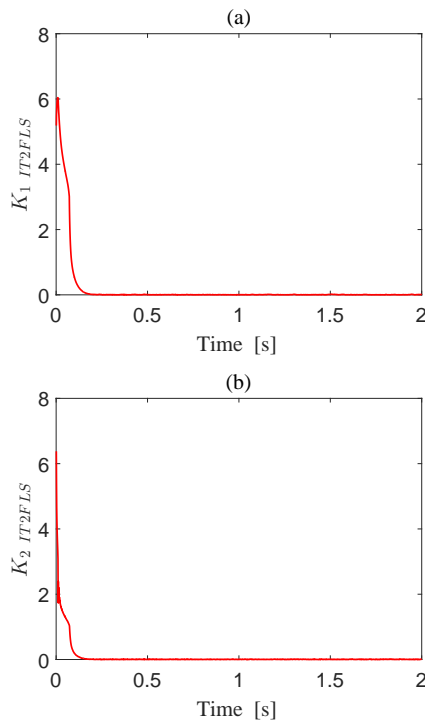


Fig. 11. Fuzzy gain K_{IT2FLS} obtained by IT2FLS for (a) Link-1, (b) Link-2

From simulation results, it is clear that IT2FSMC controller provides desired response with smooth control signal and minimum reaching time during model uncertainties and disturbances. Note that the prediction accuracy of EKF is not quite satisfactory (see Figures 8(a) and 9(a)) due to the trial-error choice for EKF covariance matrices. In what follows, the proposed method will be applied in order to resolve the prediction problem.

5.2. 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 it takes a very longtime. Therefore, a satisfactory performance estimation can only be achieved with a larger effort by the operator experienced (experts). In fact, it is not possible to easily deduce a relationship between the covariance matrices and the best estimation results. In what follows we propose to solve this problem by a new evolutionary algorithm called biogeography-based optimization, and is then compared to PSO algorithm.

5.2.1. BBO-EKF method

In proposed method, BBO-EKF is an optimization algorithm combining the BBO algorithm with the EKF observer. We suggest searching the optimal combination of six variances $q_{\theta_1}, q_{\dot{\theta}_1}, q_{\theta_2}, q_{\dot{\theta}_2}, r_1$ and r_2 simultaneously shown in (32) and (33) to find the optimal covariance matrices Q and R of the EKF, which will allow us to obtain better estimates with higher precision than the trial-error method. In our case, to evaluate the optimal response performance in a finite time T .

By running the BBO-EKF with BBO parameters cited in Table B.1 (see Appendix B), the optimized covariance matrices Q and R and their corresponding performance MSEs for various numbers of iterations are given in Table 2.

In Table 2, the best solution is a habitat having a low MSE. We observe that the MSE is decreased to 2.7394×10^{-6} after 100 iterations. Note that this MSE is very small compared to the MSE obtained by trial-error method ($MSE_{trial-error} = 1.4287 \times 10^{-5}$), which confirm the effectiveness of proposed method.

It should be noted that the convergence of BBO algorithm to the optimal solution depends on the parameters values shown in Table B.1 (see appendix B).

In order to compare the performance of BBO-EKF process with other algorithms, we give in Table 3, performance using PSO-EKF method.

Table 2. Optimized EKF performances using BBO algorithm.

Number of Species	Generation	Q and R entries						MSE_BBO (rad)
		q_{θ_1}	$q_{\hat{\theta}_1}$	q_{θ_2}	$q_{\hat{\theta}_2}$	r_1	r_2	
20	5	0.0054	0.0484	0.0054	0.0035	0.0797	0.0666	5.7355 $\cdot 10^{-6}$
	10	0.0029	0.0696	0.0063	0.0015	0.0823	0.0960	4.1097 $\cdot 10^{-6}$
	50	0.0002	0.0312	0.0302	0.0295	0.0941	0.0643	3.7545 $\cdot 10^{-6}$
	100	0.0001	0.0121	0.0564	0.0188	0.0842	0.0791	2.7394 $\cdot 10^{-6}$

Table 3. Optimized EKF performances using PSO algorithm.

