International Journal of Research in Engineering, Science and Management
Volume 4, Issue 12, December 2021
https://www.ijresm.com | ISSN (Online): 2581-5792


# Optimize PID Controller Parameters Using Particle Swarm Optimization in Inverted Pendulum Control

Huynh Thanh Son[1], Le Ngoc Dung[2*]
[1,2]*Dong Nai Technology University, Dong Nai Province, Vietnam*

***Abstract*: This paper proposes a solution using the particle swarm optimization (PSO) to self-correct and simultaneously optimize the parameters of the PID controllers in the inverted pendulum control system with the purpose to make the tolerances of arm angles and pendulum angles are the smallest, helping to keep the pendulum balanced. The rotary inverted pendulum is a very interesting object, which represents a class of control objects with complex nonlinearities and is used as a common model for many applications in control engineering. Based on the modeling of the inverted pendulum system with the use of PID controllers whose parameters $K_p$, $K_i$, $K_d$ are selected experimentally, usually still make the knife system large dynamics and does not reach a steady state. Therefore, the proposed solution has shown its superiority through the results of system simulation by Simulink_matlab tool, the parameters $K_p$, $K_i$, $K_d$ of the two PID controllers have been adjusted, updated intending to make the system stable faster. If simultaneously increasing the number of generations 15, 20 and 30 corresponding to the number of individuals of 20, 30, and 50, the Adaptation Function converges very quickly, the arm angle θ and the pendulum angle α stabilize faster.**

***Keywords*: rotary inverted pendulum, particle swarm optimization, PSO-PID, optimize PID controller.**

## 1. Introduction

The PID controller is a popular controller in the industry, with a simple design but an effective solution to many different control problems. The adjustment of three parameters $K_p$, $K_i$, $K_d$ of the PID controller suitable for each control object is a problem for the designer. Most of these parameters are determined based on experience and are "trial and error", so it will take a lot of time to adjust when encountering complex objects that the results will not be optimal. Jia-Jun Wang [1] combined two one-in-one-output PID controllers to obtain a compromise controller and successfully applied it to a rotating inverted pendulum system. However, the parameters of this compromise controller can only be determined experimentally. According to Nguyen Van Dong Hai - Ngo Van Thuyen [2] has combined a static PID and a PID-neuron controller whose parameters are updated online and has been effectively applied in controlling the inverted pendulum, but since this alternative still uses a static PID, the control quality will not be optimal.

In this study, the author proposes to research and apply the swarm algorithm to simultaneously optimize the parameters of 2 PID controllers and control applications for the inverted pendulum. This is an object with high nonlinearity, so the successful control of this object by the PSO-PID controller is scientifically meaningful and highly practical. Control methods can be applied to many objects.

## 2. Inverted Pendulum Control System

### A. *Mathematical model of an inverted pendulum system*

The inverted pendulum system consists of a pendulum of mass m, length 2L that can rotate freely, the angle of the pendulum to the vertical is α, the pendulum is attached to a horizontal bar of length r. The DC servo motor is used to move the horizontal bar in both forward and reverse directions with an angle θ [3], shown in Figure 1.

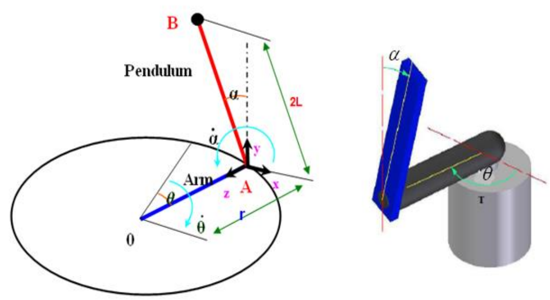



Fig. 1. Simple model of an inverted pendulum

The system of equations describing the nonlinear dynamic characteristics of the system,

$$\begin{cases} \ddot{\theta} = \frac{1}{a}.\left[b.cos\alpha.\ddot{\alpha} - b.sin\alpha.\dot{\alpha}^2 - e\dot{\theta} + f.V_m\right] \\ \ddot{\alpha} = (1/c).\left(d.sin\alpha + b.cos\alpha.\ddot{\theta}\right) \end{cases} \quad (1)$$

with
$\alpha = J_{eq} + mr^2 + J_m$;
$b = m.L.r$;

