# Particle Swam Optimization for Stabilizing Controller of a Self-erecting Linear Inverted Pendulum

Mahmud Iwan Solihin, *Member IEE* and Rini Akmeliawati, *Member IEE*

*Abstract*—**This paper presents an advanced application of particle swarm optimization, PSO to find state feedback controller gains for stabilizing controller in a linear inverted pendulum. This plant is used as an application example of the proposed method. In conventional method of state feedback control design such as pole placement and linear quadratic regulator method, controller designers often face troublesome exercise of tuning several parameters. Particularly, one has to face trial-and-error approach to select suitable Q and R matrices to design a state feedback control using linear quadratic regulator method. To overcome this problem, an intelligent approach employing PSO-based constrained optimization is proposed. The objective of the optimization is to minimize error function, while closed loop poles region is incorporated as an optimization constraint whose parameter is selected based on the desired control performance. In this study, Clerc's PSO is adopted together with dynamic objective constraint handling where efficient optimization run is shown in the simulation results.**

*Index Terms*—**PSO, optimization, state feedback controller, inverted pendulum.**

## I. Introduction

Swinging up an inverted pendulum is a classic and challenging control problem in the field of nonlinear control theory. It is also useful to demonstrate concepts such as stabilization of an unstable system. A cart-driven inverted pendulum has a structure where the pendulum is hinged to the cart via a pivot and only the cart is actuated. The motion of the pendulum has to be controlled by moving the cart back and forth within a limited travel of the cart [1].

There have been a lot of studies in this subject [2-7]. Basically, inverted pendulum control is composed of the swing-up control of the pendulum and the stabilizing (tracking) control of the whole system that consists of angular control of the pendulum at upright position and position/tracking control of the cart on the rail. First, swing-up control is to raise the pendulum from the downwards position to the upright position. This is achieved when the motor is given voltage in the appropriate direction and magnitude to drive the cart back and forth along the limited rail length repeatedly until the pendulum is close to the upright position. The swing-up method used is commonly energy based method [1,8]. Thereafter, stabilizing control is to balance the pendulum in the upright position. This control algorithm has also to maintain the pendulum upright while the cart tracks the position reference trajectory.

This paper will only focus on the stabilizing controller. State feedback controller with integral action is used for stabilizing controller. The controller is designed based on the linearized state space model of the system. Although the problem of feedback control design is conventionally handled by pole placement method or LQR method via Riccati equation [9], they still possess trial and error approach of parameter adjustment. Particularly, choosing elements of Q

Manuscript received March 15, 2010. This work was partially supported by the Ministry of Science and Technology Malaysia under eSciencefund Grant, code: 03-01-08-SF0037.

Mahmud Iwan Solihin is with Mechatronics Department, School of Engineering, UCSI University, Malaysia (e-mail: mahmudis@ucsi.edu.my). He is currently also affiliated to Intelligent Mechatronics Systems Research Unit, Mechatronics Engineering Department, IIUM Malaysia as a postgraduate student.

Rini Akmeliawati is with Mechatronics Engineering Deparment, International Islamic University Malaysia (IIUM) (e-mail: rakmelia@iium.edu.my).

and R matrices in the feedback control design using LQR method has to be done by trial.

Therefore, to resolve this difficulty, this paper proposes a novel intelligent-based tuning of state feedback gains employing particle swarm optimization (PSO) with regional closed loop pole as a constraint. Simulation study on cart-driven inverted pendulum tracking control is carried out to evaluate the effectiveness of the proposed method.

PSO is a swarm intelligence technique and is one of the latest population-based optimization algorithms. PSO has attracted a lot of attention in recent years because of the following reasons [10,11]. First, it requires only a few lines of computer code to realize the PSO algorithm. Second, its search technique using not the gradient information but the values of the objective function makes it an easy-to-use algorithm. Third, it is computationally inexpensive, since its memory and CPU speed requirements are very low. Fourth, it does not require a strong assumption made in conventional deterministic methods such as linearity, differentiability, convexity, separability or non-existence of constraints in order to solve the problem efficiently. Finally, its solution does hardly depend on initial states of particles, which could be a great advantage in engineering design problems based on optimization approaches. PSO was first introduced in 1995 by Kennedy and Eberhart [12-16]. It was developed through simulation of a simplified swarm social behavior, and has been found to be robust in solving continuous nonlinear optimization problem. Compared with other optimization algorithm such as GA (genetic algorithm), PSO is more efficient and economic for solving min-max optimization problem [17].

The remainder of this paper is as follows. Section II presents the system description and modeling. Section III discusses the controller strategy for the swing-up and tracking system of self-erecting cart-driven inverted pendulum. Section IV presents the proposed method where PSO is adopted for state feedback gains optimization/tuning. Section V shows the simulation results and Section VI is conclusions.

## II. SYSTEM MODEL

The cart-driven inverted pendulum model is derived based on physical laws. For simulation purpose, the system model and parameters value are taken from Quanser's Linear Motion Servo Module [7]. The schematic diagram is shown in Fig.1. The parameters value and the description are presented in Table I.

