A Heuristic Filter Based on Firefly Algorithm for Nonlinear State Estimation

Hadi Nobahari, Mohsen Raoufi, Alireza Sharifi
Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran
E-mail: nobahari@sharif.edu, {mohsen_raoufi, alireza_sharifi}@ae.sharif.edu

***Abstract*—A new heuristic filter, called firefly filter, is proposed for state estimation of nonlinear stochastic systems. The new filter formulates the state estimation problem as a stochastic dynamic optimization and utilizes the firefly optimization algorithm to find and track the best estimation. The fireflies search the state space dynamically and are attracted to one other based on the perceived brightness. The performance of the proposed filter is evaluated for a set of benchmarks and the results are compared with the well-known filters like extended Kalman filter and particle filter, showing improvements in terms of estimation accuracy.**

Keywords-component; Heuristic Filter; Nonlinear State Estimation; Nonlinear Stochastic System; Firefly Algorithm;

I. INTRODUCTION

In many engineering applications, one needs to estimate the states of a nonlinear dynamic system. There are many widely applied state estimation algorithms for nonlinear systems. Such algorithms are called nonlinear filters [1,2]. Nonlinear filters, such as Extended Kalman Filter (EKF) [3], Unscented Kalman Filter (UKF) [4] and Particle Filter (PF) [5], are used to estimate the states of a nonlinear dynamic system when either the system or the measurement model is nonlinear.

Analytical approximation and states sampling are two common approaches in nonlinear filtering. In the first approach, the nonlinear functions of the mathematical model are linearized and then a linear filter such as Kalman Filter (KF) [6] is utilized as well. EKF is an example of filters, work based on analytical approximation. Unlike to EKF, UKF is a sample based filter. It does not approximate the nonlinear mathematical model. Instead, it approximates the posterior PDF by a set of deterministically chosen samples. UKF is also referred to as a linear regression Kalman filter because it is based on statistical linearization rather than analytical ones [2].

Sample based filters can be categorized to mathematical and heuristic approaches [7]. UKF can be taken a mathematical sample based filter into account since it uses a deterministic sampling process, the general estimation mathematics and the mathematical operators such as unscented transform. In comparison, there are several sample based filters that utilize heuristic algorithms to sample the particles and to improve the position of them. These filters are called heuristic filters [7].

PF is an example of heuristic filters. It works based on point mass (or particle) representation of the probability densities [8]. Unlike to UKF, PF represents the required posterior PDF by a set of random samples instead of deterministic ones. Also, it uses a resampling procedure to reduce the degeneracy of particle set. Some versions of PF adapt Sequential Importance Sampling (SIS) algorithm to calculate the posterior distribution using the importance sampling density such as Auxiliary Sampling Importance Resampling (ASIR) filter [9] and Sampling Importance Resampling (SIR) filter [10,11].

Since PF algorithms are suboptimal estimators, they have some accuracy problems such as particle impoverishment and sample size dependency. Particle impoverishment happens when the likelihood is so narrow that the overlapping region of likelihood and prior distribution is quite small [2,12] and no particle lies within the region of likelihood probably. Therefore, the weights of most particles become relatively small and the efficiency of them are decreased, the result of which is the degradation of the estimation accuracy. Also, the sample size has a great effect on the performance of PF. If the sample size is relatively small, then the proper distribution of particles around the true states may not occur.

There have been some systematic approaches, proposed recently to solve limitations of PF. The first approach improves resampling such as Binary Search [10], Systematic Resampling [13] and Residual Resampling [8]. However, these methods are not ideal because the particles with high weights are statistically selected many times. This gradually leads to many repeated points and consequently the diversity among the particles is lost [2].