*Corresponding author: lengocdung@dntu.edu.vn

$$e = (B_{eq} + \mu_m.K_M.\frac{K_E}{R_m});$$
$$c = \frac{4mL^2}{3};$$
$$d = mgL; f = \mu_m.\frac{K_m}{R_m}$$

With a small angle α ($\alpha \approx 0,\ \dot{\alpha} \approx 0, sin\alpha \approx \alpha, cos\alpha \approx 1$), linearizing system (1), we have:

$$\begin{cases} \ddot{\theta} = \frac{1}{a}.[\, b.\ddot{\alpha} - e\dot{\theta} + f.V_m] \\ \ddot{\alpha} = (1/c).(d.\alpha + b.cos\alpha.\ddot{\theta}) \end{cases} \tag{2}$$

Taking Laplace on both sides of equation (3.2), we have:

$$\begin{cases} as^2\Theta(s) - bs^2A(s) - es.\Theta(s) = f.V_m(s) \\ -bs^2\Theta(s) + cs^2A(s) - d.A(s) = 0 \end{cases} \tag{3}$$

The transfer function of the linear model

$$\frac{A(s)}{V_m(s)} = \frac{bfs}{(ac-b^2)s^3 + ces^2 - ads - de} \tag{4}$$

The system of state variable equations of the linear model is obtained by solving the system (2.1) with two unknowns (θ) and α.

$$\begin{cases} \ddot{\theta} = \frac{1}{ac-b^2}\left(bd\alpha - ce\dot{\theta} + cfV_m\right) \\ \ddot{\alpha} = \frac{1}{ac-b^2}\left(ad\alpha - be\dot{\theta} + bfV_m\right) \end{cases} \tag{5}$$

Inference: The system of state variable equations of the rotary inverted pendulum system

$$\begin{cases} \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-ce}{ac-b^2} & \frac{bd}{ac-b^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-be}{ac-b^2} & \frac{ad}{ac-b^2} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \alpha \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ cf \\ 0 \\ bf \end{bmatrix} V_m \\ \begin{bmatrix} \theta \\ \alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \alpha \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{cases} \tag{6}$$

### *B. Modeling a rotating inverted pendulum system in Matlab Simulink*

Based on the mathematical equations of the rotating inverted pendulum system to build an inverted pendulum model in simulink as shown in Figure 2.

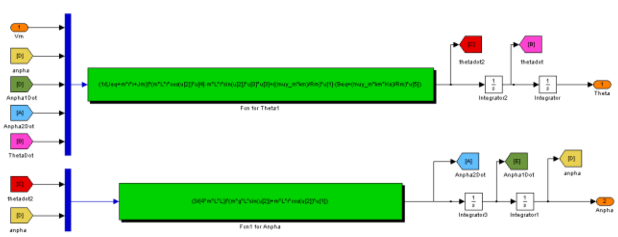


Fig. 2. Modeling a rotating inverted pendulum system in Matlab Simulink

The parameters of the system used for simulation are taken according to Table 1 below:

Table 1
Parameter values of the inverted pendulum model

| Symbol | Desciption | Values |
|---|---|---|
| ***m*** | Pendulum mass | 0.125 |
| ***L*** | Half-length of pendulum | 0.1675 |
| ***r*** | Arm's length | 0.215 |
| $J_{eq}$ | Equivalent moment of inertia | 0.0035842 |
| ***g*** | Gravity acceleration | 9.81 |
| $B_{eq}$ | Equivalent drag coefficient | 0.004 |
| $\mu_m$ | Engine performance | 0.69 |
| $K_E$ | Electromotive force constant | 0.00767 |
| $K_M$ | Moment constant | 0.00767 |
| $R_M$ | Armature resistance | 2.6 |

### *C. Schematic diagram of the control of an inverted pendulum*

Simulink simulation model using Matlab Simulink as figure 3 [3].