Iterations	Generation	Q and R entries						MSE_PSO (rad)
		q_{θ_1}	$q_{\hat{\theta}_1}$	q_{θ_2}	$q_{\hat{\theta}_2}$	r_1	r_2	
20	5	10^{-3}	0.0214	10^{-2}	0.0632	0.0793	0.0638	9.8340 $\cdot 10^{-6}$
	10	10^{-4}	0.0335	10^{-5}	0.0326	0.0500	0.0832	9.7784 $\cdot 10^{-6}$
	50	10^{-6}	0.0772	10^{-4}	0.0313	0.0673	0.0800	6.9638 $\cdot 10^{-6}$
	100	10^{-5}	0.0675	10^{-5}	0.0157	0.0893	0.0923	5.1932 $\cdot 10^{-6}$

5.2.2. PSO-EKF method

The optimized covariance matrices Q and R and their corresponding performance MSEs that have been obtained using PSO-EKF algorithm are given in Table 4 where we see that the MSE is decreased until 5.1932×10^{-6} after 100 iterations.

From the results shown in Tables 2 and 3, comparison of BBO-EKF and PSO-EKF approaches shows that all are able to find the optimum design covariance matrices Q and R . It can be easily seen that BBO-EKF gives more precise results than PSO-EKF when the number of iteration (generation) increases. It can also be seen that these MSEs are very small compared to the MSE obtained by trial-error ($MSE_{trial-error} = 1.4287 \times 10^{-5}$); therefore, it can be confirmed that IT2FSMC combined with BBO-EKF technique, outperform those of PSO-EKF approach.

It should be noted that the convergence of PSO method to the optimal solution depends on the parameters c_1 , c_2 and w , in which Self-recognition coefficient $c_1 = 1.49$, Social coefficient $c_2 = 1.49$, and Inertia weight $w = 0.73$. The comparison was carried out in the same conditions as in BBO method (initial population, Swarm size, population size).

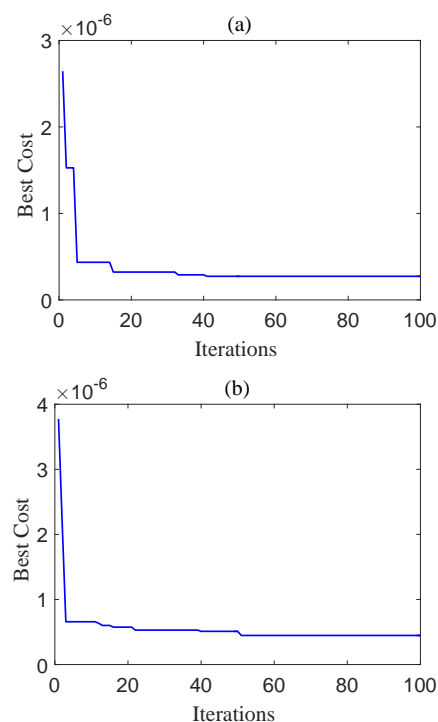


Fig. 12. Mean-square-error comparison versus 100 iterations between (a) BBO-EKF, (b) PSO-EKF

Figure 12 shows the comparison of objective function (MSE) values for the best solutions obtained through the BBO-EKF and PSO-EKF algo-