The second approach improves the prior distribution with modified PF algorithms such as Unscented Particle Filters (UPF) [14], Extended Kalman Particle Filters (EKPF) [13]. Also, in this approach, different ideas of heuristic optimization algorithm can be extended and modified to solve the estimation problem. Genetic Algorithm (GA) has been combined with PF to increase the diversity of samples after resampling[15,16]. Simulated Annealing (SA) has also been introduced into PF to improve its performance [17]. Moreover, a local search method has been inserted into PF to reduce the sample size and improve the efficiency [18].

Another approach is the use of swarm intelligence techniques to improve the sampling process. Such filters are called swam filters [19]. In this approach, the state estimation problem can be formulated as a stochastic dynamic optimization problem. Therefore, swarm filters utilize swarm intelligence techniques to solve this problem [20,21,22,23,7]. Particle Swarm Optimized Particle Filter (PSOPF) [20] merges Particle Swarm Optimization (PSO) into PF to optimize sampling step of Generic PF (GPF). In other work [22], Adaptive Particle Swarm Filter (APSF) has been proposed for state estimation of nonlinear dynamic systems based on PSO and differential evolution (DE).

Also, Ant Colony Optimization Assisted Particle Filter (PF_{ACO}) [22] incorporates Ant Colony Optimization (ACO) into PF to optimize sampling step of Generic PF (GPF). Particle Filter with Ant Colony for Continuous Domains [22,23] incorporates Ant Colony Optimization for Continuous Domains (ACO_R) [24] into PF to optimize sampling process of PF. Moreover, Continuous Ant Colony Filter (CACF) [7] is a heuristic filter, based on Continuous Ant Colony System (CACS) [25]. It utilizes a colony of moving ants, the average positions of which is returned as the current estimation.

In this study, a new swarm filter is proposed for nonlinear system state estimation, based on the swarm intelligence technique known as Firefly Algorithm (FA) [26,27]. The proposed filter is called Firefly Filter (FF). It utilized a set of moving fireflies to perform an intelligent search in the state space, the average positions of which is returned as the current estimation. In this filter, the estimation of the current states is formulated as a stochastic dynamic optimization problem and an optimization algorithm, based on FA, is utilized to iteratively find and track the best estimation. The performance of the proposed filter is investigated for a set of benchmarks, taken from the literature, to compare its results with those of EKF, as a mathematical nonlinear approach and PF, as a heuristic approach. The small estimation errors proved the good performance of FF in nonlinear estimation. Moreover, the overall results show that FF can properly compete with these well-known filters.

This paper is organized as follows: In section II, a state estimation problem is formulated. A detailed description of the new estimation algorithm is devoted in section III. Numerical results are provided in section IV. Finally, a conclusion is made in section V.

II. Estimation Problem

The problem is to estimate the states of a nonlinear stochastic system. Discrete-time state space approach is utilized to model the evolution of the system and the noisy measurements. The states are assumed to be evolved according to the following stochastic model:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \boldsymbol{\omega}_{k-1}) \qquad (1)$$

where $\mathbf{f}_k$ is a known, possibly nonlinear function of the state vector $\mathbf{x}_{k-1}$, $\boldsymbol{\omega}_{k-1}$ represents the process noise, and k is the time counter. The objective of a nonlinear filter is to recursively estimate $\mathbf{x}_k$ from the available measurements, $\mathbf{z}_k$. The measurements are related to the states via the measurement equation, stated as follows:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k) \qquad (2)$$

where $\mathbf{h}_k$ is a known, possibly nonlinear function and $\mathbf{v}_k$ is the measurement noise. The noise sequences, $\boldsymbol{\omega}_{k-1}$ and $\mathbf{v}_k$ are mutually independent and are assumed to have the uniform or multimodal distribution with known covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. A graphical illustration of the evolution and the measurement models is depicted as in “Fig. 1”.

