Expert Systems with Applications 38 (2011) 14852–14860

Contents lists available at ScienceDirect

Expert Systems with Applications

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

# Swarm intelligence-based extremum seeking control

Chen Hong [a,b,*], Kong Li [b]

[a] Wuhan Second Ship Design and Research Institute, Wuhan, Hubei, China
[b] Huazhong University of Science and Technology, Wuhan, Hubei, China

ARTICLE INFO

*Keywords:*
Particle swarm optimization
Extremum seeking control
State regulation
Adaptive control
Swarm intelligence-based optimization

ABSTRACT

This paper proposes an extremum seeking control (ESC) scheme based on particle swarm optimization (PSO). In the proposed scheme, the controller steers the system states to the optimal point based on the measurement, and the explicit form of the performance function is not needed. By measuring the performance function value online, a sequence, generated by PSO algorithm, guides the regulator that drives the state of system approaching to the set point that optimizes the performance. We also propose an algorithm that first reshuffles the sequence, and then inserts intermediate states into the sequence, in order to reduce the regulator gain and oscillation induced by population-based stochastic searching algorithms. The convergence of the scheme is guaranteed by the PSO algorithm and state regulation. Simulation examples demonstrate the effectiveness and robustness of the proposed scheme.



## 1. Introduction

Regulation and tracking of system states to optimal setpoints or trajectories are typical tasks in control engineering. However, these optimal setpoints are sometimes difficult to be chosen *a priori*, or vary with the environmental condition changes. Extremum seeking control (ESC) is a kind of adaptive control schemes that can search for the optimal setpoints online, based on the measurement of the performance output or its gradient. ESC can be regarded as an optimization problem, and many of the schemes used in ESC are transferred from optimization algorithms. However, some optimization algorithms cannot be incorporated into the ESC framework easily, for the reason that, practical issues, such as stability, noise, regulation time, control gain and oscillation limitation, will prevent the use of some optimization algorithms from ESC context. Thus, the study on suitable combination of ESC and optimization algorithms is of great interest both in academics and in engineering.

Unlike the traditional variational calculus-involved optimal control method, the explicit form of the performance function is not needed in ESC. Therefore, ESC is useful in the applications that the performance functions are difficult to model. After Krstic and Wang's stability studies (Krstic & Wang, 2000), research on ESC has received significant attention in recent years. The recent ESC application examples include active flow control (Beaudoin, Cadot, Aider, & Wesfreid, 2006), bioreactor or chemical process control (Bastin, Nescaroni, Tan, & Mareels, 2009; Hudon, Guay, Perrier, & Dochain, 2008; Hudon, Perrier, Guay, & Dochain, 2005), cascaded Raman optical amplifiers (Dower, Farrell, & Nesic, 2008), antilock braking system design (Zhang & Ordonez, 2007), thermoacoustic cooler (Li, Rotea, Chiu, Mongeau, & Paek, 2005), and fuel cell power plant (Zhong, Huo, Zhu, Cao, & Ren, 2008). There also have been considerable theoretical studies in ESC, such as stability studies on perturbation-based ESC (Krstic, 2000; Krstic & Wang, 2000), ESC for discrete-time systems (Joon-Young, Krstic, Ariyur, & Lee, 2002), PID tuning by ESC (Killingsworth & Krstic, 2006), ESC for nonlinear dynamic systems with parametric uncertainties (Guay & Zhang, 2003), and ESC for state-constrained nonlinear systems (DeHaan & Guay, 2005). The majority of ESC literature focused on two issues, the one is the searching for the optima, and the other is the regulation of the systems. The recent studies of Zhang and Ordonez (2007, 2009) presented a numerical optimization-based ESC (NOESC) framework that takes the advantage of numerical algorithms to find the optima online. However, these algorithms are unable to find the global optima if the assumption that the performance functions are convex and continuous does not hold. Furthermore, the NOESC is sensitive to measurement noise, due to the poor robustness of the numerical algorithms.