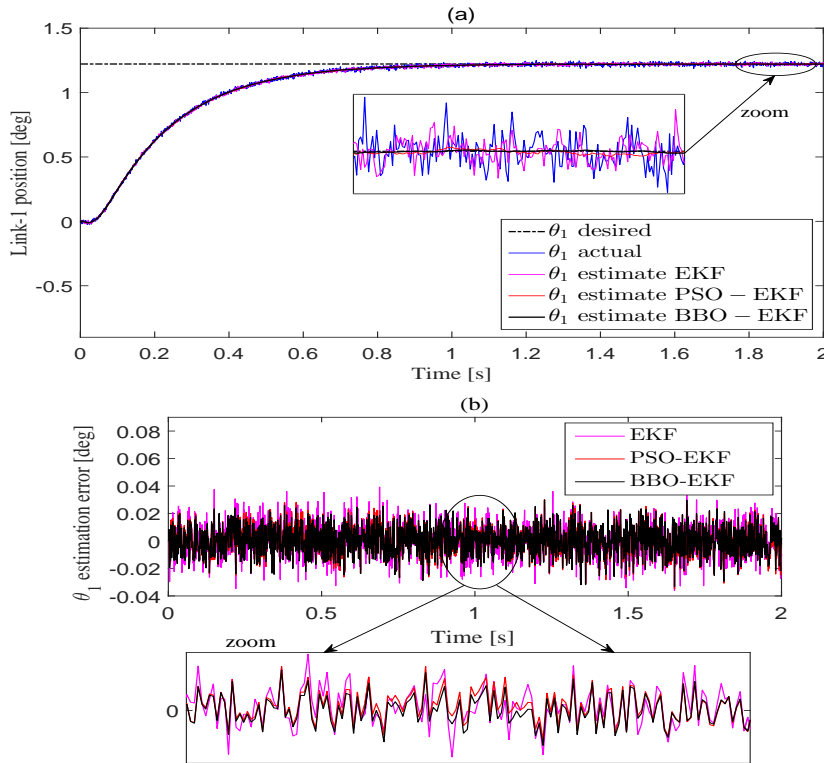


Fig. 13. (a) Desired, actual and estimated position of Link-1 (b) Position estimation errors of Link-1, using IT2FLS SMC for different EKF optimization algorithms

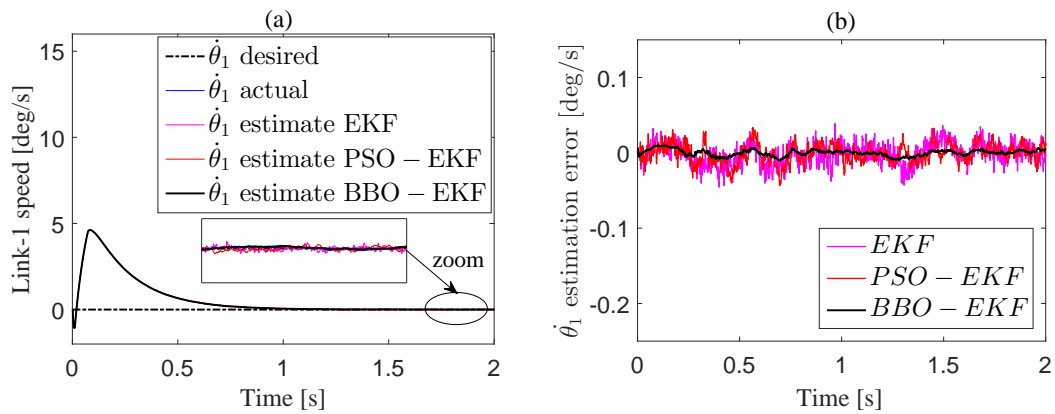


Fig. 14. (a) Desired, actual and estimated speed of Link-1 (b) Speed estimation errors of Link-1, using IT2FLS SMC for different EKF optimization algorithms