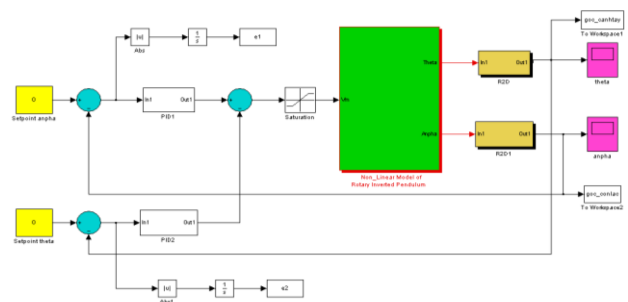


Fig. 3. Control diagram to keep the balance of the forward pendulum for both angles α and θ

Combining the arm angle PID controller and the pendulum angle PID controller we have:

$PID_\alpha = I_{out\,(\alpha)} + P_{out\,(\alpha)} + D_{out\,(\alpha)}$

$PID_\theta = I_{out\,(\theta)} + P_{out\,(\theta)} + D_{out\,(\theta)}$

The control voltage is calculated by the following formula:

$U_{dk} = U_{PID\alpha} - U_{PID\theta}$

In this case, the arm angle θ and pendulum angle α are sent to the two PIDs to control so that the pendulum is kept in the equilibrium position, and the angle of the pendulum arm is kept stable.

### *D. Control system of the inverted pendulum using a PSO-PID controller*

#### *1) PSO algorithm*

PSO is a parallel search technique, which is initialized with a random group of individuals (solutions) and then finds the optimal solution by updating generations. In each generation, each instance is updated to the two best values. The first value is the best solution obtained so far, called $P_{Pbest}$. Another optimal solution that this individual follows is the global optimal solution $P_{Gbest}$, which is the best solution that the neighbor of this individual has achieved up to the present time

[4], [5]. In other words, each individual in the population updates its position according to its best position and that of the individual in the population up to the present time as shown in Figure 4.

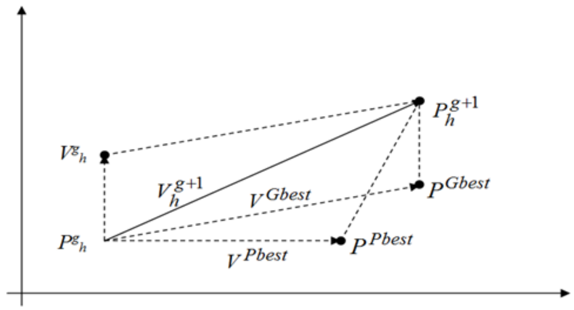



Fig. 4. Location update map of each PSO instance

The velocity and position of each individual are calculated as follows:

$$V_h^{g+1} = \omega V_h^h + C_1.rand_1().\left(P^{Gbest} - P_h^g\right) + C_2.rand_2().(P_h^{Pbest} - P_h^g)$$

$$P_h^g = P_h^g + V_h^{g+1}$$

In which,

Table 2

| | |
|---|---|
| $P_h^g$ | Individual h position at generation g |
| $V_h^g$ | Individual velocity h, at generation g |
| $P_h^{g+1}$ | Individual h position in generation g+1 |
| $V_h^{g+1}$ | Individual velocity h, at generation g + 1 |
| $P_h^{Pbest}$ | The best position of the h instance |
| $P^{Gbest}$ | The best position of an individual in a population |
| $C_1$, $C_2$ | Acceleration coefficients |
| $\omega$ | Inertial weight |
| $rand_1$, $rand_2$ | Random number between 0 and 1 |

- Diagram for controlling the inverted pendulum using the PSO-PID controller

+ Objective function (adaptation function):

The objective function uses the Integral of the Absolute magnitude of the Error (IAE) criterion to evaluate and find the optimal parameters for the two PID controllers.

$$IAE = \int_0^{t_f} |e(t)|dt$$

From the control diagram in Figure 3, the objective function is determined:

$$IAE = \int_0^5 |e_1|dt + \int_0^5 |e_2|dt$$

In that: $e_1$ and $e_2$ are the errors at the input of the two PID controllers, respectively. $P^g_h$ individual h in the g generation. Simulation time $t_f = 5$s.