Particle swarm optimization (PSO) algorithm is a population-based stochastic optimization method which first devised by Kennedy and Eberhart (1995). PSO algorithm mimics the food-seeking behavior of birds or fishes. Due to its simplicity and effectiveness, PSO algorithm witnesses a considerable interest and is applied in many areas. The convergence of PSO algorithm is studied by deterministic method (Eberhart & Shi, 2001) or stochastic process theory (Jiang, Luo, & Yang, 2007). Clerc and Kennedy (2002) presented a convergence condition of PSO algorithm. Rapaic and Kanovic (2009) studied the time-varied parameter PSO algorithm

* Corresponding author at: Wuhan Second Ship Design and Research Institute, Wuhan, Hubei, China.
*E-mail address:* Chenhong.paper@gmail.com (C. Hong).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.05.062

and the selection of the parameters. Studies have shown that PSO algorithm is able to handle a wide range of problems, such as integer optimization (Laskari, Parsopoulos, & Vrahatis, 2002), multi-objective optimization (Dasheng, Tan, Goh, & Ho, 2007), and global optimization of multimodal functions (Liang, Qin, Suganthan, & Baskar, 2006). The recent application of PSO algorithm includes power systems (del Valle, Venayagamoorthy, Mohagheghi, Hernandez, & Harley, 2008), flights control (Duan, Ma, & Luo, 2008), and nuclear power plants (Meneses, Machado, & Schirru, 2009), to name a few. In control engineering, PSO algorithm is usually employed to identify the models (Panda, Mohanty, Majhi, & Sahoo, 2007), or to optimize the parameters of the controller offline (El-Zonkoly, 2006). PSO algorithm is usually regarded as an effective global optimization method. However, it is often used in offline optimization, and depends on the explicit form and the solvability of the performance functions. However, for some complex models, e.g. active flow control problems which described by Navier–Stokes equations, it is difficult to obtain the optimal parameters of the controllers by time-consuming numerical simulations.

In this paper, we extend the numerical optimization-based ESC (Zhang & Ordonez, 2007) by incorporating PSO algorithm into the extremum seeking framework. We also address the practicability issues of this scheme, and propose a reshuffle-then-insertion algorithm to reduce the control gain and oscillation. In the proposed scheme, a sequence converging to the global optima is generated by PSO with reshuffle-then-insertion algorithm. The sequence used as a guidance to regulate the state of the plant approach to the optimal set point. This paper is organized as follows. Section 2 gives a problem statement, where a PSO-based ESC (PSOESC) framework is introduced. We then review the standard PSO algorithm in Section 3. The details of the PSOESC scheme for linear time invariant (LTI) systems and feedback linearizable systems are presented in Section 4, where the reshuffle-then-insertion approach for improving the practicability of the PSOESC is also proposed. Section 5 presents the results of the numerical experiments. Finally, Section 6 concludes the paper.

## 2. Problem statement

In control practice, the problem of seeking for an optimal set point is encountered usually. In general, this problem can be represented as model

$$\dot{x} = f(x, u), \tag{1}$$
$$y = J(x). \tag{2}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ is the performance output to be optimized, $f : D \times \mathbb{R} \to \mathbb{R}^n$ is a sufficiently smooth function on $D$, and $J : D \to \mathbb{R}$ is an unknown function. For simplicity, we assume $D = \mathbb{R}^n$ in this paper. Without loss of generality, we consider the minimization of the performance function (2). Unlike optimal control, extremum seeking control finds the optimal set point by online measurement of the performance function, without the knowledge of its explicit form.

ESC can be considered as a class of constrained optimization problem whose constraint is the differential Eq. (1), instead of the algebraic constrains in traditional optimization problems. Then, ESC control problem can be stated as:

$$\begin{aligned} \min_{x \in D} \quad & J(x) \\ s.t. \quad & \dot{x} = f(x, u). \end{aligned} \tag{3}$$

When (1) is controllable, there always exists a control $u$ such that state $x$ transfers to any position in $\mathbb{R}^n$ in a finite time. We then can apply any optimization algorithm to produce a guidance sequence to determine a trajectory to the optimal set point in the state space (Zhang & Ordonez, 2007).

