iCBST 2014 International Conference on Business, Science and Technology which will be held at Hatyai, Thailand on the 25th and 26th of April 2014.

Australian Journal of Basic and Applied Sciences, 8(5) Special 2014, Pages: 370-374

AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

A Preliminary Report on the Utilization of PSO for Solving the Hamiltonian Systems

Irianto

Universiti Teknikal Malaysia Melaka, Faculty of Engineering Technology, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

ARTICLE INFO

Article history:
Received 25 January 2014
Received in revised form 12 March 2014
Accepted 14 April 2014
Available online 25 April 2014

Keywords:
Particle swarm optimization, Optimal control problems, Hamiltonian system

ABSTRACT

When one uses the Pontryagin’s Maximum Principle for solving fixed-time and fixed-endpoint optimal control problems, one will face a Hamiltonian system. The Hamiltonian system consists of a pair of differential equations. The first equation is equipped with initial and final condition, but the second one lacks any boundary conditions. Thus, in most cases, one cannot solve this problem directly. This is a classic difficulty for using the maximum principle. We will proposed a new method for overcoming this difficulty here. This method utilizes an algorithm called Particle Swarm Optimization or PSO. At the end this paper will present some numerical results.

To Cite This Article: Irianto., A Preliminary Report on the Utilization of PSO for Solving the Hamiltonian Systems. ***Aust. J. Basic & Appl. Sci.,*** *8(5): 370-374, 2014*

INTRODUCTION

Optimum control problem have several classifications; one of them is fixed-time and fixed-endpoint problem. This type of optimum control problem is interesting, because if we try to solve it with Hamiltonian system it will gives the state condition with a very strict boundary and leaves the co-state without any boundary at all. Those type of system is very hard to solved, and that is why many scientists and researchers doing a lot of research to solve this problem.

Regarding to those issue we would like to introduce an application of Particle Swarm Optimization application to solve that kind of optimal control problem. Particle Swarm Optimization, Kennedy and Eberhart \cite{jour}, is an algorithm to find an optimum value of non-linear function based on the social behavior of animals that live in groups.

Optimal Control Problem:

Define an autonomous linear or nonlinear problem with fixed-time and fix-endpoint without terminal cost. The state equation given as follow:

$$\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y},\boldsymbol{u}) \tag{1}$$

Where $\boldsymbol{y}$ is the n-dimensional state vector, whose components y_i, $i = 1,2,\dots,n$, are the state variables, and $\boldsymbol{u}$ is the control vector in m-th dimension, whose components u_j, $j = 1,2,\dots,m$, are the continuous control functions to t. The dot denotes a time derivative and $\boldsymbol{f}$ has n components f_i. The initial and final conditions are $\boldsymbol{y}(0) = \boldsymbol{y}^0$, and $\boldsymbol{y}(t_1) = \boldsymbol{y}^1$, with the end time t_1 is fixed.
The cost function is given:

$$J = \int_0^{t_1} f_0(\boldsymbol{y},\boldsymbol{u})d\tau \tag{2}$$

With f_0 is a continuous and differentiable function. The objective of this optimal control problem is to find $u \in U$ such that the state system can be steered from $\boldsymbol{y}^0$ to $\boldsymbol{y}^1$ in t_1 time and in the same time also minimize the cost J. The $y_i(t)$ curves using optimum control $\boldsymbol{u}$ are the optimum curves and the J value that corresponds is the optimum cost.