At the same time, the author used the following formula to evaluate the convergence speed of the algorithm

$$\delta = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(IAE_{P_i} - \mu)^2}$$

$$\mu = \frac{\sum_{i=1}^{n} IAE_{p_i}}{n}$$

In which: $\sigma$ is the convergence rate of the algorithm, $IAE_{pi}$ is the objective function value of the i-th individual, n is the size of the swarm.

+ Diagram of control system and control algorithm PSO.

Optimizing PID controller parameters using PSO algorithm:

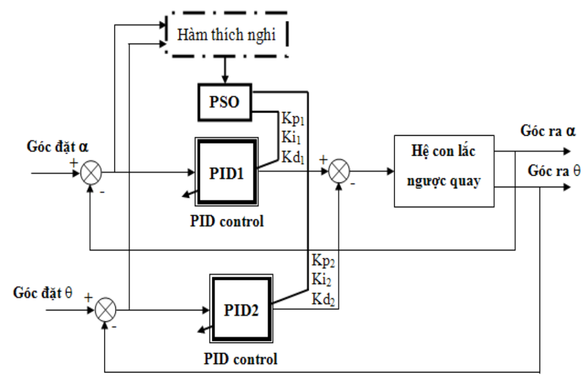



Fig. 5. PSO-PID control system

For the inverted pendulum system using feedback of both angles α and θ, it is necessary to use 2 corresponding PID controllers. The PSO algorithm will contain 6 parameters for 2 controllers including $Kp_1$, $Ki_1$, $Kd_1$ and $Kp_2$, $Ki_2$, $Kd_2$.

Algorithm flowchart of PSO-PID control system:

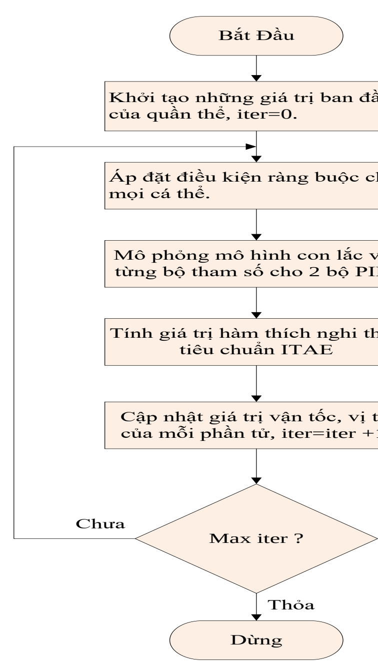



Fig. 6. Algorithm flowchart of PSO-PID control system

### 3. Simulation Results

Each value of the parameters in the PSO algorithm affects the quality of the control system. In this section, the researcher presents some cases with different values of selected control parameters, the results are as shown in Figure 7 and Figure 8.