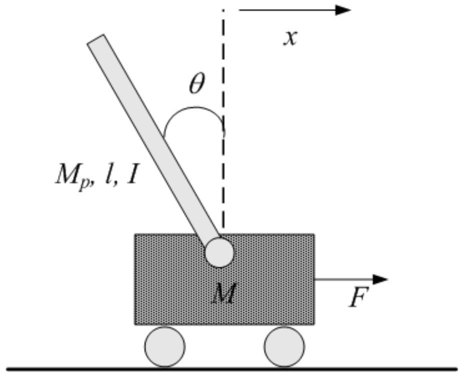


Fig. 1. Cart-driven inverted pendulum schematic

Applying Newton's second law of motion to the cart system, the equation of motion can be represented as follows:

$$M\ddot{x} = F - B_{eq}\dot{x} - F_a \quad (1)$$

where $F$ is cart driving force produced by the motor and $F_a$ is inertia force due to the armature motor rotation. The driving force acting on the cart through the motor pinion can be expressed as:

$$F = \frac{\eta_g K_g T_m}{r_m} \quad (2)$$

where $T_m$ is torque generated by the motor. Using Khirchoff 's voltage law, the following equation is obtained (by disregarding the motor inductance since $L_m << R_m$):

$$i_m = \frac{V - E_{emf}}{R} = \frac{V - K_m \omega_m}{R} \quad (3)$$

TABLE I
List of parameters

| Symbol | Description | Value/Unit |
|---|---|---|
| $R$ | Motor armature resistance | 2.6 Ω |
| $L$ | Motor armature inductance | 0.18 mH |
| $K_t$ | Motor torque constant | 0.00767 Nm/A |
| $\eta_m$ | Motor efficiency | 100% |
| $K_m$ | Motor EMF constant | 0.00767 Ns/rad |
| $J$ | Rotor moment inertia | $3.9\text{x}10^{-7}$ kgm$^2$ |
| $K_g$ | Gearbox ratio | 3.71 |
| $\eta_g$ | Gearbox efficiency | 100% |
| $r_m$ | Motor pinion radius | $6.35\text{x}10^{-3}$ m |
| $r_p$ | Position pinion radius | $1.48\text{x}10^{-2}$ m |
| $B_{eq}$ | Equivalent viscous damping coefficient at motor | 5.4 Nms/rad |
| $B_p$ | Viscous damping coefficient at pendulum pivot | 0.0024Nms/rad |
| $l$ | Pendulum length from pivot to centre of mass | 0.3302 m |
| $I$ | Pendulum moment of inertia | $7.88\text{x}10^{-3}$ kgm$^2$ |
| $M_p$ | Pendulum mass | 0.23 kg |
| $M$ | Cart mass | 0.94 kg |
| $V_m$ | Motor nominal input voltage | 5 V |

The torque generated by the DC motor is proportional to the armature current as:

$$T_m = \eta_m K_t i_m \tag{4}$$

Substituting (4) and (3) into (2) leads to:

$$F = \frac{\eta_g \eta_m K_g K_t (V - K_m \omega_m \dot{x})}{R\, r_m} \tag{5}$$

The linear velocity of the cart and motor angular velocity can be related by:

$$\omega_m = \frac{K_g \dot{x}}{r_m} \tag{6}$$

Therefore, (5) can be rearranged to:

$$F = \frac{\eta_g \eta_m K_g K_t (r_m V - K_g K_m \dot{x})}{R\, r_m^2} \tag{7}$$

As seen at the motor pinion, the armature inertia force due to the motor rotation can be expressed as:

$$F_a = \frac{\eta_g K_g T_a}{r_m} \tag{8}$$

Applying Newton's 2nd law of motion to the motor shaft:

$$J\ddot{\theta}_m = T_a \tag{9}$$

where $\theta_m$ is motor shaft rotation angle. Moreover, the mechanical configuration of the cart's pinion system gives the following equation:

$$\theta_m = \frac{K_g x}{r_m} \tag{10}$$

Using (10) and (9) and substituting into (8), gives:

$$F_a = \frac{\eta_g K_g^2 J\ddot{x}}{r_m^2} \tag{11}$$

Finally, substituting (11) and (7) into (1), applying Laplace transform and rearranging, yields the open loop transfer function for the cart system:

$$G(s) = \frac{X(s)}{V(s)} = \frac{a_1}{(b_1 s + b_2)s} \tag{12}$$

where:

$$a_1 = r_m \eta_g \eta_m K_g K_t$$

$$b_1 = RMr_m^2 + R\eta_g K_g^2 J$$

$$b_2 = \eta_g K_g^2 \eta_m K_t K_m + B_{eq} R r_m^2$$

If the parameters related to the motor in G are unknown, then $a_1$, $b_1$ and $b_2$ can be identified experimentally. Thus, this case is considered as a practical approach of modeling. However, this is beyond the discussion in this paper.