The first step is to replace the cost integral by introducing an additional state variable y_0 which satisfies the state equation

$$y_0 = f_0(\boldsymbol{y},\boldsymbol{u}), \qquad y_0(0) = 0 \tag{3}$$

Corresponding Author: Irianto, Faculty of Engineering Technology, Universiti Teknikal Malaysia Melaka.
E-mail: http://www.utem.edu.my/ftk/ or irianto@utem.edu.my

The next step we introduce the *extended state vector* $\hat{\boldsymbol{y}}$, of dimension $n+1$, whose components are y_i, $i = 0,1,\dots,n$. If we define the extended vector $\hat{\boldsymbol{f}}$ similarly, then the state equation can be rewrite as follow

$$\dot{\hat{\boldsymbol{y}}} = \hat{\boldsymbol{f}}(\boldsymbol{y},\boldsymbol{u}) \tag{4}$$

So we can get the Hamiltonian function $\mathrm{H}(\hat{\boldsymbol{y}},\hat{\boldsymbol{z}},\boldsymbol{u})$

$$\mathrm{H} = \hat{\boldsymbol{z}}^T\dot{\hat{\boldsymbol{y}}} = \sum_{i=0}^{n} \hat{z}_i f_i \tag{5}$$

with $\hat{\boldsymbol{z}}$ is a $n+1$-dimensional *co-state* vector. From this Hamiltonian we can get the co-state equations as follow:

$$\dot{\hat{\boldsymbol{z}}} = -\frac{\partial \mathrm{H}}{\partial \boldsymbol{y}} \tag{6}$$

Because H is not depending on y_0 then Eqn 6 can be rewrite as follow:

$$\dot{\hat{z}}_0 = 0, \qquad \dot{\hat{z}}_i = -\frac{\partial \mathrm{H}}{\partial y_i}, \qquad i = 1,2,\dots,n \tag{7}$$

The problem is, when we're tried to apply *Pontryagin Maximal Principle* (PMP) [1, 2] to solve the system is that the co-state equations will be left without any initial conditions. As stated before, this problem is hard to solve for the exact solution, therefore it needs a computational method to solve the problems. One we propose in this paper is the use of Particle Swarm Optimization.

Particle Swarm Optimization:

Particle swarm optimization (PSO) was designed by Kennedy (Social Psychology) and Eberhart (Engineering). This method is inspired by the social behavior of animals that live in groups [4].

In PSO we generate randomly a population of agents and put them in the search space domain of an objective function. Those agents have memories about their position and velocity; and its keep track of their position associated with the fitness it has achieved so far. Every agent flies in the search space by combining memories of their best position and the best position of whole agents. Eventually as a group whole agents will arrived to one best position in the search space.

Every agent in PSO is representation of three vectors in d-dimensional search space. This three vectors are agent position $\boldsymbol{x}_i$, agent bests position $\boldsymbol{pb}_i$ and agents velocity $\boldsymbol{v}_i$.

The idea is to move every agents based on their best position and the swarm best position. The velocity of movement is formulated as:

$$\boldsymbol{v}_i^{j+1} = \omega \boldsymbol{v}_i^j + \eta_1 R_1\left(\boldsymbol{pb}_i^j - \boldsymbol{x}_i^j\right) + \eta_2 R_2\left(\boldsymbol{gb}^j - \boldsymbol{x}_i^j\right) \tag{8}$$

Where $\boldsymbol{v}_i^{j+1}$ is the velocity vector of i-th agent at $j+1$-th iteration, η_1 is personal variable and η_2 is swarm variable. R_1 and R_2 are random numbers that are generated during the process. $\boldsymbol{pb}_i^j$ is best position vectors of the i-th agent at j-th iteration; and $\boldsymbol{gb}^j$ is the swarm best position vector at j-th iteration. ω is inertia weight, which is a linear function by iteration and can be formulated as:

$$\omega = K_2 - \frac{(K_2 - K_1)i}{N} \tag{9}$$

With K_1 and K_2 are the lower and upper bound of inertia weight, i is the number of iteration and N is the total number of iteration.

