Robotic Manipulator State Estimation using Optimized Extended Kalman Filter

Ali Medjghou[1], Yacine Aoun[2], Mouna Ghanai[1], Kheireddine Chafaa[1]
[1](LAAAS) Laboratory, Electronics department, University Batna 2 Mostefa BenBoulaid, Batna, Algeria
[2](VTRS) Laboratory, University of El-Oued, El-Oued, Algeria
medjghou.ali@gmail.com

***Abstract*—This paper presents a novel application of Biogeography-Based Optimization (BBO) to optimize the extended Kalman filter (EKF) in order to achieve high performance estimation of states. The parameters to be optimized in an off-line manner are the covariance matrices of state and measurement noises Q and R, respectively. The optimal values of the above covariance matrices are injected into EKF in an on-line manner to estimate states. The suggested approach is demonstrated using a computer simulation of two-link manipulator. Finally, simulations and comparison with particle swarm optimization (PSO) show the effectiveness of proposed method, and exhibit a more superior performance than its conventional counterpart.**

***Index Terms*—Biogeography-based optimization, particle swarm optimization, extended Kalman filter, states estimation, two-link manipulator.**

I. Introduction

The robust nonlinear control of a robot manipulator 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 robot manipulator 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 Kalman [1] and Luenberger [2], and functional observer [3]. 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 [4], [5], extended Kalman filter [6], [7], sliding mode observer (SMO) [8], model reference adaptive system [9], neural network observer [10] and fuzzy logic observer [11], [12].

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 [13]–[16], 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 [17], 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 [18]. To overcome this problem and to avoid the computational complexity of trial-error method, Shi *et al.* [19], have used genetic algorithms (AGs), downhill simplex algorithm [20], and particle swarm optimization (PSO) [21], which were used to optimize matrices Q and R.

This paper presents 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 biogeography-based optimization algorithm introduced by Simon [22], 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 of state and measurement noises Q and R, respectively, which play an important role in the performances of EKF observer. Interval type-2 fuzzy sliding mode control (IT2FSMC) is used for trajectory tracking.

The paper is structured 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.

II. Extended Kalman Filter

The Kalman filter was developed by R.E. Kalman [1]. 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.

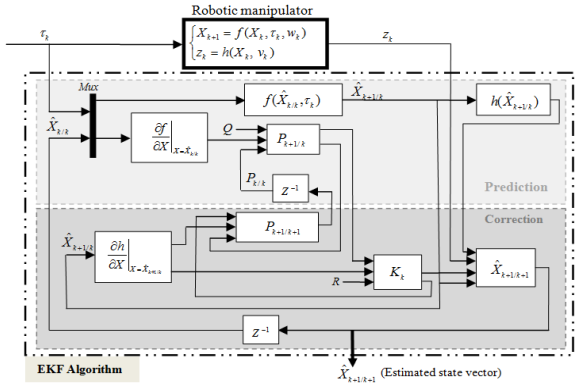

Fig. 1. Extended Kalman filter structure

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. $f(.)$ represents the evolution function of the system, whereas $h(.)$ 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:

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

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

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

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

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

Correcetion:

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

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.

III. 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 [22], [23], 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.
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.

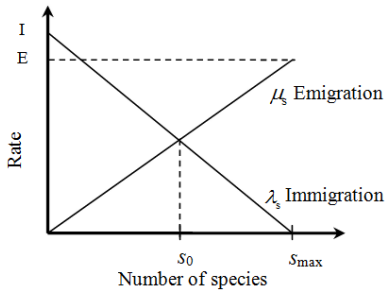

Fig. 2. A Linear model of immigration and emigration rates [22]

The immigration and emigration rates are given as

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

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

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

Algorithm 1 Migration

for $i = 1$ *to* NP **do**
  Use λ_{si} to probabilistically decide whether to immigrate to X_i
  if rand$(0, 1) < \lambda_{si}$ **then**
    for $j = 1$ *to* NP **do**
      Select the emigrating island X_j with probability $\alpha\ \mu_{sj}$
      if rand$(0, 1) < \mu_{sj}$ **then**
        Replace a randomly selected decision variable (SIV) of X_i with its corresponding variable in X_j
      end if
    end for
  end if
end for

where *NP* denote population size, 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.