Furthermore, the nonlinear model of the inverted pendulum motion can be derived using Lagrange's equation. For brevity, the nonlinear equation of motions can be obtained as follows:

$$(M + M_p)\ddot{x} + B_{eq}\dot{x} - M_p l\ddot{\theta}\cos\theta + M_p l\dot{\theta}^2 \sin\theta = F \tag{13}$$

$$-M_p l\ddot{x}\cos\theta + (I + M_p l^2)\ddot{\theta} + B_p\dot{\theta} - M_p gl\sin\theta = 0 \tag{14}$$

The nonlinear model can be linearized which is valid near the equilibrium point (upright pendulum) so that $\sin(\theta) \cong \theta$, $\cos(\theta) \cong 1$ and also neglecting higher order term. The linearized model is written in state space in order to allow the design of state feedback controller for upright pendulum stabilization.

$$\dot{z} = Az + Bu \quad (15)$$

$$y = Cz$$

where: $z = [x\ \theta\ \dot{x}\ \dot{\theta}]^T$ , $u = V$ and $y = [x\ \theta]^T$ .

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{(M_p l)^2}{q} & \frac{-B_{eq}(M_p l^2 + I)}{q} & \frac{M_p l B_p}{q} \\ 0 & \frac{(M+M_p)M_p gl}{q} & \frac{M_p l B_{eq}}{q} & \frac{(M+M_p)B_p}{q} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{(M_p l^2 + I)}{q} \\ \frac{M_p l}{q} \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

with $q = (M + M_p)I + MM_p l^2$ .

## III. CONTROL STRATEGY

The self-erecting inverted pendulum control strategy has two mains phases: the swing-up phase and the stabilizing (and tracking) phase. In swing-up phase, the swing-up controller sways the pendulum using a sinusoidal reference input while keeping the cart within the limited travel distance on the rail. In the second phase, where this paper is focused on, when the pendulum is close to the vertical position (about $|\theta|<25^o$), a linear state feedback controller takes over the control to stabilize the pendulum via switching algorithm. This controller should also be able to track the input reference trajectory of the cart while maintaining the pendulum upright. The control scheme of the self-erecting inverted pendulum is shown in Fig.2.

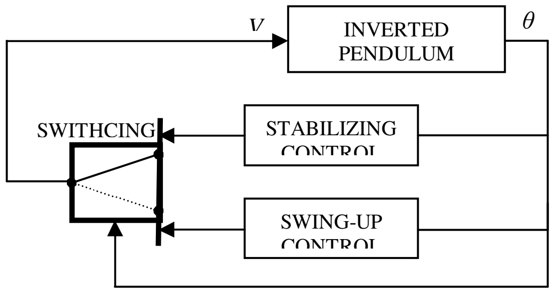


Fig. 2. Control scheme of self-erecting inverted pendulum

### A. Swing-up Controller

This controller aims at swinging up the pendulum from the rest ($\theta$=180$^o$=-180$^o$) while keeping the cart travels within the limited horizontal distance. This can be achieved by giving sinusoidal input reference to the cart as suggested by [1]. Here, to ensure the bounded travel of the cart, close loop servo control is designed based on the transfer function *G(s)*. By constructing a sinusoidal reference input, the pendulum will be swung up to reach nearly vertical posture in order to allow the stabilization controller to take over. The reference input ($x_r$) has the form:

$$x_r = K_x \sin(\omega t - \pi) \quad (16)$$

where $K_x$ is determined to limit the horizontal travel of the cart and $\omega$ is selected considering the natural frequency of the swing angle.

The close loop servo control used here is proportional velocity (PV) controller which has control law as:

$$V = K_p(x_r - x) - K_v \dot{x} \quad (17)$$

Then, the close loop transfer function of the cart servo can be expressed as:

$$\frac{X(s)}{X_r(s)} = \frac{K_p G(s)}{1 + K_p G(s) + sK_v G(s)} \quad (18)$$

This leads to a second order system as follows:

$$\frac{X(s)}{X_r(s)} = \frac{1.6K_p}{s^2 + s(12.23 + 1.6K_v) + 1.6K_p} \quad (19)$$

The desired performance is $\zeta$=0.7 and $\omega$=20rad/s. Comparing characteristic equation in (19) with the standard second order form:

$$s^2 + 2\zeta\omega s + \omega^2 \quad (20)$$

the PV controller gains are obtained as:
$K_p$=250 and $K_v$=9.86.

*B. Switching Algorithm*

A transition algorithm is designed to switch from swing-up controller to stabilizing controller. This is performed by smooth conditional switching that can be expressed as follows:
If $|\theta|>25^o$, only swing-up control is active.
If $|\theta|<20^o$, only stabilizing control is active.
If $20^o \leq |\theta| \leq 25^o$, swing-up and stabilizing control signal are averaged/weighted.
The last condition is made as transition region to avoid what known as hard switching.

*C. Stabilizing Controller*

This section is the focus of this paper. The stabilizing controller is designed based on linearized state space model in (15). It is a state feedback control with integral action for tracking system. It is assumed that the system given in (15) is completely state controllable and all state variables are available for feedback. One can use full state feedback control with integral gain as diagram shown in Fig. 3.