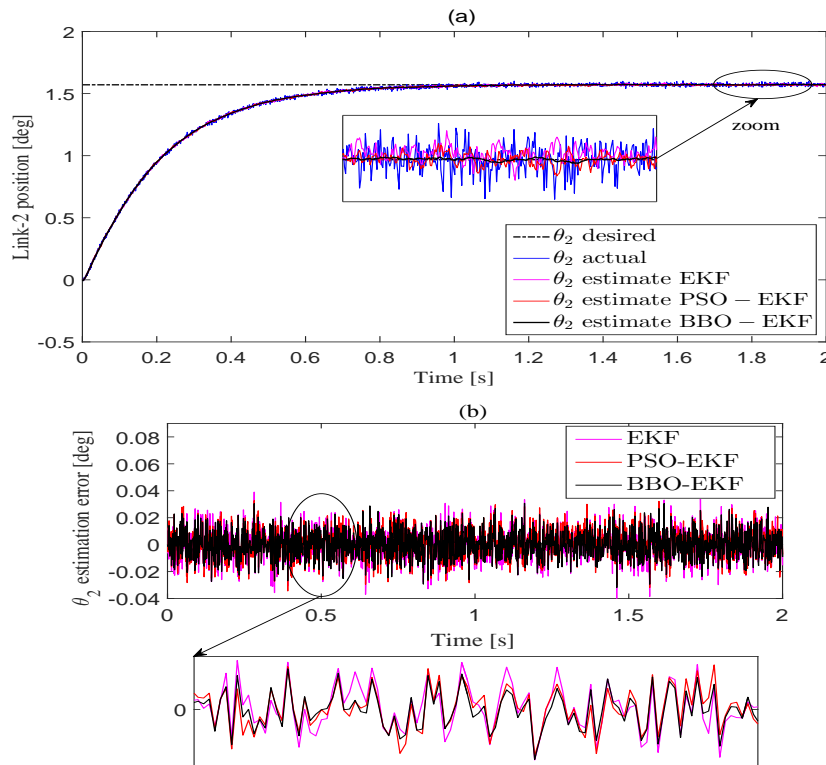


Fig. 15. (a) Desired, actual and estimated position of Link-2 (b) Position estimation errors of Link-2, using IT2FLS SMC for different EKF optimization algorithms

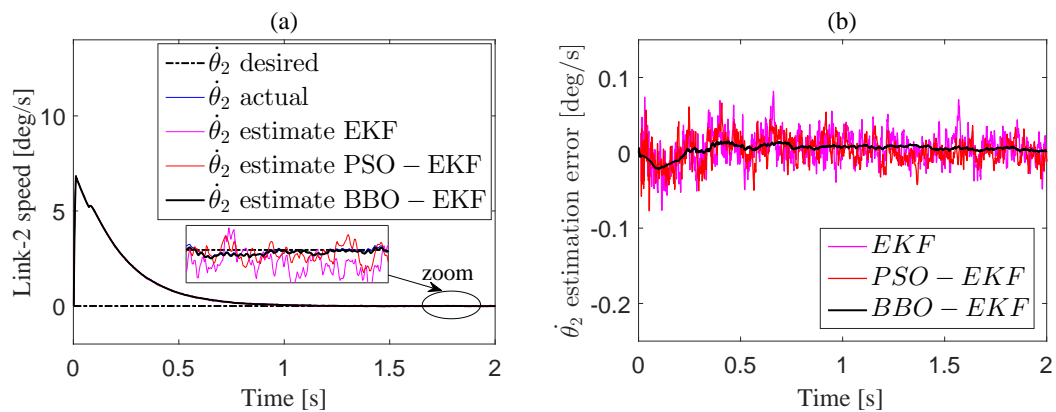


Fig. 16. (a) Desired, actual and estimated speed of Link-2 (b) Speed estimation errors of Link-2, using IT2FLS SMC for different EKF optimization algorithms

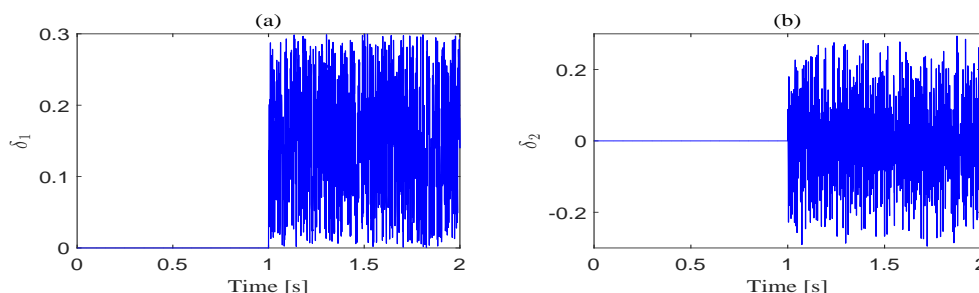


Fig. 17. External disturbances applied on (a) Link-1, (b) Link-2

rithms versus the 100 iterations, respectively. The simulation results of applying IT2FLS approach with proposed optimized EKF are shown in Figures 13 to 17. In Figure 13(a) we present the desired, actual and estimated position responses of Link-1 with optimal values of EKF covariance matrices given in Tables 1, 2 and 3 for trial-error, BBO and PSO methods, respectively. The corresponding position estimation errors are presented in Figure 13(b). The corresponding desired, actual and estimated speed and its estimation errors are presented in Figures 14(a) and (b), respectively. In Figure 15(a) we present the desired, actual and estimated position responses of Link-2 with optimal values of EKF covariance matrices given in Tables 1, 2 and 3 for trial-error, BBO and PSO methods, respectively. The corresponding position estimation errors are presented in Figure 15(b). The corresponding desired, actual and estimated speed and its estimation errors are presented in Figures 16(a) and (b), respectively. The external disturbances applied on Link-1 and Link-2, are depicted in Figures 17(a) and (b), respectively. In all these figures, we see that the best results are obtained with proposed BBO-EKF method where it can be seen that BBO-EKF fits the true state variables with higher accuracy for a two-link manipulator.