After the velocity vector has been calculated then we change position of the agents by:

$$\boldsymbol{x}_i^{j+1} = \boldsymbol{x}_i^j + \boldsymbol{v}_i^{j+1} \tag{10}$$

PSO algorithm:

1. Generate n number of agents with random positions and velocities on d-dimensions,
2. Evaluate the objective function in d-variables,
3. Compare evaluation with agent's previous best position ($\boldsymbol{pb}_i^{j-1}$): if current value $< \boldsymbol{pb}_i^j$, then $\boldsymbol{pb}_i^j = =$ current value and $\boldsymbol{x}_i^j$ = current position,
4. Compare evaluation with swarm previous best position ($\boldsymbol{gb}^{j-1}$): if current value $< \boldsymbol{gb}^{j-1}$, then $\boldsymbol{gb}^j =$ current value,
5. Change the velocity by the formula

$$\boldsymbol{v}_i^{j+1} = \omega \boldsymbol{v}_i^j + \eta_1 R_1\left(\boldsymbol{pb}_i^j - \boldsymbol{x}_i^j\right) + \eta_2 R_2\left(\boldsymbol{gb}^j - \boldsymbol{x}_i^j\right) \tag{11}$$

6. Move agents with $\boldsymbol{x}_i^{j+1} = \boldsymbol{x}_i^j + \boldsymbol{v}_i^{j+1}$, goto 2nd step until stop criteria is met.

Optimal Control Problem Solving with PSO:

A state equation for optimal control with is defined as follow:

$$\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y},\boldsymbol{u}) \tag{12}$$

With initial condition $\boldsymbol{y}(0) = \boldsymbol{y}^0$, and $\boldsymbol{y}(t_1) = \boldsymbol{y}^1$, with the end time t_1 is fixed, and the cost function define by:

$$J = \int_0^{t_1} f_0(\boldsymbol{y}, \boldsymbol{u}) d\tau \tag{13}$$

so with Pontryagin Maximal Principle we can get the state and co-state equations as follow:

$$\begin{aligned} &\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}), \qquad \boldsymbol{y}(0) = \boldsymbol{y}^0, \qquad \boldsymbol{y}(t_1) = \boldsymbol{y}^1 \\ &\dot{\boldsymbol{z}} = \boldsymbol{g}(\boldsymbol{y}, \boldsymbol{z}) \end{aligned} \tag{14}$$

As said before, it is hard to find the exact solution for Eqn 14 because of lack of initial condition in the co-state.

If $\boldsymbol{z}^0 \in \mathbb{R}^n$ is a random initial condition vectors for Eqn 14,

$$\boldsymbol{z}^0 = \begin{pmatrix} z_1(0) \\ \vdots \\ z_i(0) \end{pmatrix} = \begin{pmatrix} z_1^0 \\ \vdots \\ z_i^0 \end{pmatrix} \tag{15}$$

then Eqn 14 will become:

$$\begin{aligned} &\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}), \qquad \boldsymbol{y}(0) = \boldsymbol{y}^0 \\ &\dot{\boldsymbol{z}} = \boldsymbol{g}(\boldsymbol{y}, \boldsymbol{z}), \qquad \boldsymbol{z}(0) = \boldsymbol{z}^0 \end{aligned} \tag{16}$$

Because all the initial conditions of Eqn 16 are complete, now we can solve it analytically or numerically. Let

$$\boldsymbol{y}^*(t), \qquad \boldsymbol{z}^*(t) \tag{17}$$

be the solution of Eqn 16, with $\boldsymbol{y}^*(t)$ is solution for the state and $\boldsymbol{z}^*(t)$ solution for the co-state, then for $t = t_1$ we will get

$$\boldsymbol{y}^*(t_1) = \boldsymbol{y}^{*1} \tag{18}$$

And now let we define a new function F as a distance function between $\boldsymbol{y}^{*1}$, the value of $\boldsymbol{y}$ of Eqn 16 at $t = t_1$, and $\boldsymbol{y}^1$, the value of $\boldsymbol{y}$ of Eqn 14 at $t = t_1$, such that

$$F(\boldsymbol{y}^{*1}, \boldsymbol{y}^1) = \|\boldsymbol{y}^{*1} - \boldsymbol{y}^1\| \tag{19}$$

The problem now is how to find $\boldsymbol{z}^0 \in \mathbb{Z} \subset \mathbb{R}^n$ such that F is minimize, with $\mathbb{Z}$ is the domain of F.