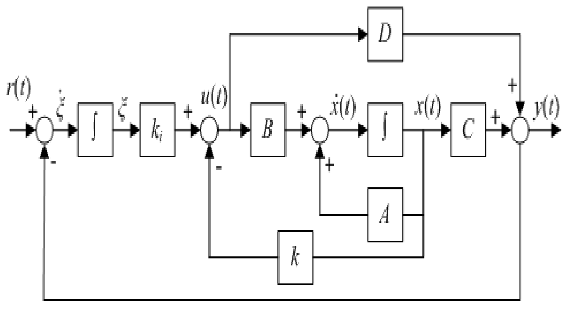


Fig. 3. State feedback control with integral gain for tracking system

Controller gains $k:=(k_1,k_2,k_3, k_4, \text{and } k_i)$ can be determined by common method such as pole placement or LQR method. However, the issue in pole placement method becomes a matter of trial and error to meet the desired response, same as choosing the appropriate element of Q and R matrices in LQR method. The problem becomes more intractable when the system has multiple outputs.

Therefore, the objective in this work is to search for a feedback controller gains to stabilize the inverted pendulum upright and after that it has also to track the input reference of the cart position while maintaining the pendulum upright. PSO-based constrained optimization is adopted in the controller gain tuning with allowable closed loop pole region as a constraint.

## IV. CONTROLLER OPTIMIZATION BY PSO

*A. Overview of PSO*

PSO is a population-based stochastic search algorithm. The basic PSO is developed from research on swarm such as fish schooling and bird flocking. PSO is popular mainly due to its simplicity in the concept and computationally efficient. Compared with other optimization algorithms such as GA, PSO is more efficient for solving min-max optimization problem [15]. Its advantages are highlighted in [16]. First, it requires only a few lines of computer code to realize the PSO algorithm. Second, its search technique using not the gradient information but the values of the objective function makes it an easy-to-use algorithm. Third, it is computationally inexpensive, since its memory and CPU speed requirements are very low. Fourth, it does not require a strong assumption made in conventional deterministic methods such as linearity, differentiability, convexity, separability or non-existence of constraints in order to solve the problem efficiently. Finally, its solution does hardly depend on initial states of particles, which could be a great advantage in engineering design problems based on optimization approaches.

After it was firstly introduced in 1995 by Kennedy and Eberhart, modified versions of PSO have been introduced to improve the performance of the original PSO. One of them introduces a new parameter called inertia weight [17]. This is a common PSO where inertia weight is linearly decreasing during iteration. Another common version of PSO in the early development of PSO was proposed by Clerc [18] that is called PSO with constriction factor. These two famous versions of PSO can be considered as standard version of PSO. Later, PSO has attracted many researchers to improve its performance as can be seen in [19-22].

In PSO, instead of using genetic operators, individuals called as particles are "evolved" by cooperation and competition among themselves through generations. A particle represents a potential solution to a problem. Each particle adjusts its flying according to its own flying experience and its companion flying experience. Each particle is treated as a point in a D-dimensional space. The ith particle is represented as $X_i=(x_{i1},x_{i2},\ldots,x_{iD})$. The best previous position (giving the minimum fitness value in the case of minimization) of any particle is recorded and represented as $P_i=(p_{i1},p_{i2},\ldots,p_{iD})$, this is called *pbest*. The index of the best particle among all particles in the population is represented by the symbol g, called as *gbest*. The velocity for the particle i is represented as $V_i=(v_{i1},v_{i2},\ldots,v_{iD})$. The particles are updated according to the following equations:

$$v_{id}^{j+1} = w^j.v_{id}^j + c_1.rand_1().(p_{id}^j - x_{id}^j) + c_2.rand_2().(p_{gd}^j - x_{gd}^j) \quad (21)$$

$$x_{id}^{j+1} = x_{id}^j + v_{id}^{j+1} \quad (22)$$

where $c_1$ and $c_2$ are two positive constants which contribute the weights for "cognition" part (second term of (21)) and "social" part (third term of (22)) [23], $rand_i(.)$ are uniformly distributed random numbers between 0 and 1, and $j$ represents iteration. Equation (21) is used to calculate particle's new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group's best experience. Then the particle flies toward a new position according to (22). The performance of each particle is measured according to a pre-defined fitness function (objective function), which is related to the problem to be solved. Inertia weight, w is brought into the equation to balance between the global search and local search capability [13]. A standard PSO where w is linearly decreasing value from 0.9 to 0.4 during iteration is common in many studies using PSO.

Another common version is PSO with constriction method [14] that was proposed by Clerc to improve the convergence of the original PSO (21-22). Afterward, it is revealed this Clerc's PSO is equivalent to the original PSO with inertia weight where the optimal parameters are selected as: $c_1=c_2=1.494$ and w=0.729 [15]. Thus, we will use this version of PSO throughout this work.

Furthermore, an important feature in PSO is velocity clamping. The idea of velocity clamping is to prevent swarm (particle) explosion that was revealed in the early development of PSO. It is therefore necessary to keep the velocity of particle within the range [$-V_{max}$, $V_{max}$] before update the particle position, see Table II. Eberhart and Shi [16] concluded that a better approach to use as a prudent rule of thumb is to limit $V_{max}$ to $u_b$ (upper bound of particle).

