2nd International Conference on Engineering Optimization
September 6-9, 2010, Lisbon, Portugal

Chaos Particle Swarm Optimized PID Controller for the Inverted Pendulum System

O. Tolga Altinoz, A. Egemen Yilmaz, Gerhard Wilhelm Weber

Hacettepe University Bala Vocational School, Ankara, Turkey, taltinoz@hacettepe.edu.tr
Ankara University Electronics Engineering Department, Ankara, Turkey, aeyilmaz@eng.ankara.edu.tr
Middle East Technical University Institute of Applied Mathematics, Ankara, Turkey, gweber@metu.edu.tr

Abstract
This paper will be focus on the application of the chaos embedded particle swarm optimization algorithm (CPSO) into one of the popular problem setups in the engineering application area of control systems, which is called the inverted pendulum.

The inverted pendulum is composed of a cart and a free moving pendulum. By adjusting the cart position, the pendulum can be maintained at upright position. The position of the cart system is controlled via PID controller, which is a linear control method. The performance of the PID controller depends on its parameters. Thus, in this study, the parameters of the PID algorithm are determined by CPSO, which can be used in order to obtain the global optima of any system.
Keywords: Particle Swarm Optimization, Chaos Theory, PID Control, Inverted Pendulum System, Multidimensional Optimization.

1. Introduction

The pendulum is one of the benchmark problems of control theory for illustration and comparison of different control methodologies. The popularity of this problem comes from the motivation of development of missiles, rockets, robots and other transportation means.

Fig. 1 shows the general model of the inverted pendulum. The pendulum is composed of a free moving pendulum with the mass m and length l attached to the cart with mass M, where a force f is applied to this setup. The following expressions give the differential equations of the inverted pendulum.

$$\ddot{x} = \frac{mgsin(\theta)cos(\theta) - ml\dot{\theta}^2 sin(\theta) - f}{mcos(\theta)^2 - (M+m)} \tag{1}$$

$$\ddot{\theta} = \frac{-(M+m)gsin(\theta) + ml\dot{\theta}^2 sin(\theta)cos(\theta) + fcos(\theta)}{mlcos(\theta)^2 - (M+m)l} \tag{2}$$

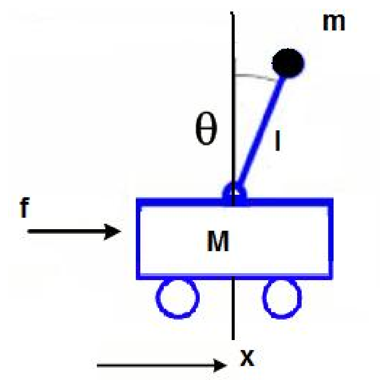

Figure 1: Inverted Pendulum Model

Eq.s 1 and 2 represent the nonlinear mathematical model of the inverted pendulum. The rod angle θ at the model can be changed under the influence of the environment. Thus, the control structure is needed to correct the pole angle with applied the force f to the cart. In order to solve this problem, various control methodologies (such as optimal control, linear/nonlinear control, intelligent and adaptive control methods) were previously applied. Some of these control methods use the linearized model of the system. Hence, for such methods, the performance of the controller is only valid around the operation point. In this study, the angle control loop will be defined and the PID controller will be designed by using the CPSO algorithm.

This paper is organized as follows: Section 2 gives a general description of the conventional PSO together with its advantages and drawbacks. Then, the CPSO algorithm is defined. Section 2.1 to 2.5 discusses the CPSO algorithm components. The last two section gives the simulation results and conclusions.

2. Particle Swarm Optimization

The PSO algorithm depends on motions of particles (swarm members) searching for the global best in an N-dimensional continuous space. The position of each particle is nothing but a solution candidate, and every time, the fitness of this candidate is re-evaluated. In addition to its exploration capability (i.e. the tendency for random search throughout the domain), each particle has a cognitive behavior (i.e. remembering its own good memories and having tendency to return there); as well as a social behavior (i.e. observing the rest of the swarm and having tendency to go where most other particles go).

