Commun Nonlinear Sci Numer Simulat 15 (2010) 3632–3639

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

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

PID controller design of nonlinear systems using an improved particle swarm optimization approach

Wei-Der Chang *, Shun-Peng Shih

Department of Computer and Communication, Shu-Te University, Kaohsiung 824, Taiwan

ARTICLE INFO

Article history:
Received 9 May 2009
Received in revised form 2 January 2010
Accepted 6 January 2010
Available online 14 January 2010

Keywords:
PID controller
Improved particle swarm optimization
Nonlinear systems
Inverted pendulum control

ABSTRACT

In this paper, an improved particle swarm optimization is presented to search for the optimal PID controller gains for a class of nonlinear systems. The proposed algorithm is to modify the velocity formula of the general PSO systems in order for improving the searching efficiency. In the improved PSO-based nonlinear PID control system design, three PID control gains, i.e., the proportional gain K_p, integral gain K_i, and derivative gain K_d are required to form a parameter vector which is called a particle. It is the basic component of PSO systems and many such particles further constitute a population. To derive the optimal PID gains for nonlinear systems, two principle equations, the modified velocity updating and position updating equations, are employed to move the positions of all particles in the population. In the meanwhile, an objective function defined for PID controller optimization problems may be minimized. To validate the control performance of the proposed method, a typical nonlinear system control, the inverted pendulum tracking control, is illustrated. The results testify that the improved PSO algorithm can perform well in the nonlinear PID control system design.

1. Introduction

The particle swarm optimization (PSO) is a new evolutionary computation technique and has been introduced in various application fields in recent years [1–7]. This algorithm combines the social psychology principles in socio-cognition human agents and evolutionary computations. It is initially motivated by the behavior of organisms, such as a fish school and bird flock. The computational efficiency of the PSO algorithm is rather excellent and it is also easy to be implemented. Moreover, unlike other heuristic optimization methods, PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities. The algorithm begins with randomly generating an initial population, which is composed of a number of candidate solutions. In the PSO algorithm, every candidate solution is called a particle or an individual and such a particle is moved by a velocity updating in accordance with itself own experience and other particles' experiences. It possesses a constructive cooperation and sharing information relatively between particles of the population [6,7].

Due to the good features of PSO algorithm, nowadays it has been emerged as a new and attractive optimizer and applied in variety of research fields. For example, the system identification for a class of nonlinear rational filters was solved using the PSO algorithm [5]. The rational filter parameters can be correctly estimated. In [6], the author used the PSO algorithm to solve the optimal design problem of multimachine power-system stabilizers (PSSs). Two eigenvalue-based objective functions were considered to enhance the system damping of electromechanical modes. In [7], a hybrid PSO algorithm which incorporates chaos dynamics was proposed to enhance the performance of PSO. In the case the population-based evolutionary searching and chaotic searching are well combined.

* Corresponding author.
E-mail address: wdchang@stu.edu.tw (W.-D. Chang).

1007-5704/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cnsns.2010.01.005

The design of PID controller is another important subject in this study. In the control engineering, the PID controller has been employed for a very long time due to its good characteristics such as the simplicity in architecture, easy implementation, and mature theoretical analysis. Until now, it is still widely applied to the industrial control, even if many new control approaches have been consequently developed. This controller contains three adjustable gain parameters, the proportional gain K_p, integral gain K_i, and derivative gain K_d, respectively. In the traditional PID controller design, there are the following methods for the linear model plant or linear model plant with time delay: (1) process reaction curve (Z-N tuning method), (2) sustained oscillation PID tuning, (3) damped oscillation or quarter amplitude decay PID tuning, and (4) relay tuning method, etc. [8]. In [9], the author utilized the PSO algorithm to the optimum design of PID controller in the automatic voltage regulator (AVR) system comprising four components: amplifier, exciter, generator, and sensor. Each component is expressed as a simple first-order linear system. Three PID control gains were determined by minimizing a new time-domain performance criterion via the PSO algorithm. This work is about the PSO-based PID controller design for linear systems.