III. Firefly Filter

In this section, the new heuristic filter, called Firefly Filter (FF), is introduced as a tool for nonlinear systems state estimation. FF models the state estimation problem as a stochastic dynamic optimization problem and utilizes an optimization scheme, based on Firefly Algorithm (FA) [28,29] to solve this problem. Fireflies interact with each other while they are learning from the swarm experiences, and looking for the best estimation. “Fig. 2” shows the general iterative structure of FF. A high-level description of the sequential steps is shown in this figure. The parameters of FF and the initial position of fireflies are set during the initialization, as discussed in section A.

FF has two loops. The main outer loop iterates every time a new measurement is entered. The inner loop iterates to find the best estimation of the current states, corresponding to the entered measurement. At first, the inner loop propagates the initial distribution of fireflies. Then, the output, estimated by each firefly, is made. The estimated outputs are compared with the real measurement and the cost of each firefly is evaluated. Fireflies use their experiences to update their position in the state space. As in FA [30], each firefly moves to brighter/attractive firefly wherein firefly’s brightness is characterized by its cost function. The inner loop is terminated after a predefined number of iterations. Finally, the current state estimation is made using a mean operator. “Fig. 3” shows pseudo-code of FF. In the following subsections, these steps are discussed in detail.

A. Initialization

The new algorithm has some control parameters that must be set before the execution of the algorithm. The inner loop is terminated after q iterations. Moreover, the initial position of fireflies $\mathbf{x}_0^j$, (j=1,…,N), is initialized using a uniform random generator.

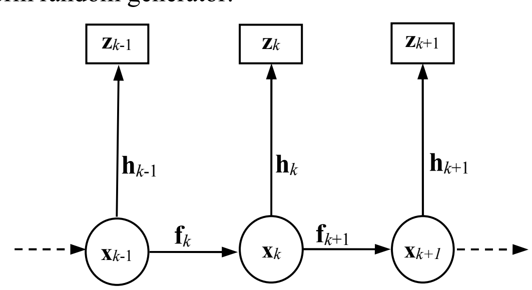

Figure 1. Process and measurement model of a dynamic system [19]

B. Propagation of Fireflies' Location

At the beginning of the i-th iteration of the inner loop, the position of firefly j at time k-1, defined as $\mathbf{x}_{k-1}^{i,j}$, is propagated as follows:

$$\hat{\mathbf{x}}_k^{i,j} = \mathbf{f}_k(\mathbf{x}_{k-1}^{i,j}, \boldsymbol{\omega}_{k-1}^{i,j}) \quad (3)$$

C. Measurement Update

In this step, the current output, ($\hat{\mathbf{z}}_k^{i,j}$), estimated in iteration i by firefly j at time k, is calculated as follows:

$$\hat{\mathbf{z}}_k^{i,j} = \mathbf{h}_k(\hat{\mathbf{x}}_k^{i,j}) \quad (4)$$

D. Computing cost function

After the propagation and measurement update, the cost function of each firefly, $c_k^{i,j}$, is calculated as the square error between the real measurement, $\mathbf{z}_k$, and the j-th estimated output, $\hat{\mathbf{z}}_k^{i,j}$. Therefore, the cost, assigned in iteration i to firefly j at time k, is calculated as follows:

$$c_k^{i,j} = (\mathbf{z}_k - \hat{\mathbf{z}}_k^{i,j})^{\mathrm{T}}(\mathbf{z}_k - \hat{\mathbf{z}}_k^{i,j}) \quad (5)$$

In this way, the cost function is calculated at different points of the state space and some knowledge about the problem is acquired, used later to move the fireflies from their current position in the state space toward the minimum cost destinations.