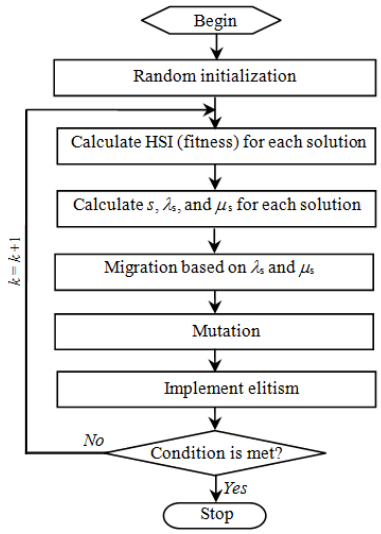

Fig. 3. Algorithm flowchart of BBO

Mutation operator is used to changinge 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:

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

where m_{max} is a user-defined parameter, and $P_{max} = \underset{x}{\text{argmax}}\ P_s$, and $s = 1, \cdots, s_{max}$.

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

IV. Problem Formulation and Proposed Method

A. Problem formulation

The dynamic model of a robotic manipulator in the standard form can be represented by [24]:

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

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 $M_0(\theta), C_0(\theta, \ddot{\theta})$ and $G_0(\theta)$ are nominal terms, whereas $\Delta M(\theta), \Delta C(\theta, \ddot{\theta})$ 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:

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

where $d(\theta, \dot{\theta}, \ddot{\theta}) = -\Delta M(\theta)\ddot{\theta} - \Delta C(\theta, \ddot{\theta})\dot{\theta} - \Delta G(\theta) + \delta \in \mathbb{R}^n$ 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 approach [25], [26], this choice is motivated by its high robustness against uncertainties and external disturbances.

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, Vas [17] were 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 [18]–[20] 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 algorithm [21] (see Figure 4).

B. 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 Vas [17], 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 consumption [18]. 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 III. 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 $s \in [l,\ n]$. 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 (12) 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.

$$MSE = \frac{1}{T}\sum_{k=1}^{T}(z_i(k) - \hat{z}_i(k))^2,\ i = 1, 2 \qquad (12)$$

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 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 (see Figure 4).

V. Simulation and discussions

In order to verify the effectiveness of proposed method, let us consider two degrees of freedom planar manipulator with revolute joints [22] shown in Figure 5.

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 representa-

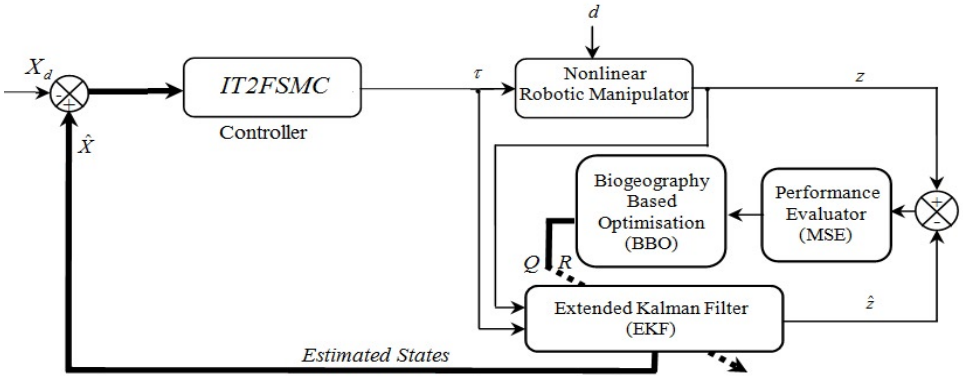

Fig. 4. Block diagram of the control system based on the proposed BBO-EKF observer