TABLE II
Velocity clamping mechanism

| |
|---|
| **If** $\lvert v_{id}^j \rvert > V_{\max}$ |
| $v_{id}^j = V_{\max}.sign(v_{id}^j)$ |
| **Else** |
| $v_{id}^j = v_{id}^j$ |
| **End** |

In addition to velocity clamping, we also apply random mode bound handling mechanism that is necessary to confine the solution within a certain bound. If a particle exceeds the lower or upper limit of the *d-th* dimension, a random value, uniformly distributed between [$l_b$, $u_b$], is assigned to the *d-th* component of the particle's position vector (see Table III).

TABLE III.
Bound handling mechanism

| |
|---|
| **If** $p_{id}^j > u_b$ |
| $p_{id}^j = u_b.rand()$ |
| **Elseif** $p_{id}^j < l_b$ |
| $p_{id}^j = l_b.rand()$ |
| **Else** $p_{id}^j = p_{id}^j$ |
| **End** |

*B. Constrained optimization*

The objective of the optimization is to minimize an error performance ($J$) defined as:

$$J = \int (x - x_r)^2 + (\theta)^2 \, dt \quad (23)$$

Based on our approach, the searching procedure of the robust controller gains using constrained optimization can be formulated as follows (Table IV).

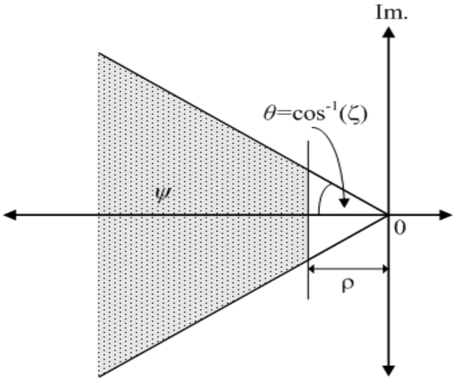


Fig. 4. A wedge region in complex plane for closed loop poles placement

TABLE IV.
Constrained optimization

| | | |
|---|---|---|
| **Minimize:** $f(X) = -J^{-1}$ | | |
| **Subject to constraint:** | $\lambda_n(X) \in \psi$ | **for n=1,2,...** |
| **and boundary constraint**: | $X \in [l_b,\ u_b]$ | |

where $X=K=(k_1,k_2,k_3,k_4,k_i)$ is the vector solutions such that $X \in S \subseteq R^{n+1}$. $S$ is the search space, and $F \subseteq S$ is the feasible region or the region of $S$ for which the constraint is satisfied. The constraint here is the closed loop poles region; in the feasible region, the controller gains are found such that the closed loop poles (λ) lie within a wedge region ($\psi$) of a complex plane as given in Fig. 4. The wedge region can be specified by two parameters θ and ρ which can be related to desired transient response characteristics i.e.: damping ratio (ζ) and settling time.

*C. Constraint handling*

An efficient and adequate constraint-handling technique is a key element in the design of stochastic algorithms to solve complex optimization problems. Although the use of penalty functions is the most common technique for constraint-handling, there are a lot of different approaches for dealing with constraints [17].

Instead of using penalty approach like in [18] where the optimizer seemed to be inefficient (high iterations), we adopt a dynamic-objective constraint-handling method (DOCHM) [19] in order to improve the efficiency. Through defining distance function *F*(*X*), DOHCM converts the original problem into bi-objective optimization problem min(*F*(*X*),*f*(*X*)), where F(*X*) is treated as the first objective function and f(x) is the second (main) one.

The auxiliary distance function *F*(*X*) will be merely used to determine whether or not a particle (candidate of solution) is within the feasible region and how close a particle is to the feasible region. If a particle lies outside the feasible region (at least an eigenvalue lies outside the wedge region), the particle will take *F*(*X*) as its optimization objective. Otherwise, the particle will instead optimize the real objective function *f*(*X*). During the optimization process if a particle leaves the feasible region, it will once again optimize *F*(*X*). Therefore, the optimizer has the ability to dynamically drive the particles into the feasible region.

The procedure of the DOCHM applied to the eigenvalue assignment in the wedge region is illustrated in the following pseudo-code (Table V). Referring to Fig. 5, let dn is an outer distance of an eigenvalue ($\lambda_n$) to the wedge region. It is noted that if an eigenvalue lies within the wedge region, $d_n$=0. *F*(*X*) is defined by:

$$F(X) = \sum_{i=1}^{n+1} \max(0, d_n(\lambda_n(X))) \tag{24}$$

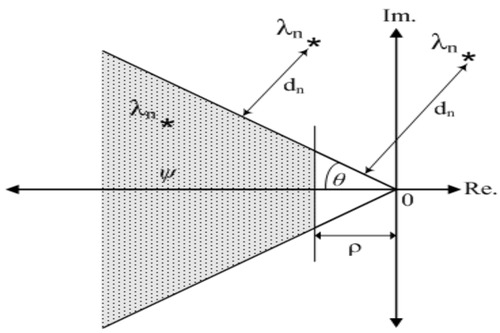


Fig.5. An eigenvalue distance to the wedge region in complex plane

*D. Stopping criterion*

When using optimization algorithms the goal is usually clear that is to find the global optimum solution. However, in general it is not clear when this goal is achieved, especially if real-world problems are optimized for which no knowledge about the global optimum is available. Therefore, it is not easy to decide when the execution of an optimization algorithm should be terminated [20].