E. Attraction to other fireflies

FF utilizes a set of moving fireflies to perform an intelligent search in the state space, looking for the best estimation. Each firefly is attracted only to those that are brighter than itself. The brightness of a firefly is determined by the landscape of the cost function, explained in D. The attractiveness of a firefly is directly proportional to its brightness, which exponentially decreases with distance r between any two fireflies j and l as follow:

$$\beta = \beta_0 e^{-\gamma(\mathbf{r}_k^{i,jl})^2} \quad (6)$$

where β_0 is attractiveness at $\mathbf{r}=0$ and $\mathbf{r}_k^{i,jl}$ is the distance between two fireflies j and l at positions $\hat{\mathbf{x}}_k^{i,j}$ and $\hat{\mathbf{x}}_k^{i,l}$ can be defined as follows:

$$\mathbf{r}_k^{i,jl} = \left\| \hat{\mathbf{x}}_k^{i,j} - \hat{\mathbf{x}}_k^{i,l} \right\|_2 \quad (7)$$

Also, γ is the light absorption coefficient. Each firefly compares its brightness with that of other fireflies, one at a time. If l-th firefly is brighter than j, then j will make a move that includes a random component, and a component that is directed toward $\hat{\mathbf{x}}_k^{i,j}$, as in FA. The traversal of a firefly towards other brighter fireflies is given by

$$\hat{\mathbf{x}}_k^{i+1,j} = \hat{\mathbf{x}}_k^{i,j} + \beta\,\mathbf{r}_k^{i,jl} + \alpha\,\boldsymbol{\varepsilon}^i \quad (8)$$

where α is the randomness parameter and $\boldsymbol{\varepsilon}$ is a vector of zero mean random numbers with the Gaussian probability distribution, in the form $N(0,1)$. In FF, the parameters β_0 and α determine the tradeoff between exploitation (attraction to other fireflies) and exploration (random search), as FA.

F. Stopping condition

FF has two loops, each with its own specific stopping condition. The inner loop stops when the maximum number of iterations (q) is reached. The outer loop terminates when the measurements are finished.

G. State estimation

After termination of the inner loop, the states are estimated based on the average position of top fireflies as follows:

$$\hat{\mathbf{x}}_k = \frac{1}{\mathrm{N}_t}\sum_{j=1}^{\mathrm{N}_t} \hat{\mathbf{x}}_k^{q,j} \quad (9)$$

where N_t denotes the number of top fireflies.

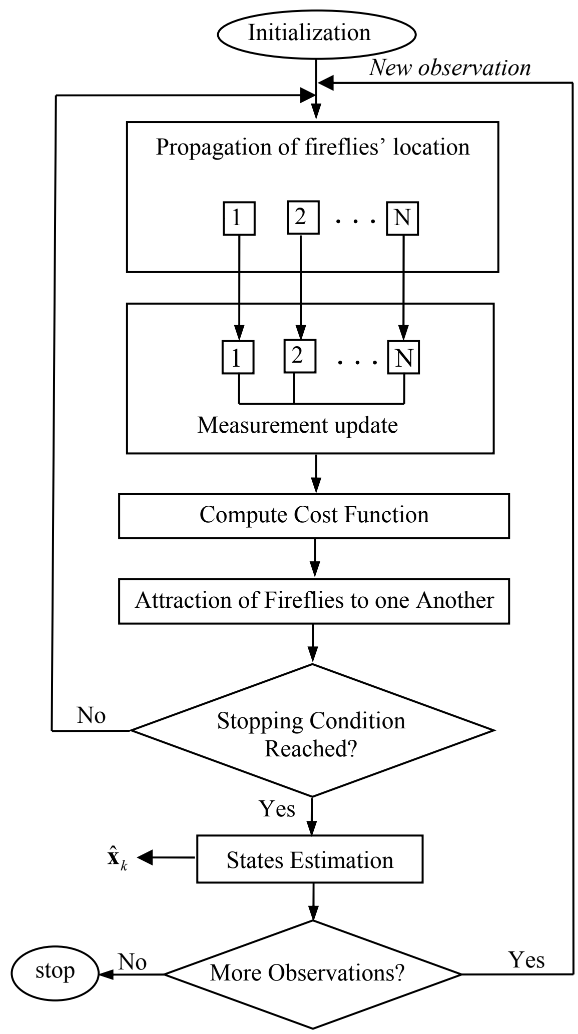

Figure 2. Firefly Filter (FF) Algorithm

Initialize the parameters

Initialize a random population { $\hat{x}_0^j$ } for $j\in[1,N]$

While (Measurements are available)