6. Conclusions

In this paper, considering parameter uncertainties and external disturbances, we have proposed a novel application of biogeography-based optimization approach for optimizing the extended Kalman filter. The interval type-2 fuzzy system was combined with

sliding mode control to ensure a good robustness. The stability of closed-loop system is guaranteed by means of the Lyapunov stability criterion. The performance of EKF has been improved by adjusting parameters of the covariance matrices Q and R , in which BBO algorithm is used, and it is compared to PSO technique. The proposed optimization methods enable the noise covariance matrices Q and R , on which the performance of EKF critically depends, to be properly selected. A comparison between the IT2FSMC control combined with BBO-EKF, and with PSO-EKF were done in the presence of stochastic measurement noises, confirms that the performance of IT2FSMC combined with BBO-EKF technique was better than it combine with PSO-EKF technique. Simulation results show a significant improvement of the performance while using the proposed optimization methods to improve state variables estimation performance of the two-link manipulator and it was concluded that, the control performance of the IT2FSMC combined with BBO-EKF technique was better than it combine with PSO-EKF technique. Finally, we can say that the obtained results yield better performance while using proposed approach than traditional one.

References

1. N. Kapoor and J. Ohri, Fuzzy Sliding Mode Controller (FSMC) with Global Stabilization and Saturation Function for Tracking Control of a Robotic Manipulator, *J. Control Syst. Eng.* **1** (2013) 50–56.
2. M. R. Soltanpour, B. Zolfaghari, M. Soltani et al. Fuzzy Sliding Mode Control Design for a Class of Nonlinear Systems with Structured and Unstructured Uncertainties, *Int. J. Innov. Comput. Inform. Control*