TABLE V.
Pseudo-code for constraint handling

**If** $F(X) = 0$

$f(X) = -J^{-1}$

**Else**

$f(X) = F(X)$

**End**

In literatures, mostly two stopping criteria are applied in single-objective optimization: either an error measure if the optimum value is known is used or the number of function evaluations (number of iterations). There are some drawbacks for both. The optimum has to be known in the first method, so it is generally not applicable to real-world problems because the optimum is usually not known a priori. The second method is highly dependent on the objective function. Because generally no correlation can be seen between an optimization problem and the required number of function evaluations (iterations), it has to be determined by trial-and-error methods usually. Improper selection of the number of iterations to terminate the optimization can lead to either premature convergence or excessive computational effort.

As a result, it would be better to use stopping criterion that consider knowledge from the state of the optimization run. The time of termination would be determined adaptively, so the optimization run would be efficient. Several stopping criterions are reviewed in [20]. Although the authors did not conclude which one is the best for all problems, it is believed that performance improvement can be obtained with adaptive stopping criterion. In this work, we adopt the stopping criterion which is distribution-based criterion which considers the diversity in the population. If the diversity is low, the individuals are close to each other, so it is assumed that convergence has been obtained [20]. Standard deviation (σ) of particles' best positions in each dimension is checked. If it is below a threshold $\varepsilon$ (small number) for sufficiently large number of iterations $\eta$, the optimization will be terminated. It can be formulated as in Table VI.

TABLE VI
Stopping criterion

**If**

$$\sigma_d = \sqrt{\frac{1}{\eta}\sum_{j=1}^{\eta}(p_{gd}^{j} - \bar{p}_{gd})^2} < \varepsilon(\max(p_{gd}) - \min(p_{gd}))$$

**(for d=1,2,…,D)**

**stop iterations.**

**End**

## V. SIMULATION RESULTS

The PSO-based optimization and simulation work is facilitated by Matlab 2006. Before the optimization is executed, a number of parameters must be specified. The main PSO parameters are chosen based on explanation that has been discussed in previous section. These parameters are listed in Table VII.

TABLE VII.
PSO-based optimization parameters

| | | |
|---|---|---|
| Dimension of the problem | $D$ | 5 |
| Swarm (population) size | $N$ | 50 |
| Cognitive acceleration constant | $c_1$ | 1.494 |
| Social acceleration constant | $c_2$ | 1.494 |
| Inertia weight | $w$ | 0.729 |
| Upper bound of initial swarm matrix | $u_b$ | 75 |
| Lower bound of initial swarm matrix | $l_b$ | -75 |
| Maximum velocity | $V_{max}$ | 75 (=ub) |
| Maximum iteration | $j_{max}$ | 2000 |
| Number of iteration for which stopping criterion applies | $\eta$ | 500 |
| Standard deviation threshold for which stopping criterion applies | $\varepsilon$ | 1% |

The next is to choose the parameters of the wedge region (Fig. 4) whose role is to locate the closed loop poles of the system. The damping ratio is usually set to $\zeta$=0.7 to produce sufficient overshoot damping in the response. The transient margin ($\rho$) is specified according to the desired speed of the response. This is problem-dependent parameter. Here, we set $\rho$=1.5.

Since PSO is stochastic optimization, a number of optimization runs need to be executed with different initial random seeds. To evaluate the quality of the solution (robustness, convergence, repeatability) obtained by PSO, 15 runs have been executed here. The mean value, the standard deviation of the fitness value ($f(X)=J$) and other results are recorded in Table VIII.

TABLE VIII.
Optimization results

| | |
|---|---|
| Average $f(X)$ | -0.00029 |
| Median $f(X)$ | -0.00028 |
| Standard deviation $f(X)$ | 0.000027 |
| Range of $f(X)$ | -0.00034 to -0.00025 |
| Average number of iteration | 1285 |
| Average computation time | 11.5 seconds |

It can be seen that the optimization results in a robust solution where a small standard deviation is obtained, the range of the fitness value is also relatively small. Fig. 6 shows the distribution of the eigenvalues (closed loop poles) for 15 runs. All eigenvalues lie within the desired wedge region. It is also noted that the dominant eigenvalues lie nearby the straight horizontal margin line ($\rho$=1.5) and they are close to each others.

Furthermore, for the purpose of evaluation and comparison with LQR-based controllers, an optimum solution is picked from the median data of those 15 runs. The corresponding optimum feedback controller gains obtained by PSO are shown in Table IX.