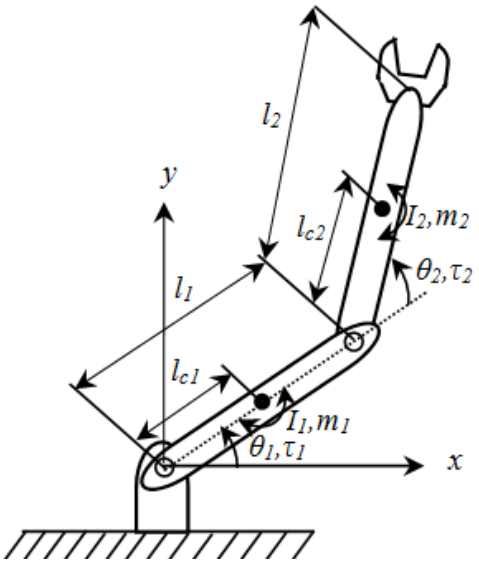

Fig. 5. Two-link robot manipulator

tion:

$$\begin{cases} x_1(k+1) = x_1(k) + \Delta t\ x_2(k+1) + w_1(k) \\ x_2(k+1) = x_2(k) + \Delta t\ \ [f_1(X,k) + \\ \qquad\qquad g_1(X,k)\ \tau(k) + d_1(k)] + w_2(k) \\ x_3(k+1) = x_3(k) + \Delta t\ x_4(k+1) + w_3(k) \\ x_4(k+1) = x_4(k) + \Delta t\ \ [f_2(X,k) + \\ \qquad\qquad g_2(X,k)\ \tau(k) + d_2(k)] + w_4(k) \\ z_1(k) = x_1(k) + v_1(k) \\ z_2(k) = x_3(k) + v_2(k) \end{cases} \tag{13}$$

where $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = M_0^{-1}\left(-C_0[x_2, x_4] - G_0\right)$, $\begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = M_0^{-1}$, $\tau = [\tau_1,\ \tau_2]^T$ 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 $m_1 = m_2$ =1Kg, $l_1 = l_2$ =0.5m, $l_{c1} = l_{c2}$ =0.25m, $I_1 = I_2$ =0.1Kg.m^2, $g = 9.81$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 $\Delta t = 0.001$s.

The desired reference trajectories is chosen as $X_d = [70°, 90°]^T$. 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: $d_1 = 0.3 \times$ rand, and $d_2 = 0.3 \times$ rand $\times$ sin(t). Note that both disturbances first and second sum to d and they will be applied at $t >$ 1s. 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 $P_{0/0} = ones(4, 4)$, 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

$$Q = diag\left(q_{\theta_1}, q_{\dot{\theta}_1}, q_{\theta_2}, q_{\dot{\theta}_2}\right) = \begin{bmatrix} q_{\theta_1} & 0 & 0 & 0 \\ 0 & q_{\dot{\theta}_1} & 0 & 0 \\ 0 & 0 & q_{\theta_2} & 0 \\ 0 & 0 & 0 & q_{\dot{\theta}_2} \end{bmatrix} \tag{14}$$

$$R = diag\left(r_1, r_2\right) = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \tag{15}$$

For comparison purposes, the performance of EKF with diverse compositions of Q and R is evaluated by using the mean-squared-error (12) of the position-estimating response, which is defined as: $MSE = \frac{1}{T}\sum_{k=1}^{T}[\theta_i(k) - \hat{\theta}_i(k)]^2$, $i = 1, 2$.

Table 1 shows typical EKF performance with their corresponding covariance matrices' entries ($q_{\theta_1}, q_{\dot{\theta}_1}$, q_{θ_2}, $q_{\dot{\theta}_2}$, r_1 and r_2) obtained by a trial-error method. It is found that good

TABLE I
EKF PERFORMANCES FOR A TWO-LINK MANIPULATOR USING TRIAL-ERROR ESTIMATIONS.