For i=1 to q

If $i \neq 1$

For each individual $\hat{x}_k^j$

For each individual $\hat{x}_k^l \neq \hat{x}_k^j$

If c($\hat{x}_k^l$) < c($\hat{x}_k^j$)

Move firefly l towards j as shown in "(8)"

End if

Next $\hat{x}_k^l$

Next $\hat{x}_k^j$

End if

For j =1 to N

propagate each firefly as shown in "(3)"

compute cost function as shown in "(4)"

Next firefly j

Next Iteration i

Sort fireflies according to cost

Mean the position of top fireflies as shown in "(9)"

Next step time k

Figure 3. Firefly filter pseudo-code

IV. RESULTS AND DISCUSSION

This section presents numerical results, obtained from the simulation of FF in state estimation. First, the performance of the new filter is investigated for a benchmark, taken from the literature. This study is intended to provide a comparison of the proposed estimation method with more established approaches. Then, the performance of the proposed filter is evaluated in the presence of modeling error.

A. A highly nonlinear scalar system

In this case, a nonlinear single variable economic model [22], defined by (10) and (11), is employed to test the performance of FF and compare it with that of EKF, GPF, and CACF. The high degree of nonlinearity in both the process and measurements makes this a difficult state estimation problem for a KF. This stochastic system can be expressed as

$$x_k = 1 + \sin(4\times10^{-2}\pi k) + 0.5\, x_{k-1} + \omega_{k-1} \qquad (10)$$

where ω_{k-1} is zero-mean Gaussian white noise sequence with variance equal to 1×10^{-5}. Also, the measurement model can be written as

$$z_k = \begin{cases} \dfrac{x_k^2}{5} + v_k & k \le 30 \\ -2 + \dfrac{x_k}{2} + v_k & k > 30 \end{cases} \qquad (11)$$

where v_k stands for Gamma distribution with the parameters k and θ, are equal to 7 and 2, respectively [22]. The initial state and the initial estimation covariance for EKF are taken as x_0=0 and P_0=100, respectively. Also, 200 particles in PF are considered. The tuned parameters of FF is presented in TABLE I. To make the results comparable with those of [22], the simulations are performed from t = 1 to 60 and the average performance, obtained for 30 different runs, are compared. The performance of a sample output of FF in the estimation of state (x_k) is shown in "Fig. 4". "Fig. 4"(a) shows the true and estimated state and "Fig. 4"(b) represents the estimation error. These figures show that the output of FF well matches with the true output. Also, the box plot is shown in "Figure 5. and the mean Root Mean Square Error (RMSE) is presented in TABLE II. It can be observed that FF produces better results than EKF, GPF, and CACF.

B. Mismodeling

In the previous section, the model of the system was assumed to be precisely known. But, there may be low confidence in the accuracy of the model. To investigate the FF performance under such mismodeling, the model used by fireflies was assumed to have some error. Therefore, the true system is given as the following two-state model

$$\begin{cases} \mathbf{x}_k(1) = \mathbf{x}_{k-1}(1) + 5\sin(0.1\pi k) + \mathbf{x}_{k-1}(2) \\ \mathbf{x}_k(2) = \mathbf{x}_{k-1}(2) \end{cases} \qquad (12)$$

$$z_k = \mathbf{x}_k(1) + v_k \qquad (13)$$

However, the assumed model, used by fireflies, is given as follows:

$$x_k = x_{k-1} \qquad (14)$$

$$z_k = x_k + v_k \qquad (15)$$

where v_k is a zero-mean Gaussian white noise sequence with variance equal to 1. The tuned parameters of FF in the presence of modeling error is presented in TABLE III. The performance of FF in state estimation is shown in "Fig. 6". "Fig. 6"(a) shows the true and estimated state and "Fig. 6"(b) represents the estimation error. This figure indicates that FF accurately works in the presence of modeling error.