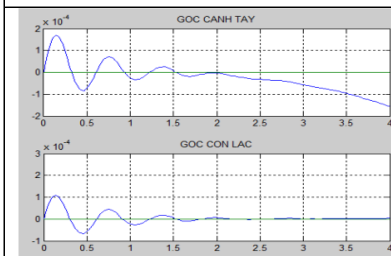
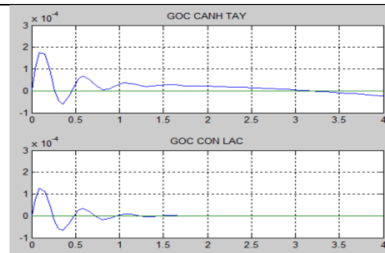
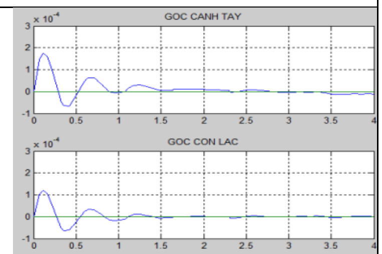
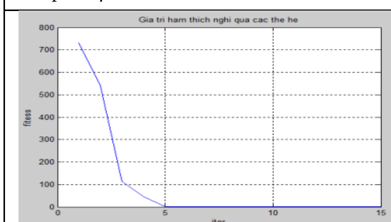
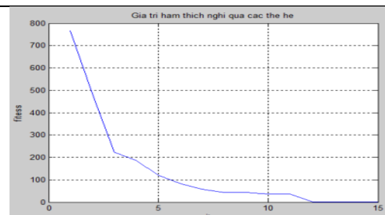
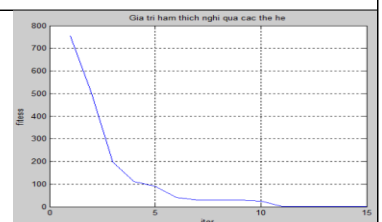
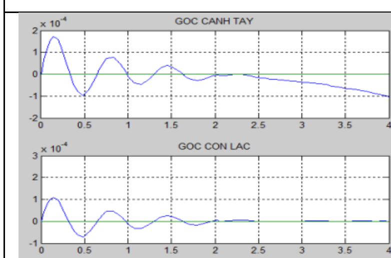
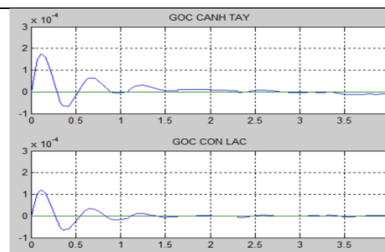
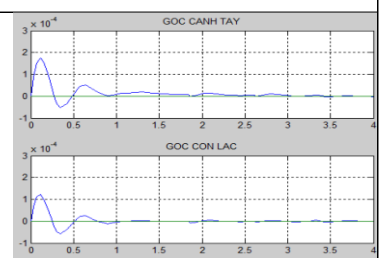
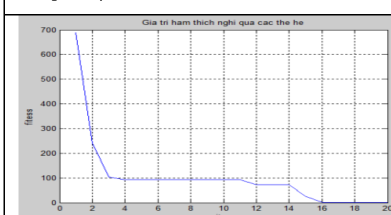
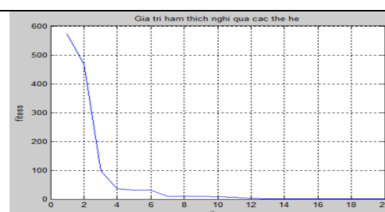
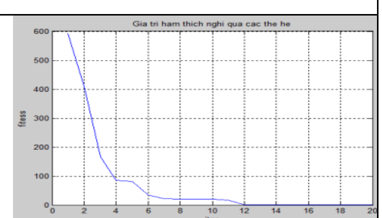
<table>
<tr><td rowspan="2">Number of Generation =15</td><td colspan="3">Swarm size</td></tr>
<tr><td>N=20</td><td>N=30</td><td>N=50</td></tr>
<tr><td rowspan="4">c1=1,5<br>c2=1,5<br>ω=0,9</td><td colspan="3">Arm angle and pendulum angle after adjustment</td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td colspan="3">Adaptive function</td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td rowspan="2">Number of Generation =20</td><td colspan="3">Swarm size</td></tr>
<tr><td>N=20</td><td>N=30</td><td>N=50</td></tr>
<tr><td rowspan="4">c1=1,5<br>c2=1,5<br>ω=0,9</td><td colspan="3">Arm angle and pendulum angle after adjustment</td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td colspan="3">Adaptive function</td></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
</table>