Case	Q and R entries						MSE (rad)	Estimation quality
	q_{θ_1}	$q_{\dot{\theta}_1}$	q_{θ_2}	$q_{\dot{\theta}_2}$	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

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.

From Table 1, it is clear that the prediction accuracy of EKF is not quite satisfactory 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.

A. Prediction problem

Note that in 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 it is then compared to PSO algorithm.

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 (14) and (15) 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 IV (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 ($\text{MSE}_{trial-error} = 1.4287 \times 10^{-5}$), 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 IV (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.

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 3 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 ($\text{MSE}_{trial-error} = 1.4287 \times 10^{-5}$); therefore, it can be confirmed that 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

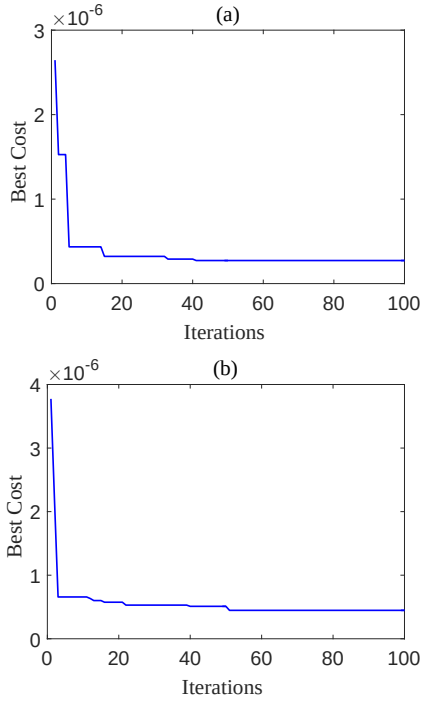

Fig. 6. Mean-square-error comparison versus 100 iterations between (a) BBO-EKF, (b) PSO-EKF

TABLE II
OPTIMIZED EKF PERFORMANCES USING BBO ALGORITHM.

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

TABLE III
OPTIMIZED EKF PERFORMANCES USING PSO ALGORITHM.

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

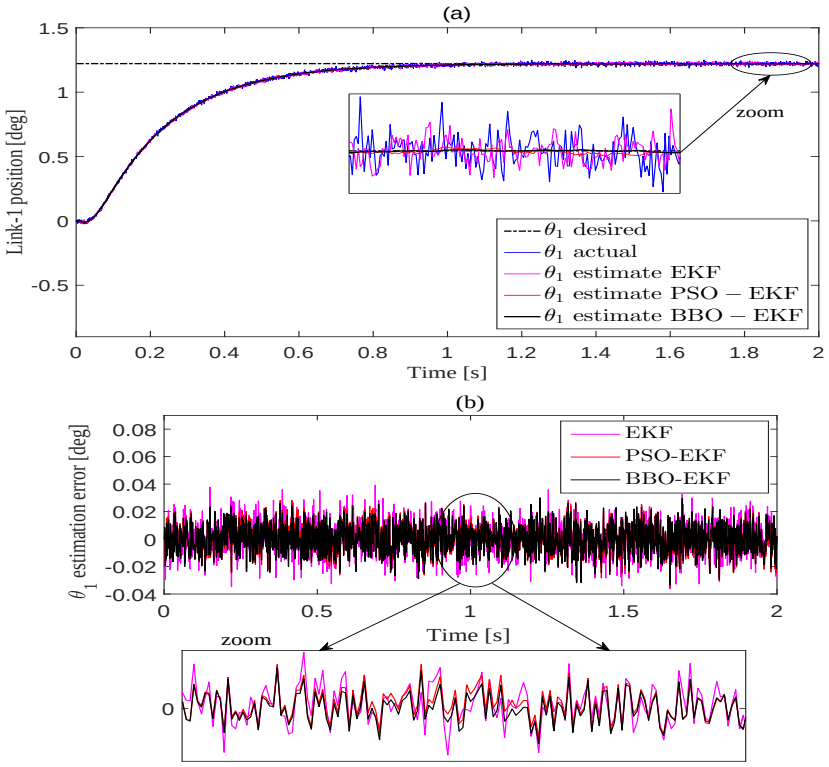

Fig. 7. (a) Desired, actual and estimated position of Link-1 (b) Position estimation errors of Link-1, using IT2FLS SMC for different EKF optimization algorithms