Similar to numerical optimization-based ESC (Zhang & Ordonez, 2009), we present a PSO-based ESC block diagram as Fig. 1, where the nonlinear plant $\mathbb{F}$ is modeled as (1) and the performance function $J$ is (2). Unlike Zhang & Ordonez, 2009, we apply PSO algorithms to substitute the numerical optimization algorithms to produce the extremum seeking sequence $\{X_k\}$. The sequence $\{X_k\}$ is generated as the target state in every seeking iterative step, based on the online measurement of $y$. The regulator $K$ regulates the state of $F$ follows the sequence as $X_1, X_2, \ldots, X_k$, toward the optimal set point.

## 3. Particle swarm optimization

PSO is a population-based random optimization algorithm that mimics the behavior of bird or fish swarm in searching food. In the swarm, each particle has a variable speed, moving toward the positions of its own best fitness achieved so far and the best fitness achieved so far by any of its neighbors.

Let $S \subset \mathbb{R}^n$ is an $n$-dimensional search space. The size of the particle swarm is denoted as $N$. The position of the $i$th particle is represented as an $n$-dimensional vector $\tilde{X}_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T \in S$, and its velocity is denoted as $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})^T \in S$, where $i = 1, 2, \ldots, N$ is the identity of each particle. The best position it has achieved so far is $P_i^{pbst} = (p_{i1}, p_{i2}, \ldots, p_{in})^T \in S$. The best position achieved so far in its neighborhood is $P_{gbst}$. On the time step $k$, the update equation of the basic PSO algorithm is (Kennedy & Eberhart, 1995)

$$\begin{aligned} V_i(k+1) = V_i(k) + cr_1(k)\left(P_i^{pbst}(k) - \tilde{X}_i(k)\right) \\ + cr_2(k)\left(P_{gbst}(k) - \tilde{X}_i(k)\right), \end{aligned} \tag{4}$$

$$\tilde{X}_i(k+1) = \tilde{X}_i(k) + V_i(k+1), \tag{5}$$

where, $c$ is an acceleration coefficient, $r_1$, $r_2$ are two independent random numbers with uniform distribution in the range of $[0, 1]$. Though this version of PSO has been shown to perform well in some optimization cases, it is not suitable in ESC context for its possible explosion property will lead to an unfeasible large control gain.

Clerc and Kennedy (2002) suggested a PSO algorithm which is known as the standard PSO, by introducing constriction coefficient $\chi$. The update equation in the standard PSO is

$$\begin{aligned} V_i(k+1) = \chi\Big[V_i(k) + \varphi_1 r_1(k)\left(P_i^{pbst}(k) - \tilde{X}_i(k)\right) \\ + \varphi_2 r_2(k)\left(P_{gbst}(k) - \tilde{X}_i(k)\right)\Big], \end{aligned} \tag{6}$$

$$\tilde{X}_i(k+1) = \tilde{X}_i(k) + V_i(k+1), \tag{7}$$

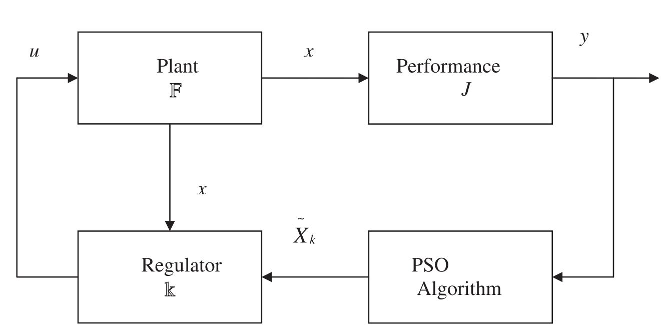


**Fig. 1.** PSO-based extremum seeking block diagram.