This paper proposes a new improved PSO algorithm and uses it to solve the PID controller design problem for a class of nonlinear systems. The remainder of this paper is organized as follows. In Section 2, the improved PSO algorithm is clearly presented and some function optimization tests are demonstrated. Section 3 presents the full design steps of the improved PSO-based nonlinear PID control system. In Section 4, the tracking control of the inverted pendulum system is typically illustrated to verify the feasibility of the proposed method. Many simulation examinations are also given. Finally, a brief conclusion is stated in Section 5.

2. Improved PSO algorithm

In 1995, Kennedy and Eberhart initially proposed the particle swarm concept and PSO algorithm [1]. This algorithm is one of optimization methods and evolutionary computations. It has been proven to be efficient in solving optimization problem especially for nonlinearity and nondifferentiability, multiple optimum, and high dimensionality. It guides searches using a population constructed by many particles rather than individuals. In the PSO algorithm, each particle then represents a candidate solution to the optimization problem. The particle keeps track of its coordinates in the problem space which are associated with the best solution it has achieved so far. This value is called *pbest*. Another "best" value that is tracked by the global version of the particle swarm optimizer is the overall best value and its location obtained from any particle of the population so far. This location is named *gbest*. At each time step, the PSO algorithm concept consists of changing the velocity that accelerates each particle toward its *pbest* and *gbest* locations. Acceleration is weighted by a random term with separate random numbers being generated for acceleration toward *pbest* and *gbest* locations, respectively [3].

Assume that each particle is considered in the N-dimensional space, $\Theta_i = [\theta_{i1}, \theta_{i2}, \ldots, \theta_{iN}]$ denotes the ith particle's position, and $V_i = [v_{i1}, v_{i2}, \ldots, v_{iN}]$ denotes the ith particle's velocity. The best previous position of the ith particle, i.e., particle best, is represented as $P_i = [p_{i1}, p_{i2}, \ldots, p_{iN}]$ and the index of the best particle among all particles in the population, i.e., global best, is assigned by the symbol g. The velocity and position updating equations of the original PSO algorithm, for $n = 1, 2, \ldots N$, are given below

$$v_{in} = w * v_{in} + c_1 * rand() * (p_{in} - \theta_{in}) + c_2 * Rand() * (p_{gn} - \theta_{in}), \tag{1}$$

$$\theta_{in} = \theta_{in} + v_{in}, \tag{2}$$

where w is called the inertia weight which balances the global and local search, c_1 and c_2 are two positive constants, $rand()$ and $Rand()$ are two random numbers in the interval [0, 1]. Eqs. (1) and (2) are the main evolutionary mechanisms of the PSO algorithm for solving the optimization problem.

In this study, an improved velocity updating equation is first proposed. The above Eq. (1) is modified as

$$v_{in} = w * v_{in} + c_1 * rand() * (p_{in} - \theta_{in}) + c_2 * Rand() * (p_{gn} - \theta_{in}) + c_3 * RAND() * (p_{sn} - \theta_{in}), \tag{3}$$

where p_{sn} represents the best particle's position among all particles in the sub-population that the ith particle belongs to, c_3 is a positive constant, and $RAND()$ is also a random number in the range [0, 1]. From Eq. (3), it can be seen that the third kind of best particle information is considered. We can illustrate the improved velocity updating formula of Eq. (3) by a student-class-school model. In the model a student can be regarded as a particle, a class as a sub-population, and a school as a population, respectively. In the original PSO algorithm, the experience of a student (particle) is only influenced by his/her own past best experience (*pbest*) and the experience of the global best student (*gbest*) in the school (population). Nevertheless, in the general environment the experience of a student may be not only influenced by these two factors, but really also by the experience of the best student (we here call sub-best, *sbest*) in his/her own class (sub-population). The improved velocity formula is reasonably motivated from this simple concept. The following gives the design step for implementing the improved version of the PSO algorithm:

Step 1. Randomly generate an initial population that contains H particles (population size) from certain search interval.
Step 2. If a prescribed number of iterations (generations) is achieved, then stop the algorithm.
Step 3. Evaluate the objective function of every particle and record each particle's best previous position (*pbest*), sub-best position (*sbest*), and global best position (*gbest*).