method (initial population, Swarm size, population size).

Figure 6 shows the comparison of objective function (MSE) values for the best solutions obtained through the BBO-EKF and PSO-EKF algorithms versus the 100 iterations, respectively.

The simulation results of applying IT2FSMC with proposed optimized EKF are shown in Figures 7 to 10 were in Figure 7(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 7(b). The corresponding desired, actual and estimated speed and its estimation errors are presented in Figures 8(a) and (b), respectively. In Figure 9(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 9(b). The corresponding desired, actual and estimated speed and its estimation errors are presented in Figures 10(a) and (b), respectively. In all these

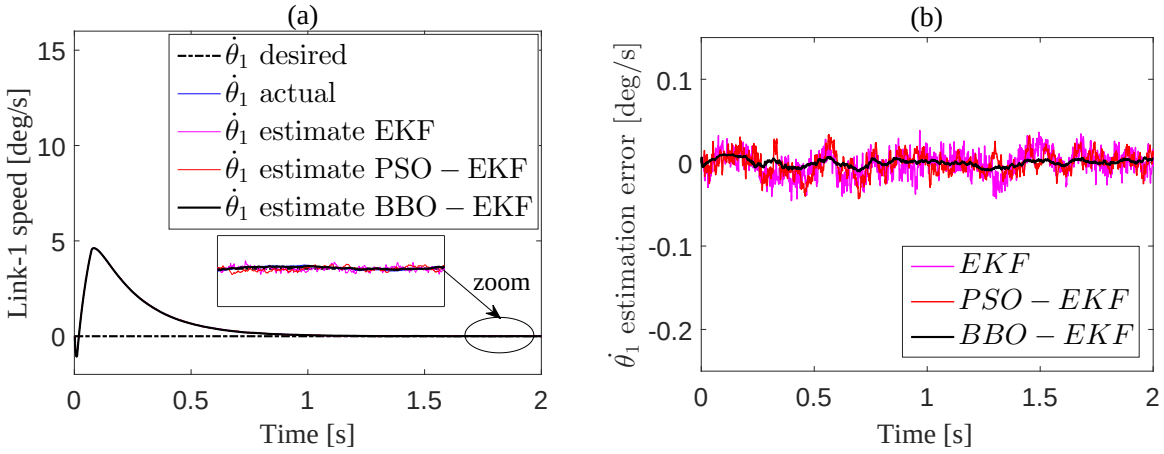

Fig. 8. (a) Desired, actual and estimated speed of Link-1 (b) Speed estimation errors of Link-1, using IT2FLS SMC for different EKF optimization algorithms

