Applied Soft Computing 34 (2015) 1–17

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

State estimation of nonlinear dynamic systems using weighted variance-based adaptive particle swarm optimization

M. Kiani, Seid H. Pourtakdoust*

Center for Research and Development in Space Science and Technology, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 9 February 2015
Accepted 13 April 2015
Available online 7 May 2015

Keywords:
Heuristic filter
Adaptive population size
Particle swarm optimization
Differential evolution

ABSTRACT

New heuristic filters are proposed for state estimation of nonlinear dynamic systems based on particle swarm optimization (PSO) and differential evolution (DE). The methodology converts state estimation problem into dynamic optimization to find the best estimate recursively. In the proposed strategy the particle number is adaptively set based on the weighted variance of the particles. To have a filter with minimal parameter settings, PSO with exponential distribution (PSO-E) is selected in conjunction with jDE to self-adapt the other control parameters. The performance of the proposed adaptive evolutionary algorithms i.e. adaptive PSO-E, adaptive DE and adaptive jDE is studied through a comparative study on a suite of well-known uni- and multi-modal benchmark functions. The results indicate an improved performance of the adaptive algorithms relative to original simple versions. Further, the performance of the proposed heuristic filters generally called adaptive particle swarm filters (APSF) or adaptive differential evolution filters (ADEF) are evaluated using different linear (nonlinear)/Gaussian (non-Gaussian) test systems. Comparison of the results to those of the extended Kalman filter, unscented Kalman filter, and particle filter indicate that the adopted strategy fulfills the essential requirements of accuracy for nonlinear state estimation.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Successful control of many closed loop dynamic systems relies on exact and complete knowledge of the system states. Lack of proper information in the feedback signals results in ineffective control law. Unfortunately, for many physical systems not all states are measurable due to restrictions on the utilized sensors such as cost and/or weight. Therefore, state estimation or in other words filtering out the bad information from the available noise-corrupted measurements is an essential task for all controlled dynamic systems. On-line state estimation enhances system security, data accuracy and reduces the cost of the measurement package due to requiring less measurement sensors.

State estimation problem has been in the center of attention for many years. Dynamic system state estimation has been usually formulated as a weighted least squares problem. This widespread method is considered as a batch estimator that uses the complete history of measurements to estimate the unknown states. In comparison to the batch strategies, recursive filters receive and process measurements sequentially and as such are regarded more important and useful. Every progressive step of a recursive filter consists of two stages: prediction and data assimilation. Prediction propagates the system states from a time step to the next one ahead, while data assimilation is employed to refine the predicted states utilizing the new measurements. Fortunately, there exist various schemes in the literature for state estimation of nonlinear dynamic systems. Although, Extended Kalman filter (EKF) [1] is the simplest and the most utilized nonlinear filter applied to different systems, it suffers from two important drawbacks. First, it is prone to divergence as it is based on linearization of nonlinear dynamics and measurement functions. Second, it is only suitable for systems with Gaussian noise models. Other nonlinear filters have been presented to remedy these weaknesses. Unscented Kalman filter (UKF) [2], the next well-known nonlinear filter, originates from the idea that approximating a Gaussian distribution is easier than approximating an arbitrary nonlinear function. In this regard, UKF removes the need for linearization, while the assumption of Gaussian distribution is still kept. Particle filter (PF) family [3] is suggested to complement UKF and to cope with its shortcomings. In contrast to the EKF, as an analytical approach for state estimation, UKF and PF are both sampling approaches to this aim. However, there exists a fundamental difference between the two latter. UKF is based on a

* Corresponding author. Tel.: +98 21 66164610; fax: +98 21 66022731.
E-mail address: pourtak@sharif.edu (S.H. Pourtakdoust).

http://dx.doi.org/10.1016/j.asoc.2015.04.029
1568-4946/© 2015 Elsevier B.V. All rights reserved.

deterministic sampling methodology, while samples are stochastically generated in the PF. Therefore, PF can be considered as a member of the newly defined class of heuristic filters.

In the simplest form of the PF, the particles are initially propagated through the dynamic model and are next weighted according to the likelihood function that determines how closely the particles match the measurements. Subsequently, at the re-sampling step, those that best match the measurements are multiplied and the rest are discarded. Although due to novelty of the PF, its members have received considerable attention in lots of different scientific fields, there are still some outstanding difficulties with this method such as sample impoverishment due to lack of population diversity.