TABLE I. PARAMETERS OF FF FOR A HIGHLY NONLINEAR SCALAR SYSTEM

Parameter	Value	Description
N	40	Number of fireflies
q	64	Maximum number of iterations
N_t	39	Number of top fireflies
β_0	0.094	Attractiveness of a firefly
γ	40.933	Light absorption coefficient
α	0.0001	Randomness parameter

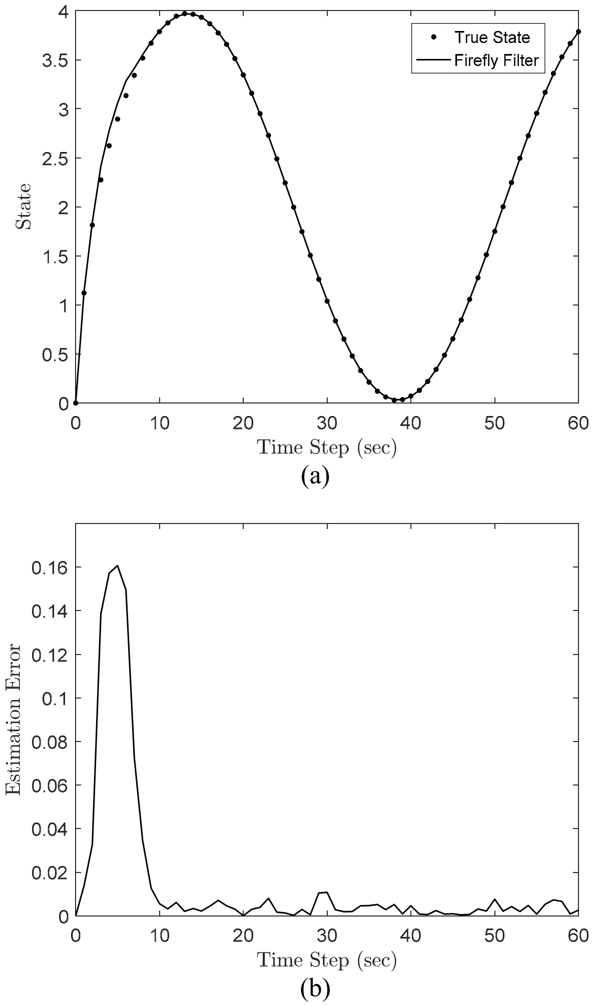

Figure 4. The performance of firefly filter for a highly nonlinear scalar system: (a) True versus estimated state. (b) Estimation error.

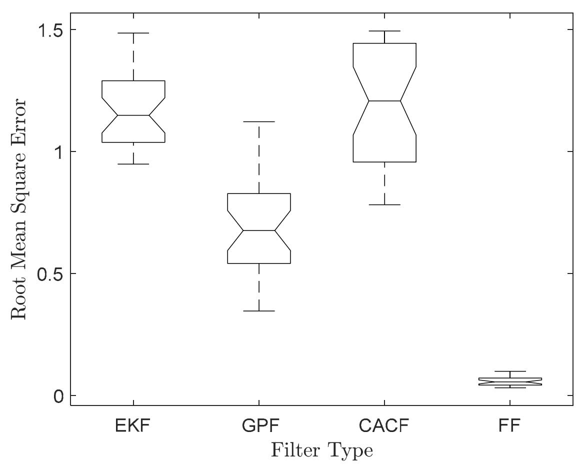

Figure 5. Boxplot of RMSE for EKF, GPF, CACF, and FF.