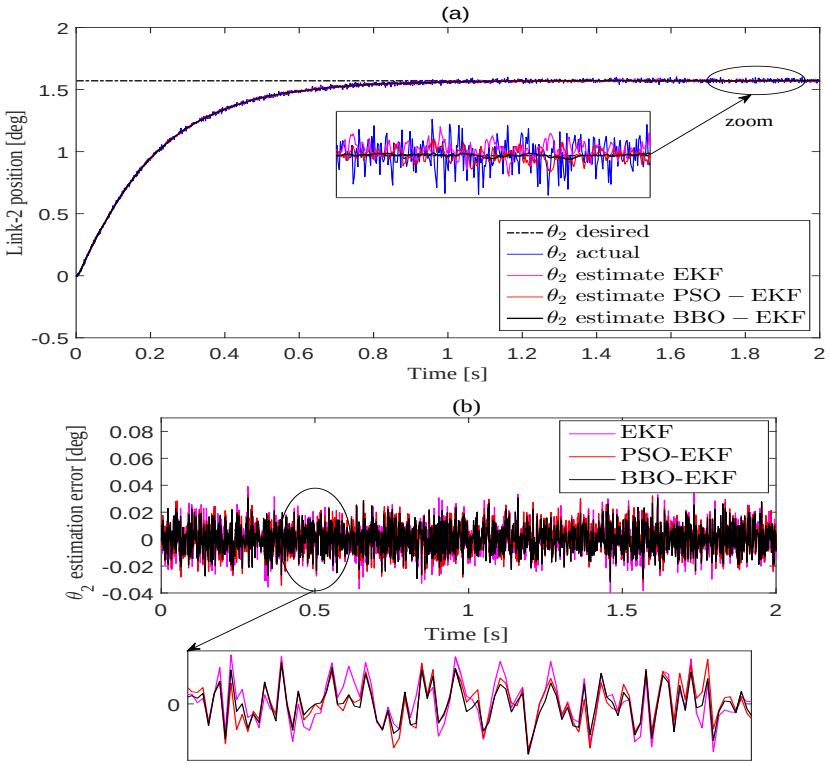

Fig. 9. (a) Desired, actual and estimated position of Link-2 (b) Position estimation errors of Link-2, using IT2FLS SMC for different EKF optimization algorithms

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.

VI. 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 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

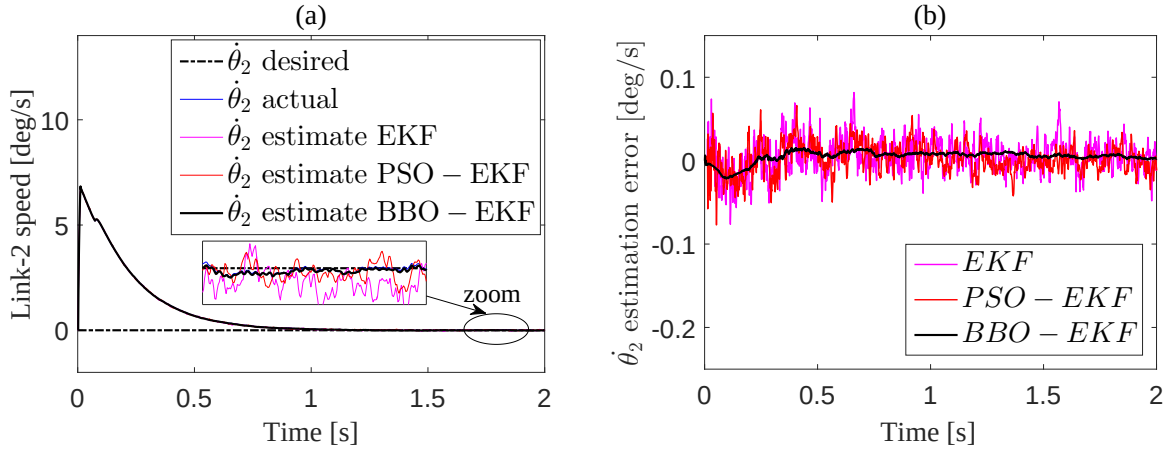

Fig. 10. (a) Desired, actual and estimated speed of Link-2 (b) Speed estimation errors of Link-2, using IT2FLS SMC for different EKF optimization algorithms

using the proposed optimization methods to improve state variables estimation performance of the two-link manipulator and it was concluded that, the performance of BBO-EKF technique was better than PSO-EKF technique. Finally, we can say that the obtained results yield better performance while using proposed approach than traditional one.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," Trans. ASME J. Basic Eng., vol. 82, pp. 35–45, 1960.