If we see this problems from PSO point of view, then we can spread agents in domain$\mathbb{Z} \subset \mathbb{R}^n$, and we move the all agents using particle swarm optimization principles such that $F(\boldsymbol{y}^{*1}, \boldsymbol{y}^1)$ is minimize.

RESULT AND DISCUSSION

In this paper we try to use PSO to solve unicycle problem. Let $q = (x, y, \theta)$ be the generalized coordinates of unicycle, where (x, y) is the Cartesian position of the unicycle and θ is its orientation with respect to the x axis. The kinematic model of the system is

$$\begin{aligned} &\dot{x} = v \cos\theta \\ &\dot{y} = v \sin\theta \\ &\dot{\theta} = w \end{aligned} \tag{20}$$

Where v and w are the control variables respectively known as the driving and steering velocity inputs. The goal of this control is to carry a unicycle from point A to point B in some sort of time and also minimize the cost. In this case we would like to take the unicycle from $(10,7)$ to $(0,0)$ of Cartesian coordinate in 5 second. And the cost function is:

$$J = \frac{1}{2} \int_0^5 (v^2 + w^2) d\tau \tag{21}$$

Then we can get the Hamiltonian function

$$\mathrm{H} = -\frac{1}{2}(v^2 + w^2) + z_1(v\cos\theta) + z_2(v\sin\theta) + z_3 w \tag{22}$$

With Pontryagin Maximal Principle the state and co state system has the form

$$\begin{aligned} &\dot{x} = v\cos\theta, \qquad x(0) = 10, \qquad x(5) = 0 \\ &\dot{y} = v\sin\theta, \qquad y(0) = 7, \qquad y(5) = 0 \\ &\dot{\theta} = w, \qquad \theta(0) = 0, \qquad \theta(5) = 0 \\ &\dot{z}_1 = 0 \\ &\dot{z}_2 = 0 \\ &\dot{z}_3 = z_1 v \sin\theta - z_2 v \cos\theta \end{aligned} \tag{23}$$

To find the optimum solution using Pontryagin Maximal Principle then

$$\begin{aligned} &\frac{\partial \mathrm{H}}{\partial v} = 0 \\ &v = z_1\cos\theta + z_2\sin\theta \end{aligned} \tag{24}$$

and

$$\frac{\partial \mathrm{H}}{\partial w} = 0 \qquad (25)$$
$$w = z_3$$

If we substitute Eqn 24 and Eqn 25 into Eqn 23, then we will get

$$\begin{aligned}
&\dot{x} = (z_1 \cos\theta + z_2 \sin\theta)\cos\theta, \quad x(0) = 10, \quad x(5) = 0\\
&\dot{y} = (z_1 \cos\theta + z_2 \sin\theta)\sin\theta, \quad y(0) = 7, \quad y(5) = 0\\
&\dot{\theta} = z_3, \quad \theta(0) = 0, \quad \theta(5) = 0\\
&\dot{z}_1 = 0\\
&\dot{z}_2 = 0\\
&\dot{z}_3 = (z_1^2 - z_2^2)\sin\theta\cos\theta - z_1 z_2(2\sin^2\theta - 1)
\end{aligned} \qquad (26)$$

Now we can applied PSO to Eqn 26 to find the solution of optimal control problem. In this paper the domain are z_1, z_2 and z_3. The number of agents are 100 and the number of iterations are 50 times. The domains are $z_1 \in [-10,0]$, $z_2 \in [-10,0]$ and $z_3 \in [0.10]$. The result is given as the following table and figures.

Table. 1: Comparison several result of unicycle optimum control problem with PSO.