The original PSO formulation of Kennedy and Eberhart [1] depends on the update of the position x_i and the velocity v_i of the ith particle (swarm member) as follows:

$$v_i = v_i + c_1 \times rand() \times (pbest_i - x_i) + c_2 \times rand() \times (gbest_i - x_i) \tag{3}$$

$$x_i = x_i + v_i \times \Delta t \tag{4}$$

where c_1 and c_2 are measures indicating the tendencies of approaching to *pbest* and *gbest*, which are the best positions achieved personally by the ith particle and the whole swarm, respectively. In other words, c_1 and c_2 are the measures of the cognitive and the social behaviors (called cognitive and social parameters respectively), respectively. $rand()$ is a random number between 0.0 and 1.0; and the time step size Δt is usually taken to be unity for simplicity.

This optimization algorithm demonstrates an outstanding performance under complicated problems. But still, occasionally this algorithm faces problems such as getting stuck at local optima and stagnation. Thus, various improvements have been introduced in order to get rid of these problems.

First, Shi and Eberhart [2] introduced a term called "inertial weight" in order to improve the performance of the method (use for control local and global exploration behaviour of the population). By the introduction of this term, which puts an additional control on the current velocity of the particles, Eq. 3 is modified as:

$$v_i = w \times v_i + c_1 \times rand() \times (pbest_i - x_i) + c_2 \times rand() \times (gbest_i - x_i) \tag{5}$$

which is referred to as the "inertial weight PSO", and is currently accepted as the de-facto PSO formulation. Moreover, in a following study [3], Shi and Eberhart showed that the ideal choice for the inertial weight is to decrease it linearly from 0.9 to 0.4.

Later, various hybrid and innovative methods were introduced in order to get rid of the problems. One of the main solutions for getting stuck at the local optimum is the so-called Chaos Particle Swarm Optimization (CPSO). In this study, the CPSO is used to solve the inverted pendulum problem. The outline of the CPSO flow can be listed as follows:

```
Choas PSO Algorithm
Initialize random velocity and position
Do
  For i = 1 to swarm size
        1) Calculate fitness function (fit) which is the function that aim to minimized.
        This is the function that the mean squre error of the referance signal and the
        output of the system that the rod angle
        If fit < best pattern
             ith particle best position = ith particle position
             best pattern = fit
        end if
  end for
  find min best pattern and corresponding particle
  For i = 1 to swarm size
        Update velocity
        Update position
  end for
```

```
  Check the limits of the maximum velocity
  For k = 0 to any number
     1) Execute variable mapping that maps the positions into chaos variables
     (Section 2.1)
     2) Use Logistic map and find new chaos variable
     (Section 2.2)
     3) Execute Inverse Variable Mapping that convert chaos variable into positions
     (Section 2.3)
     If (new position < old position)
         Position = New Position
         Break
     End if
     4) Change the search area
     (Section 2.4)
  end for
  end for
while break if maximum iterations or minimum error
```

2.1. Chaos and Variable Mapping

Application of the chaotic map might be helpful for escaping from a local minimum [4]; and it can improve the global/local searching capabilities. Chaos maps are used in interaction with the PSO algorithm in order to search the solution space better and directly [5]. Chaotic maps are used for chaotic search operation. In this manner, the speed of convergence increases.

For chaotic search, the initial value of the chaotic map must be determined. The easiest way for this is to determine the initial value as a random number between [0,1]. The most common way to find the initial value is to define the *carrier method* by defining a decision variable from PSO variables, and map this variable into the chaotic domain by using carrier equation as defined in Eq. **??**.

$$cx_i = (x - X_{min})/(X_{max} - X_{min}) \tag{6}$$

where cx_i is the decision variable which is the initial value of the chaotic map, x is the position of the particle and X_{min} and X_{max} are the boundaries for the position.