- 9 (7) (2013) 2713–2726.
3. F. Baklouti, S. Aloui, and A. Chaari, Fuzzy sliding mode fault tolerant control for a class of perturbed nonlinear systems, *Int. J. Model. Identif. Control* **25** (4) (2016) 313–322.
4. R. Naoual et al., Fuzzy sliding mode control for the two-link robot, *Int. J. Syst. Control Commun.* **6** (1) (2014) 84–96.
5. L. Chen, Z. Zhang, H. Wang et al., Sliding mode adaptive control for DC motors using function approximation form, *Int. J. Model. Identif. Control* **26** (3) (2016) 238–252.
6. A. Medjghou, M. Ghanai, N. Bounouara, and K. Chafaa, Fuzzy switch-gain sliding mode control for robotic manipulators based on an extended Kalman filter, in *Proc. 8th Int. Conf. on Modelling, Identification and Control*, (IEEE Xplore, Algiers, Algeria, 2016), pp. 198–206.
7. N. N. Karnik and J. M. Mendel, An Introduction to Type-2 Fuzzy Logic Systems, in *The 1998 IEEE Int. Conf. Fuzzy Syst. Proc.*, (IEEE World Congress on Comput. Intellig., 1998), pp. 915–920.
8. N. N. Karnik, J. M. Mendel, and Q. Liang, Type-2 fuzzy logic systems, *IEEE Trans. Fuzzy Syst.* **7** (6) (1999) 643–658.
9. M. Biglarbegan, W. Melek, and J. M. Mendel, On the robustness of Type-1 and Interval Type-2 fuzzy logic systems in modeling, *Expert Systems with Applications, Elsevier* **73** (2017) 161–17.
10. G. Hasanifard, A. A. Gharaveisi, and M. A. Vali, (2014). Fuzzy modeling for chaotic systems via interval type-2 T-S fuzzy model with parametric uncertainty, *J. Theoret. Appl. Phys.* **8** (1) (2014) 115.
11. O. Castillo, *Type-2 fuzzy logic in intelligent control applications*, (Springer, 2011). Doi: 10.1007/978-3-642-24663-0.
12. G. Zong, Y. Wu, and L. Zhang, Finite time tracking control for rigid robotic manipulators with friction and external disturbances, *J. Syst. Sci. Syst. Engineering* **14** (1) (2005) 115–125.
13. V. Utkin and H. Lee, Chattering Analysis. Advances in Variable Structure and Sliding Mode Control, *Springer Verlag: Berlin* **334** (2006) 107–123.
14. A. Msaddek, A. Gaaloul, and F. M'sahli, Adaptive fuzzy supervision of the gain of the higher order sliding mode control, *Int. J. of Automation and Control* **9** (3) (2015) 228–246.
15. Y. Koubaà, M. Boukattaya, and T. Damak, Path tracking control of non-holonomic wheeled mobile robot with skidding and slipping, *Int. J. of Modelling, Identification and Control* **26** (3) (2016) 218–226.
16. R. E. Kalman, A new approach to linear filtering and prediction problems, *Trans. ASME J. Basic Eng.* **82** (1960) 35–45.
17. D. Luenberger, Observing the state of linear system, *IEEE Trans. Military Electron* **8** (1964) 74–90.
18. A. E. Baran, E. Golubovic, and A. Sabanovic, Functional observers for motion control systems, *AUTOMATIKA* **54** (2) (2013) 231–241.
19. J. Birk and M. Zeitz, Extended-Luenberger observer for non-linear multivariable systems, *Int. J. Control* **47** (6) (1988) 1823–1836.
20. N. Ding, W. ChenR, Y. Zhang et al., An extended Luenberger observer for estimation of vehicle sideslip angle and road friction, *Int. J. of Vehicle Design* **66** (4) (2014) 385–414.
21. M. Hamraoui, M. Balde, and N. Assala, Observability and observer design for kinetic networks, *Int. J. Eng. Syst. Model. Simul.* **1** (2/3) (2009) 79–91.
22. L. Liu, B. Qi, S. Cheng et al., High precision estimation of inertial rotation via the extended Kalman filter, *Springer. Eur. Phys. J. D* **69** (261) (2015) 1–6.
23. Y. Xiong and M. Saif, Sliding mode observer for non-linear uncertain systems, *IEEE Trans. on Automatic Control* **46** (12) (2001) 2012–2017.
24. S. Kurode and P. Dixit, Sliding mode control of flexible link manipulator using states and disturbance estimation, *Int. J. Adv. Mech. Syst.* **5** (2) (2013) 129–137.
25. Y. Agrebi, M. Triki, Y. Koubaa et al., Rotor speed estimation for indirect stator flux oriented induction motor drive based on MRAS scheme, *J. Electr. Syst.* **3** (2007) 131–143.
26. W. Chatlatanagulchai, H. C. Nho, and P. Meckl, Robust observer backstepping neural network control of flexible-joint manipulators, in *Proc. Amer. Control Conf.*, (Boston, MA, 2004), pp. 1154–1159.
27. R. Shahnazi and M.-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) (2008) 187–197.
28. 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, 2011), pp. 45–50.
29. J. Chang, Adaptive Second-order Sliding Mode Observer for Quadrotor Attitude Estimation, in *Proc. Amer. Control Conf. (ACC)*, (2016), pp. 2246–2251.
30. L. Saihi, A. Bouhenna, M. Chenafa et al., A Robust Sensorless SMC of PMSM based on Sliding Mode Observer and Extended Kalman Filter, in *Proc. 4th Int. Conf. Electr. Eng. (ICEE)*, (2015), pp. 1–4.
31. F. Chen and M. W. Dunnigan, Comparative study of a sliding-mode observer and Kalman filters for full state estimation in an induction machine, *IEE Proceedings, Electric Power Applications* **149** (1) (2002) 53–64.
32. 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, in *Proc. IEEE IECON94*, (1994), pp. 2149–2154.
33. P. Vas, *Sensorless Vector and Direct Torque Control*,

Monographs in Electrical and Electronic Engineering, Oxford University Press, (1998).