Iteration	$\boldsymbol{x}$	$\boldsymbol{y}$	$\boldsymbol{\theta}$
10	0.0457	0.0322	-0.0915
20	-0.0383	-0.0115	-0.0430
30	0.0387	0.0018	0.0391
40	-0.0197	-0.0182	-0.0329
50	0.0192	0.0096	-0.0328

Table 1 shows us several results for several iterations when we applied PSO into the problem. i.e the first row is shows a result of 10 iteration, it is shown that the x, y, and θ at the end of iteration is close to zero, which are our expected endpoint. And it is also shows that the bigger the iteration the more accurate result we will get, by mean that all the variables are closer into the endpoint that we're expected.

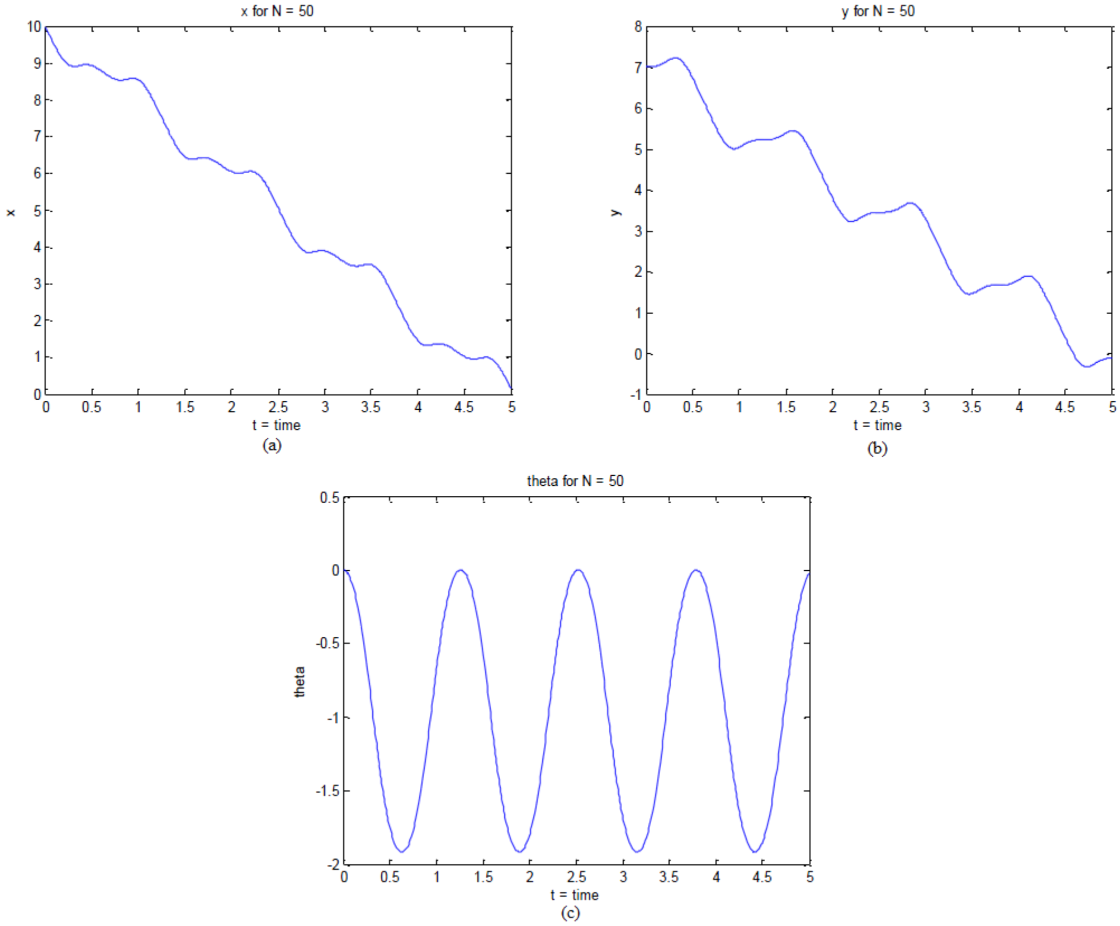

Fig. 1: (a) x curve by time (b) y curve by time (c) θ curve by time.