[2] D. Luenberger, "Observing the state of linear system," IEEE Trans. Military Electron, vol. 8, pp. 74–90, 1964.

[3] A. E. Baran, E. Golubovic and A. Sabanovic, "Functional observers for motion control systems," AUTOMATIKA, vol. 54, no. 2, pp. 231–241, 2013.

[4] J. Birk and M. Zeitz, "Extended-Luenberger observer for non-linear multivariable systems," Int. J. Control, vol. 47, no. 6, pp. 1823–1836, 1988.

[5] 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, vol. 66, no. 4, pp. 385–414, 2014.

[6] M. Hamraoui, M. Balde and N. Assala, "Observability and observer design for kinetic networks," Int. J. Eng. Syst. Model. Simul., vol. 1, no. 2/3, pp. 79–91, 2009.

[7] L. Liu, B. Qi, S. Cheng et al., "High precision estimation of inertial rotation via the extended Kalman filter," Springer. Eur. Phys. J. D, vol. 69, no. 261, pp. 1–6, 2015.

[8] Y. Xiong and M. Saif, "Sliding mode observer for nonlinear uncertain systems," IEEE Trans. on Automatic Control, vol. 46, no. 12, pp. 2012–2017, 2001.

[9] 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., vol. 3, pp. 131–143, 2007.

[10] 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.

[11] 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, vol. 16, no. 1, pp. 187–197, 2008.

[12] 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.

[13] 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.

[14] 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, vol. 149, no. 1, pp. 53–64, 2002.

[15] J. Chang, "Adaptive Second-order Sliding Mode Observer for Quadrotor Attitude Estimation," in Proc. Amer. Control Conf. (ACC), 2016, pp. 2246–2251.

[16] A. Medjghou, M. Ghanai and K. Chafaa, "A Robust Feedback Linearization Control Framework Using an Optimized Extended Kalman Filter," J. Engineering Science and Technology Review, vol. 10, no. 5, pp. 1–16, 2017.

[17] P. Vas, "Sensorless Vector and Direct Torque Control," Monographs in Electrical and Electronic Engineering, Oxford University Press, 1998.

[18] S. Bolognani, R. Oboe and M. Zigliotto, "Sensorless full-digital PMSM drive with EKF estimation of speed and rotor position," IEEE Trans. Ind. Electron., vol. 46, pp. 184–191, 1999.

[19] 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, vol. 49, no. 1, pp. 124–133, 2002.

[20] T. Powell, "Automated tuning of an extended Kalman filter using the downhill simplex algorithm," J. Guid. Control and Dyn., vol. 25, no. 5, pp. 901–908, 2002.

[21] 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, vol. 97, no. 2, pp. 129–138, 2014.

[22] D. Simon, "Biogeography-Based Optimization," IEEE Trans. Evol. Comput., vol. 12, no. 6, pp. 702–713, 2008.

[23] J.S. Wang and J. D. Song, "Migration Ratio Model Analysis of Biogeography-Based Optimization Algorithm and Performance Comparison," Int. J. Comput. Intellig. Syst., vol. 9, no. 3, pp. 544–558, 2016.

[24] A. Tayebi, "Adaptive iterative learning control for robot manipulators," Automatica, Elsevier, vol. 40, no. 7, pp. 1195–1203, 2004.

[25] J-J. E. Slotine and W. Li, "Applied Nonlinear Control," London: Prentice-Hall, Inc., 1991, ISBN: 0130408905.

[26] A. Medjghou, M. Ghanai and K. Chafaa, "BBO optimization of an EKF for interval type-2 fuzzy sliding mode control," International Journal of Computational Intelligence Systems, vol. 11, iss. 1, pp. 770–789, 2018.

APPENDIX A
JACOBIAN MATRICES F_k, W_k, H_k, V_k FOR A TWO-LINK MANIPULATOR

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

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 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 + (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_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$.

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

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

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

$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 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 + (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_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$.

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

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 IV
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