34. S. Bolognani, R. Oboe, and M. Zigliotto, Sensorless full-digital PMSM drive with EKF estimation of speed and rotor position, *IEEE Trans. Ind. Electron.* **46** (1999) 184–191.
35. K. L. Shi, T. F. Chan, Y. K. Wong et al., Speed estimation of an induction motor drive using an optimized extended Kalman filter, *IEEE Trans. on Industrial Electronics* **49** (1) (2002) 124–133.
36. T. Powell, Automated tuning of an extended Kalman filter using the downhill simplex algorithm, *J. Guid. Control and Dyn.* **25** (5) (2002) 901–908.
37. Y. Laamari, K. Chafaa, and B. Athamena, Particle swarm optimization of an extended Kalman filter for speed and rotor flux estimation of an induction motor drive, *Electrical Engineering, Springer-Verlag Berlin Heidelberg* **97** (2) (2014) 129–138.
38. D. Simon, Biogeography-Based Optimization, *IEEE Trans. Evol. Comput.* **12** (6) (2008) 702–713.
39. J.S. Wang and J. D. Song, Migration Ratio Model Analysis of Biogeography-Based Optimization Algorithm and Performance Comparison, *Int. J. Comput. Intellig. Syst.* **9** (3) (2016) 544–558.
40. A. Tayebi, Adaptive iterative learning control for robot manipulators, *Automatica, Elsevier* **40** (7) (2004) 1195–1203.
41. J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, (London: Prentice-Hall, Inc., 1994), ISBN: 0130408905.
42. V. I. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electro-Mechanical Systems*, 2nd ed.; Automation and Control Engineering; CRC Press: Boca Raton, FL, USA, (1999).
43. H. Sira-Ramirez, Structure at infinity, zero dynamics and normal forms of systems undergoing sliding motion, *International Journal of Systems Science* **21** (4) (1990) 665–674.
44. L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-I, *Information Sciences* **8** (3) (1975) 199–249.
45. Q. Liang and J. M. Mendel, Interval type-2 fuzzy logic systems: theory and design, *IEEE Trans. Fuzzy Systems* **8** (5) (2000) 535–550.
46. J.M. Mendel, R.I. John, and F. Liu, Interval type-2 fuzzy logic systems made simple, *IEEE Trans. Fuzzy Systems* **14** (6) (2006) 808–821.
47. S. Zeghlache, K. Kara, and D. Saigaa, Fault Tolerant Control Based on Interval Type-2 Fuzzy Sliding Mode Controller for Coaxial Trirotor Aircraft, *ISA Trans.* **51** (11) (2015) 215–231.

Appendix A

Jacobian matrices F_k , W_k , H_k , V_k for a two-link manipulator

$$F_k = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ f_{21} & f_{22} & f_{23} & f_{24} \\ 0 & 0 & 1 & \Delta t \\ f_{41} & f_{42} & f_{43} & f_{44} \end{bmatrix}, W_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \text{ and } V_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

where

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

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

$$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} \times m_2 C_2 \dot{\theta}_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{\theta}_2(\dot{\theta}_1 l_1 l_{c2} m_2 C_2 + \dot{\theta}_2 l_1 l_{c2} m_2 C_2) + g l_{c2} m_2 S_{12} + \dot{\theta}_1 \dot{\theta}_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 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 \times (l_1 l_{c2} m_2 S_2 \dot{\theta}_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(l_1 l_{c2} m_2 S_2 \dot{\theta}_1^2 + g l_{c2} m_2 C_{12}))(m_2 l_{c2}^2 + l_1 m_2 \times C_2 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 \times 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 l_{c2}^2 + I_2)(\dot{\theta}_2(\dot{\theta}_1 l_1 l_{c2} m_2 S_2 + \dot{\theta}_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{\theta}_1 \dot{\theta}_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 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_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 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).$$