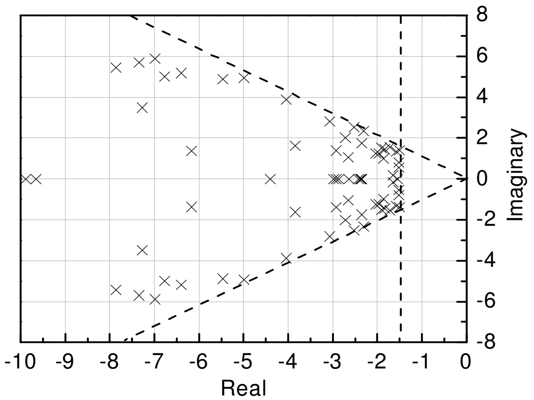


Fig 6. Distribution of eigenvalues within the wedge region

*A. Swing-up and Tracking Response*

In this simulation, the pendulum is swung-up from the rest (downward pendulum) by giving sinusoidal input of the cart position. The travel distance of the cart is bounded by the amplitude of the reference input to ±0.3m. Then after the pendulum is stabilized by the stabilizing controller, the tracking mode is started at 10$^{th}$ second by giving a square wave reference input of cart position (±0.1m). The response is shown in Figs. 7-9. It can be seen from Fig. 8, the swing-up controller is able to swing the pendulum to upward posture within less than 4s.

TABLE IX
State feedback controller gains by PSO

| $k_1$ | $k_2$ | $k_3$ | $k_4$ | $k_i$ |
|---|---|---|---|---|
| -35.01 | 65.64 | -27.07 | 13.43 | 25.70 |

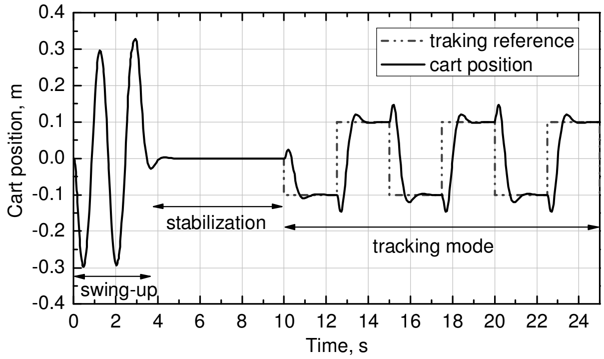


Fig. 7. Cart position response

Furthermore, as seen in Fig. 9, the proposed state feedback control is able to maintain the pendulum upright with amplitude of less than 8$^o$ while keeping the cart moving back and forth to track the square wave input trajectory (Fig. 7).

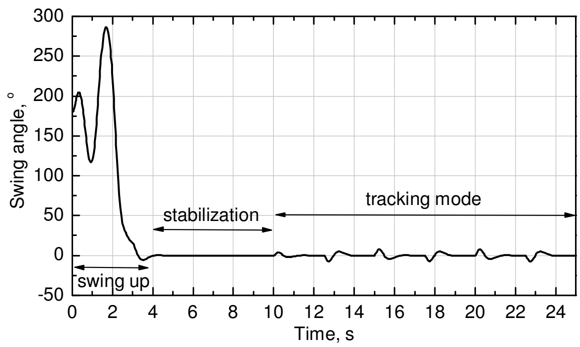


Fig. 8. Pendulum swing angle response

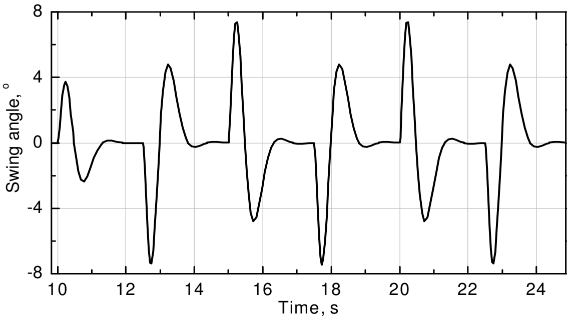


Fig. 9. Zoomed view of swing angle during tracking mode

### B. Disturbance Rejection

In this section, disturbance rejection ability of the controller strategy is shown. After the pendulum swing-up and stabilization, disturbances at 10th and 18th second are introduced to the pendulum. The disturbance signal and the response to reject disturbance are shown in Figs. 10-11 respectively. It can be seen that the given disturbances are well compensated so that the pendulum is maintained upright even though the angle is beyond the switching margin ($|\theta|>25^o$).

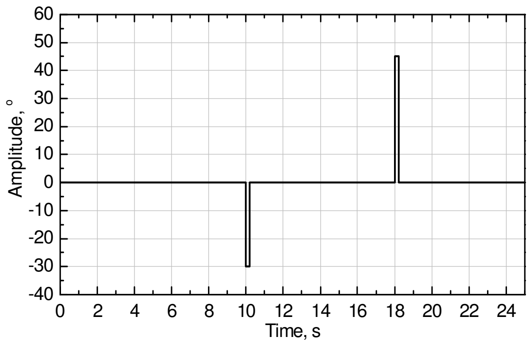


Fig. 10. Disturbance signal to the pendulum swing angle

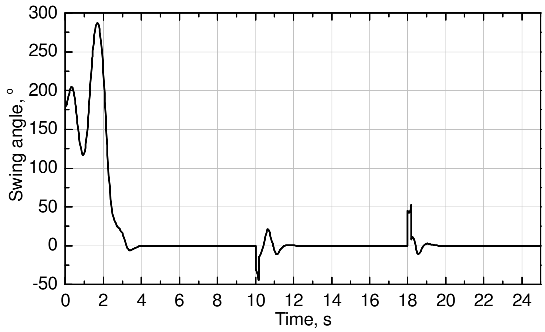


Fig. 11. Pendulum swing angle response to disturbance signal in Fig. 8

## VI. CONCLUSIONS

In this paper, an intelligent tuning method of state feedback control for stabilizing controller of a self-erecting inverted pendulum based on PSO has been presented. This approach has been motivated especially by the fact that the designers often have to face inconvenient free parameter adjustment in the controller design using conventional methods like LQR. Instead, by constrained optimization approach, it is transformed into parameter setting of the closed loop poles region (wedge region) that is more sensible in term of the desired system performance in time domain.