As a characteristic of any sample based method, PF estimation performance depends on the number of utilized particles as well. Increasing the number of particles increases the estimation accuracy to some extent, but at the same time results in a dramatic increase in run time, which is not desirable especially for on-line applications. A more detailed review of literature reveals that heuristic optimization methods such as particle swarm optimization (PSO), genetic algorithm (GA), ant colony optimization (ACO), etc. have been used to enhance the performance of the PF. Zhong et al. [4] and Hao et al. [5] have tried using ACO to optimize the re-sampling step in order to avoid the sample impoverishment problem. In [5], the optimization allows the particles to move closer to their local highest posterior density function, thus producing a better estimation results. Zhang et al. [6] have combined PSO with PF before re-sampling stage in order to increase diversity of the particles. In [6], particles' weights in PF are considered as the fitness values in PSO, and the particles set which is formed after re-sampling step is operated by PSO. This process caused the particles to move to a point with a higher fitness value. Yu et al. [7] have incorporated ACO into PF in order to optimize the sampling process. Unlike the standard PF, re-sampling is not required in their scheme.

As it already mentioned, state estimation can be thought as a stochastic dynamic optimization problem as well. In this context, various kinds of powerful heuristic optimization methods can be utilized to estimate the states of nonlinear dynamic systems.

Parpinelli et al. [8] has compared three swarm intelligence algorithms of bacterial foraging optimization (BFO), PSO and artificial bee colony (ABC) for the optimization of hard engineering problems. It is shown that PSO results in the best balance between quality of solution and number of function evaluations.

According to [9], an appropriate initial population size plays a key role in the effectiveness and efficiency in the performance of the evolutionary algorithms such as PSO, differential evolution (DE), etc. Therefore, a new method is presented in the current study that allows setting the number of needed particles to search the state space adaptively. In this paper, the sample size is determined based on the weighted variance of the participating particles.

Leong and Yen [10] have presented a method to tune the swarm size based on the rank and density of the population. The presented algorithm has many parameters to set in advance that makes its utilization complicated. Sun et al. [11] have also proposed a scheme to improve the performance of PSO based on time-variant particle population function, which makes the population decrease gradually in order to reduce the computational cost, and produce random particles at periodical phases to avoid trapping in the local optima.

Coelho and de Oliveira [12] have introduced two usual ideas of population resizing in GA to the PSO. The first idea controls the birth and death of the particles at every generation; while the second one increases the swarm size at initial iterations to enhance the exploration and decreases it subsequently to improve the exploitation capability. Applying these ideas to some benchmark functions has shown the superiority of the first idea. Lei [13] has presented two dynamic population size improvements for the standard PSO as well. The first method starts with a small number of particles and increases the number of particles iteratively, while the second algorithm starts with a large number of particles and decreases the swarm size gradually. Both methods result in reducing computational time, but the second method is more successful in converging to the global optimum. Sample size is adaptively determined in [14] based on a complicated combination of the density and the average Hamming distance of particles.

In following, application of heuristic optimization methods for state estimation will be reviewed briefly. Nobahari et al. [15], Heris and Khaloozadeh [16] and Naka et al. [17] have proposed state estimators based on the traditional ACO algorithm and in a framework similar to the PF. Xu et al. [18] have used PSO to search the optimal or near optimal parameters that produce desirable steady state filters such as α–β filter and α–β–γ filter. Jeong and Park [19] have used a hybrid PSO (HPSO) for a cluster of PC systems to minimize the difference between measured and calculated state variables in order to estimate the system states. Zhang et al. [20] have proposed utilizing the Gaussian PSO to approximate the optimal solution

Table 1
Pseudo code of the proposed APSO.

Begin
Parameters initialization
Population initialization and corresponding fitness evaluation
Determine the global best particle
for R = 1:[iter$_{max}$/T]
  for generation = 1:T
    for i = 1:Population size
      Update velocity and position of particles (according to pertinent PSO version)
      Fitness evaluation
      Update the personal best and global best positions
    end
  end
  ***Rank population** according to their fitness*
  ***Calculate weighted variance** according to Eq. (7)*
  if $\frac{\Delta\sigma_\omega(R,R-1)}{\sigma_\omega(R-1)} > (1+\varepsilon)$
    add particles
  else if $\frac{\Delta\sigma_\omega(R,R-1)}{\sigma_\omega(R-1)} < (1-\varepsilon)$
    remove worse particles
  else
    keep the population without any changes
  end
end

Table 2
Pseudo code of the proposed ADE.

Begin
Parameters initialization
Population initialization and corresponding fitness evaluation
Determine the best individual
for R = 1:[iter$_{max}$/T]
  for generation = 1:T
    for i = 1:Population size
      Mutation (DE/rand/1)
      Crossover
      Mapping the offspring onto the search space (if necessary)
      Selection (offspring-parent competition)
    end
  end
  ***Rank population** according to their fitness*
  ***Calculate weighted variance** according to Eq. (7)*
  if $\frac{\Delta\sigma_\omega(R,R-1)}{\sigma_\omega(R-1)} > (1+\varepsilon)$
    add particles
  else if $\frac{\Delta\sigma_\omega(R,R-1)}{\sigma_\omega(R-1)} < (1-\varepsilon)$
    remove worse particles
  else
    keep the population without any changes
  end
end

Download English Version:

https://daneshyari.com/en/article/494928

Download Persian Version:

https://daneshyari.com/article/494928

Daneshyari.com