$$\begin{aligned}
 f_{24} &= (\Delta t(m_2 l_{c2}^2 + I_2)(2 \dot{\theta}_1 l_1 l_{c2} m_2 S_2 + 2 \dot{\theta}_2 l_1 l_{c2} m_2 \times \\
 &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_{41} &= -\Delta t(((g S_1(l_1 m_2 + l_{c1} m_1) + g l_{c2} m_2 S_{12})(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)) - \\
 &(g l_{c2} m_2 S_{12}(m_2 l_1^2 + 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{\theta}_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{\theta}_1 l_1 l_{c2} m_2 S_2(m_2 l_1^2 + \\
 &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_{43} &= \Delta t(((g l_{c2} m_2 S_{12} - l_1 l_{c2} m_2 C_2 \dot{\theta}_1^2)(m_2 l_1^2 + \\
 &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)) - ((\dot{\theta}_2(\dot{\theta}_1 l_1 \times l_{c2} m_2 C_2 + \\
 &\dot{\theta}_2 l_1 l_{c2} m_2 C_2) + g l_{c2} m_2 S_{12} + \dot{\theta}_1 \dot{\theta}_2 l_1 l_{c2} m_2 C_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)) + \\
 &(l_1 l_{c2} m_2 S_2(\dot{\theta}_2(\dot{\theta}_1 l_1 l_{c2} m_2 S_2 + \dot{\theta}_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{\theta}_1 \dot{\theta}_2 l_1 l_{c2} m_2 \times \\
 &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_1 l_{c2} m_2 S_2(l_1 l_{c2} m_2 S_2 \dot{\theta}_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_2^2 S_2(m_2 l_{c2}^2 + \\
 &l_1 m_2 C_2 l_{c2} + I_2)(\dot{\theta}_2(\dot{\theta}_1 l_1 l_{c2} m_2 S_2 + \dot{\theta}_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{\theta}_1 \dot{\theta}_2 l_1 l_{c2} m_2 \times \\
 &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 +
 \end{aligned}$$

$$\begin{aligned}
 &(2 l_1^2 l_{c2}^2 m_2^2 C_2^2 S_2(l_1 l_{c2} m_2 S_2 \dot{\theta}_1^2 + g l_{c2} m_2 C_{12}))(m_2 l_1^2 + \\
 &2 m_2 C_2 l_1 l_{c2} + m_1 l_{c1}^2 + m_2 l_{c2}^2 + I_1 + I_2))/((I_1 I_2 + \\
 &l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + \\
 &l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)^2) + (\Delta t l_1 l_{c2} m_2 \tau_1 S_2)/(I_1 \times \\
 &I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + \\
 &l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2) - (2 \Delta t l_1 l_{c2} \times \\
 &m_2 \tau_2 S_2)/(I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + \\
 &I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2) - \\
 &(2 \Delta t l_1^2 l_{c2}^2 m_2^2 \tau_2 C_2 S_2(m_2 l_1^2 + 2 m_2 C_2 l_1 l_{c2} + \\
 &m_1 l_{c1}^2 + m_2 l_{c2}^2 + I_1 + I_2))/((I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + \\
 &I_2 l_1^2 m_2 + I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - \\
 &l_1^2 l_{c2}^2 m_2^2 C_2^2)^2) + (2 \Delta t l_1^2 l_{c2}^2 m_2^2 \tau_1 C_2 S_2(m_2 l_{c2}^2 + \\
 &l_1 m_2 C_2 l_{c2} + I_2))/((I_1 I_2 + l_1^2 l_{c2}^2 m_2^2 + I_2 l_1^2 m_2 + \\
 &I_2 l_{c1}^2 m_1 + I_1 l_{c2}^2 m_2 + l_{c1}^2 l_{c2}^2 m_1 m_2 - l_1^2 l_{c2}^2 m_2^2 C_2^2)^2).
 \end{aligned}$$

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

such as: $S_1 = \sin(\theta_1)$, $S_2 = \sin(\theta_2)$, $S_{12} = \sin(\theta_1 + \theta_2)$, $C_1 = \cos(\theta_1)$, $C_2 = \cos(\theta_2)$, $C_{12} = \cos(\theta_1 + \theta_2)$.

Appendix B

Parameters values used in BBO algorithm

Table B.1. BBO parameters.

Parameter	value
Number of habitats (Population size) NP	20
Maximal number of generation	100
Number of decision variables (SIVs)	6
Immigration and emigration rates E, I	1
Absorption coefficient α	0.9
Probability mutation m_{max}	0.1