TABLE II. COMPARISON OF FF WITH EKF, GPF AND CACF

Filter	Mean RMSE
EKF	1.1644
GPF	0.6799
CACF	1.1924
FF	0.0582

TABLE III. PARAMETERS OF FF IN THE PRESENCE OF MODELING ERROR

Parameter	Value	Description
N	40	Number of Fireflies
q	64	Maximum number of iterations
N_t	4	Number of top fireflies
β_0	3.62	Attractiveness of a firefly
γ	0.0933	Light absorption coefficient
α	0.0001	Randomness

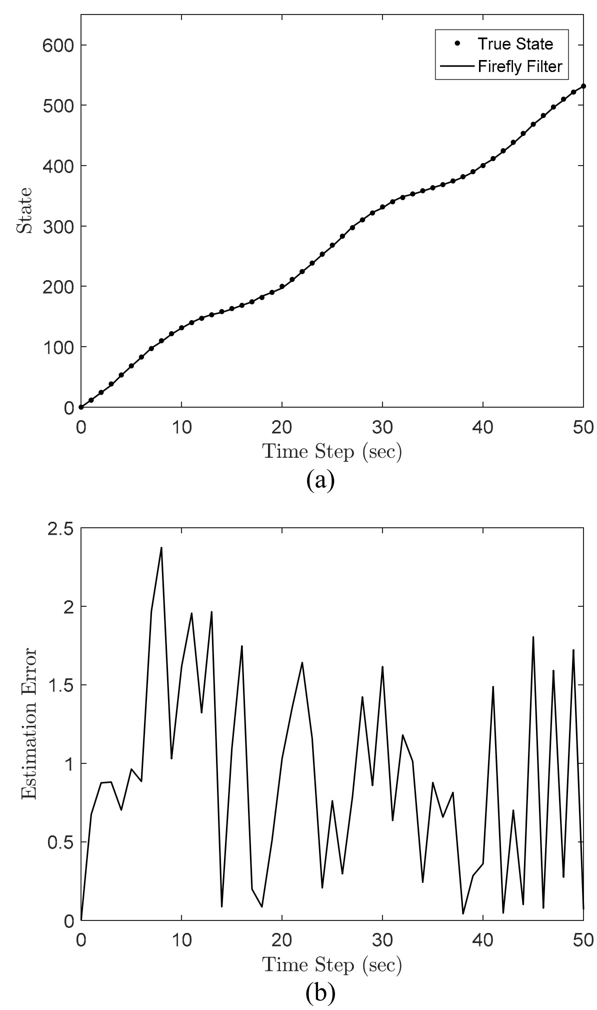

Figure 6. The performance of FF in the presence of modeling error: (a) True versus estimated state. (b) Estimation error.

V. Conclusion

In this paper, a novel heuristic filter was proposed for state estimation of nonlinear stochastic systems. The proposed filter, called FF, models the state estimation problem as a stochastic dynamic optimization problem and utilizes an optimization scheme, based on FA, to solve this problem. The performance of FF was successfully tested over a nonlinear benchmark and in the presence of the modeling error. The small estimation errors show the good performance of FF in nonlinear state estimation. Moreover, the overall results show that FF can properly compete with well-known filters such as EKF and PF.

References

[1] D. Simon, Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons, 2006.

[2] B. Ristic, S. Arulampalam, and N. Gordon, Beyond the Kalman filter: Particle filters for tracking applications: Artech house Boston, 2004.

[3] A. H. Jazwinski, Stochastic Processes and Filtering Theory: Academic Press, 1970.

[4] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in AeroSense'97, pp. 182-193, 1997

[5] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," IEE Proceedings-Radar, Sonar and Navigation, vol. 146, pp. 2-7, 1999.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," Journal of basic Engineering, vol. 82, no. 1, pp. 35-45, 1960.

[7] H. Nobahari and A. Sharifi, "A novel heuristic filter based on ant colony optimization for non-linear systems state estimation," in Computational Intelligence and Intelligent Systems, ed: Springer, pp. 20-29, 2012.

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," IEEE Transactions on signal processing, vol. 50, pp. 174-188, 2002.