The chaotic search is used in order to find a better global best in PSO. The general usage of the chaotic search is that if the global best solution is found, the chaotic search starts and tries to find a better solution in the given boundary. Hence, it is essential that the chaotic orbit is ergotic, which means that the initial value of the chaotic map must be varied.

2.2. Chaos Search

Chaotic maps are used for chaotic search. By this way, the speed of convergence increases. The particles escape from local optima. Chaos, benefiting from the properties of periodicity and stochasticity, is definitely a good candidate, but currently the well-known logistic map, piecewise linear chaotic map (PWLCM) [5] is prevalently used in order to perform the chaotic search. In [4], same approach was applied to PSO algorithm by using the Henon map; in [6], Zaskauskii map and in [7], the tent map was used.

In [8], both logistic and tent maps are used for chaos search, and the performances were compared with the conventional PSO algorithm. The results showed that, the chaos based PSO algorithms are efficient in the sense of solution quality and convergence properties. However, because of the relative complex structure of Chaotic PSO algorithms, the computational efficiency is inevitably less than that of conventional PSO. In terms of the solution quality, both logistic and tent maps have equal performance. However, CPSO with tent map is faster than the logistic map CPSO while reaching the optimal solution. The sequence of the logistic map is not symmetric, and this fact affects the capability of chaos search [9]. In this study, following the general acceptance, the logistic map is used for chaotic search in the CPSO.

The discrete-time dynamical system in the iteration form in equation **??** is called mapping or map.

$$x_{i+1} = F(x_i, P) \tag{7}$$

where P is the control parameter, x is a vector and F is a nonlinear transformation [10]. The *Poincare section* is defined of studying the dynamical behavior of the mapping. The *Poincare map* summarizes the behavior of the system dynamics. As the control parameter is varied, the steady state points are varied and the $bifurcations$ of these new points are interested [10]. Two-dimensional maps are the models of the Poincare maps of forced oscillator systems, while one-dimensional maps constitute nothing but a special case of this map [10].

At each dimension of the problem space and at each particle, the chaotic map is applied. Hence, the time requirements for SM depend on the problem dimension and the swarm population.

One-dimensional maps include the chaotic behavior. These maps have so far been used in many applications because of their simple definitions. The *logistic map*, which is a model of population biology, is frequently used with PSO. This map is a unimodal map, which has the finite interval and single maximum.

$$x_{i+1} = \mu x_i(1 - x_i) \tag{8}$$

where μ is set to 4 for ergodicity.

2.3. Inverse Variable Mapping

After determination of the chaotic variable from the chaotic map with the initial value from variable mapping, the chaotic variable is converted into the particle velocity by using Eq. **??**.

$$x = X_{min} + cx_i(X_{max} - X_{min}) \tag{9}$$

The position, which is taken from the inverse mapping, is used for evaluation of the new solutions. If the new solution is better than the non-chaos-search one, the new solution is put pursuant to chaos search.

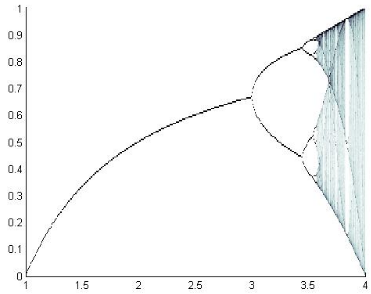

Figure 2: Bifurcation Diagram of Logistic Map

2.4. Search Range

If the search space extends in a wide area, the searching cannot be completed in the optimal area in a short time [5]. Hence, in order to obtain high performance, the chaotic search is run in a small range. This search area is changed in the current optimal solution neighborhood [9].

$$X_{min} = max(X_{min}, x_g - r(X_{max} - X_{min})) \tag{10}$$
$$X_{max} = min(X_{max}, x_g + r(X_{max} - X_{min})) \tag{11}$$

where x_g is the global best position and r is the variable between 0 and 1. In this study, r is choosen as 0.45. Decreasing the range of the chaotic search will increase the searching efficiency.