As our needs in this example is to take the unicycle from $(10,7)$ to $(0,0)$ in 5 time unit and the orientation, θ, start from 0^o and end in 0^o also in 5 time unit. Figure 1 gave the x, y, and θ curves, and they shows that the end point for all variables are achieved in 5 time unit. For example (a) shows the curve of x variable start from 10 at $t = 0$ and will end up at 0 when $t = 5$, etc.

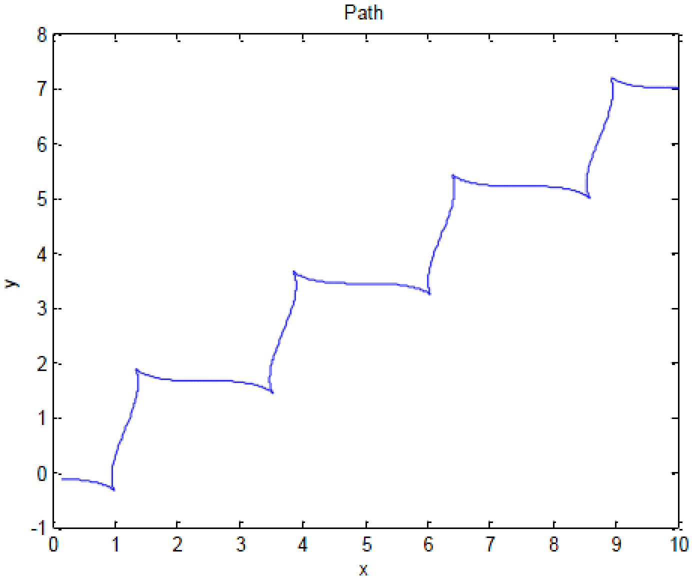

Fig. 2: Unicycle path.

Figure 2 shows us the path that unicycle take in 5 time unit. It is the optimum path when we try to solve the Eqn 23 and minimize Eqn 21, it start at $(10,7)$ and ends at approximately $(0,0)$.

Conclusion:

Particle swarm optimization has been developing since its first appearance. One of the developments is by using PSO for solving control optimal problems. The proposed approach of applying PSO for optimum control problem gave our example a convergence numerical result. Although the convergence hasn't been proved sufficiently by theoretical base, our numerical shows that the performances of our approach are good. It can be seen by the Table 1 that just for 10 iteration we already got a good result, but of course we've made some assumptions such as we already know the domain of our search space.

For the next improvement of this paper, we can try to prove this approach theoretically and maybe also a comparison with other numerical methods such as genetic algorithm.

ACKNOWLEDGEMENT

The author is grateful to the Faculty of Engineering Technology for providing the facilities to carry out the research.

REFERENCES

Boyce, W.E. and R.C. DiPrima, 2001. Elementary Differential Equations and Boundary Value Problems. Wiley, US.

Chen, X. and Y. Li, 2008. On Convergence Parameter Selection of an Improved Particle Swarm Optimization. International Journal of Control and System, 6: 559-570.

Eberhart, R. and J. Kennedy, 1995. A New Optimizer Using Particle Swarm Theory. The Sixth International Symposium on Micro Machine and Human Science, 39-43.

Eberhart, R. and Y. Shi, 1998. A Modified Particle Swarm Optimizer. IEEE International Conference on Evolutionary Computation, 69-73.

Eberhart, R. and Y. Shi, 2009. Monitoring of Particle Swarm Optimization. Front. Comput. Sci. China, 3(1): 31-37.

Hocking, L.M., 1991. Optimum Control An Introduction to the Theory with Applications. Clarendon Press, New York.

Jurdjevic, V., 1996. Geometric Control Theory. Cambridge University Press.

Poli, R., J. Kennedy and T. Blackwell, 2007. Particle Swarm Optimization An overview. Swarm Intelligent, 1: 33-57.