[9] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," Journal of the American statistical association, vol. 94, pp. 590-599, 1999.

[10] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in IEE Proceedings F-Radar and Signal Processing, pp. 107-113, 1993.

[11] S. McGinnity and G. W. Irwin, "Multiple model bootstrap filter for maneuvering target tracking," IEEE Transactions on Aerospace and Electronic Systems, vol. 36, pp. 1006-1012, 2000.

[12] W. Zang, Z. Shi, S. Du, and K. Chen, "Novel roughening method for reentry vehicle tracking using particle filter," Journal of Electromagnetic Waves and Applications, vol. 21, pp. 1969-1981, 2007.

[13] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in Sequential Monte Carlo methods in practice, ed: Springer, pp. 3-14, 2001.

[14] R. Van Der Merwe N. De Freitas, A. Doucet and E. Wan, "The Unscented Particle Filter", Cambridge University Engineering Department, Cambridge, 2000.

[15] T. Higuchi, "Monte Carlo filter using the genetic algorithm operators," Journal of Statistical Computation and Simulation, vol. 59, pp. 1-23, 1997.

[16] S. Park, J. P. Hwang, E. Kim, and H.-J. Kang, "A new evolutionary particle filter for the prevention of sample impoverishment," IEEE Transactions on Evolutionary Computation, vol. 13, pp. 801-809, 2009.

[17] T. Clapp, "Statistical methods for the processing of communications data," University of Cambridge, 2001.

[18] P. Torma and C. Szepesvári, "LS-N-IPS: an improvement of particle filters by means of local search," in Proc. Non-Linear Control Systems (NOLCOS'01) St. Petersburg, Russia, pp. 715-719, 2001.

[19] H. Nobahari, A. Sharifi, and H. MohammadKarimi, "Swarm intelligence techniques applied to nonlinear systems state estimation," in Advances in Heuristic Signal Processing and Applications, ed: Springer, pp. 219-241, 2013.

[20] G. Tong, Z. Fang, and X. Xu, "A particle swarm optimized particle filter for nonlinear system state estimation," in Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, 2006, pp. 438-442.

[21] M. Kiani and S. H. Pourtakdoust, "State estimation of nonlinear dynamic systems using weighted variance-based adaptive particle swarm optimization," Applied Soft Computing, vol. 34, pp. 1-17, 2015.

[22] J. Zhong, Y.-f. Fung, and M. Dai, "A biologically inspired improvement strategy for particle filter: Ant colony optimization assisted particle filter," International Journal of Control, Automation and Systems, vol. 8, pp. 519-526, 2010.

[23] Y. Yu and X. Zheng, "Particle filter with ant colony optimization for frequency offset estimation in OFDM systems with unknown noise distribution," Signal Processing, vol. 91, pp. 1339-1342, 2011.

[24] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," European journal of operational research, vol. 185, pp. 1155-1173, 2008.

[25] S. H. Pourtakdoust and H. Nobahari, "An extension of ant colony system to continuous optimization problems," in International Workshop on Ant Colony Optimization and Swarm Intelligence, 2004, pp. 294-301.

[26] X.-S. Yang, "Firefly algorithms for multimodal optimization," in International Symposium on Stochastic Algorithms, 2009, pp. 169-178.

[27] I. Fister, X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," Swarm and Evolutionary Computation, vol. 13, pp. 34-46, 2013.

[28] I. Fister, M. Perc, and S. M. Kamal, "A review of chaos-based firefly algorithms: perspectives and research challenges," Applied Mathematics and Computation, vol. 252, pp. 155-165, 2015.

[29] D. Simon, Evolutionary optimization algorithms: John Wiley & Sons, 2013.

[30] M. N. Murty, X. He, R. R. Chillarige, and P. Weng, Multi-disciplinary Trends in Artificial Intelligence: 8th International Workshop, MIWAI 2014, Bangalore, India, December 8-10, 2014, Proceedings vol. 8875: Springer, 2014.