Fig. 7. Simulation result of angle θ and α after 15 and 20 generation

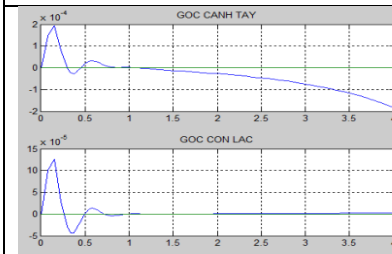
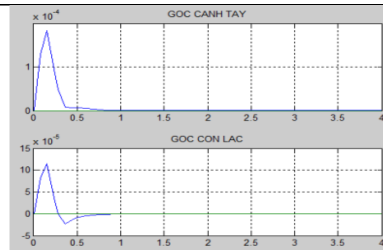
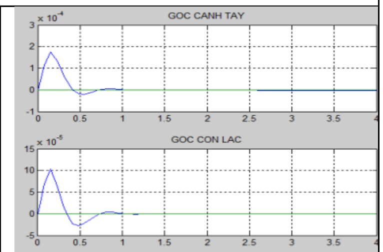
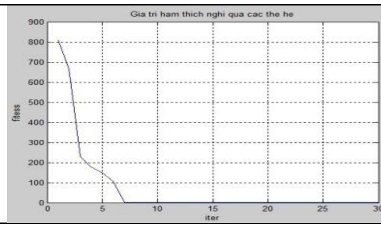
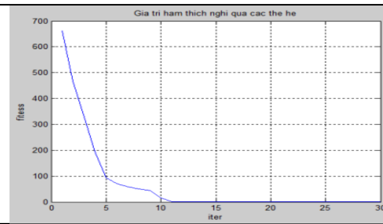
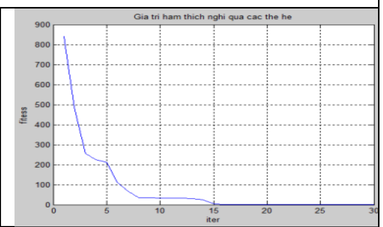
| Number of Generation =30 | Swarm size | | |
|---|---|---|---|
| | N=20 | N=30 | N=50 |
| c1 = 1,5<br>c2 = 1,5<br>ω = 0,9 | Arm angle and pendulum angle after adjustment | | |
| |  |  |  |
| | Adaptive function | | |
| |  |  |  |
| Parameters of 2 PID controllers after tuning for good response | | Kp1=30.653; Ki1=1.844; Kd1=3.879; Kp2=8.447; Ki2= 0.630; Kd2= 2.939; | Kp1=25.518; Ki1=-6.16; Kd1=3.664; Kp2=6.198; Ki2=-1.893; Kd2=2.55; |

Fig. 8. Simulation result of angle θ and α after 30 generation

## 4. Conclusion

The paper presented how to optimize the parameters of the PID controller using the swarm optimization algorithm applied in the control of the rotary inverted pendulum. The simulation process also proved that the parameters of the algorithm have a great influence on the process of optimizing the parameters of the PID controller as well as the time to reach the desired objective function value, specifically:

- When increasing the number of individuals from 20, 30 and 50 and keeping the number of generations the same, the Adaptation function converges quickly, the arm angle θ and the pendulum angle α also gradually increase the stability.
- When increasing the number of generations from 15, 20 and 30 and keeping the number of individuals the same, the Adaptation function converges slowly, the arm angle and the pendulum angle also increase in stability but slower than keeping the same number generations and increase the number of individuals.
- If simultaneously increasing the number of generations 15, 20 and 30 corresponding to the number of individuals of 20, 30 and 50, the Adaptation Function converges very quickly, the arm angle θ and the pendulum angle α stabilize faster.

From this simulation result, it can be applied to control a class of correlated nonlinear objects.

## References

[1] Wang, Jia-Jun (2011), "Simulation Studies of Inverted Pendulum based on PID Controller," *Simulation Modelling Practice and Theory*", vol. 19, no. 1, pp. 440-449.

[2] Nguyễn Văn Đông Hải, Ngô Văn Thuyên (2012), Xây dựng bộ điều khiển PID-Neuron cho hệ con lắc ngược quay, Tạp chí khoa học giáo dục kỹ thuật, Trường Đại học Sư phạm Kỹ thuật Tp. Hồ Chí Minh.

[3] Iraj Hassanzadeh and Saleh Mobayen (2008), PSO-Based Controller Design for Rotary Inverted Pendulum System.

[4] Gaing, Z.L. (2004), "A particle swarm optimization approach for optimum design of PID controller in AVR system", *IEEE Transactions On Energy Conversion*, vol. 19, no.2, pp. 284-291.

[5] Kennedy J. and Eberhart R (1995), "Particle swarm optimization", *Proc. IEEE Int. Conf. Neural Networks*, vol. 4, Perth, Australia, pp. 1942-1948.