3. Simulation Results

In Fig. 3 the controller block of the system is given. The angle of the rod is controlled via the PID controller where the aim is to hold the rod at the upright position. The model of the inverted pendulum was given in the introduction section, and is used in the simulation environment.

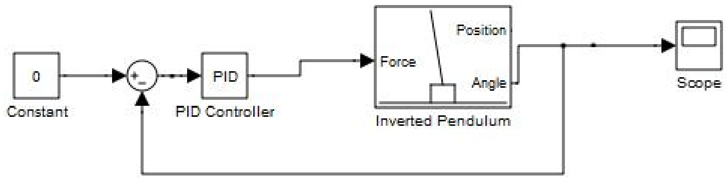

Figure 3: Inverted Pendulum with PID Control Model

PID controller with CPSO optimized parameters (which are detemined via minimization of the mean square error between the reference signal and the system output), is applied to the system. Fig. 4 gives the change of the rod angle, the inverse error and the output of the inverted pendulum. From the results, it is observed that the controller parameters are adjusted by using CPSO in $0.5ms$.

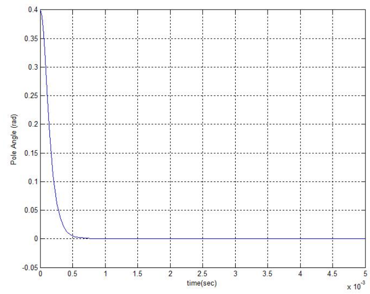

Figure 4: Change of the Pole Angle

3. Conclusion

In this study, the novel chaos particle swarm optimization algorithm (CPSO) is developed, and applied to the popular engineering problem called inverted pendulum. The PID controller is designed by applying CPSO to the nonlinear system model. At each iteration, the PID parameters are found off-line; then, the optimized controller is applied to the system. The results shows that a smoooth response can be obtained.

References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proc. IEEE Int. Conf. on Neural Networks*, 1942-1948, 1995.

[2] Y. Shi, R. C. Eberhart, A modified particle swarm optimizer, *Proc. IEEE International Conference on Evolutionary Computation*, 69 - 73, 1998.

[3] Y. Shi, R. C. Eberhart, Empirical study of particle swarm optimization, *Proc. Congress of Evolutionary Computing*, 1945 - 1950, 1999.

[4] L.S. Coelho, V.C. Mariani, A novel chaotic particle swarm optimization approach using Henon map and implicit filtering local search for economic load dispatch, *Chaos, Solitons and Fractals*, Vol. 39, 510-518, 2009.

[5] T. Xiang, X.Liao, K.W. Wang, An Improved Particle Swarm Optimization Combiined with Piecewise Linear Chaotic Map, *Applied Mathematics and Computation*, 1637-1645, 2007.

[6] L.S. Coelho, A novel quantum particle swarm optimizer with chaotic mutation operator, *Chaos, Solitons and Fractals*, Vol. 37, 1409-1418, 2008.

[7] Y. Song,Z. Chen,Z. Yuan, New chaotic PSO-based Neural Netwrok Predictive Control for Nonlinear Process, *IEEE Transaction on Neural Networks*, Vol. 18, No. 2, 595-601, 2007.

[8] C. Jiejin, M. Xiaoqian, L. Lixiang, P. Haipeng, Chaotic particle swarm optimization for echonnomic dispatch considering the generator constraints, *Energy Conversion and Management*, Vol. 48, 645-653, 2007.

[9] X.Y. Gao, L.Q. Sun, D.S. Sun, An enhanced particle swarm optimization algorithm, *Information Technology Journal*, vol. 8, 1263-1268, 2009.

[10] J.M.T. Thompson, H.B. Stewart, Nonlinear Synamics and Chaos, *John Wiley Sons*, 2nd Edition, 2002.