In addition, the ability of the Clerc's PSO combined with the efficient constraint handling is also demonstrated. The PSO-based optimizer did not take much time to obtain the convergence solutions. For further research, robust criterion and model uncertainty may be included and verification through experimental work is required as well.

## REFERENCES


[1] Yoshida, K (1999) Swing-up control of an inverted pendulum by energy based methods. Proceedings of the American Control Conference, pp.4045-4047.

[2] M. I. H. Nour, J. Ooi, , and K. Y. Chan (2007). Fuzzy logic control vs. conventional PID control of an inverted pendulum robot. Proc. of IEEE International Conference on Intelligent and Advanced Systems, pp. 209-214.

[3] Ibanez, C.A., Frias, O.G., and Castanon, M.S. (2005). Lyapunov-based controller for the inverted pendulum cart system. Nonlinear Dynamics 40: 367–374, Springer.

[4] Yu, H., Liu, Y., and Yang, T. (2007). Tracking control of a pendulum-driven cart-pole underactuated system. Proc. IEEE International Conference on Systems, Man and Cybernetics, pp.2425-2430.

[5] N. M. Singh, J. Dubey, and G. Laddha. (2008). Control of pendulum on a cart with state dependent Riccati equations. Proceedings of World Academy of Science, Engineering and Technology, Volume 31, ISSN 2070-3740.

[6] Bugeja, M. (2003). Non-linear swing-up and stabilizing control of an inverted pendulum system. Proc. of EUROCON'03, Vol 2, pp.437-441.

[7] A.A. Saifizul, Z. Zainon, N.A Abu Osman, C.A. Azlan and U.F.S.U. Ibrahim. (2006). Intelligent control for self-erecting inverted pendulum via adaptive neuro-fuzzy inference system. American Journal of Applied Sciences 3 (4): 1795-1802.

[8] K. J. Astrom and K. Furuta. (2000). Swinging up a pendulum by energy control, Automatica, vol. 36.

[9] Ogata, K. (2002). Modern control engineering. New Jersey: Prentice Hall.

[10] Kim, T.H., Maruta, I., and Sugie, T. (2008). Robust PID controller tuning based on the constrained particle swarm optimization. Automatica vol. 44, pp.1104-1110.

[11] Parsopoulos, K. E., & Vrahatis, M. N. (2002). Recent approaches to global optimization problems through particle swarm optimization. Natural Computing, 1(2–3), 235–306.
[12] Kennedy, J. and Eberhart, R.C. (1995). Particle swarm optimization. Proc. IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, pp. IV: 1942- 1948.
[13] Shi, Y.H. and Eberhart, R.C. (1998). A modified particle swarm optimizer. Proc. of IEEE International Conference on Evolutionary Computation.
[14] Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multi-dimensional complex space, IEEE Transaction on Evolutionary Computation, vol. 6, pp. 58–73.
[15] Poli, R., Kennedy, J. and Blackwell, T. (2007). Particle swarm optimization: an overview. Swarm Intelligent, 1: 33–57.
[16] Eberhart, R.C. and Shi, Y.H. (2000).Comparing inertia weights and constriction factors in particle swarm optimization. Proc. The 2000 Congress on Evolutionary Computation. Vol.1, pp. 84-88.
[17] Coello Coello, C.A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. Computer Methods in Applied Mechanics and Engineering, vol. 191, no. 11-12, pp. 1245–1287.
[18] Solihin, M.I, Wahyudi, Legowo, A and Akmeliawati, R. (2009). Self-erecting inverted pendulum employing PSO for stabilizing and tracking controller. Procs. of 5th International Colloquium on Signal Processing & Its Applications, pp.63-68.
[19] Lu, H. and Chen, W. (2006). Dynamic-objective particle swarm optimization for constrained optimization problems. Journal of Global Optimization, 12:409-419.
[20] Zielinski, K. and Laur, R. (2007). Stopping criteria for a constrained single-objective particle swarm optimization algorithm. Informatica 31: 51–59.

Mahmud Iwan Solihin received B.Eng. from University of Gadjah Mada, Yogyakarta, Indonesia in 2002 and M.Sc in mechatronics engineering from International Islamic University Malaysia (IIUM) in 2007. Currently he is a lecturer in Mechatronics Department, UCSI University and also attached to Mechatronics Enginering, IIUM as a postgraduate student. His research interest includes control system, intelligent systems and optimization. He has been a member of IEEE since 2008.

Rini Akmeliawati received the Bachelor of Engineering (Honours) degree in electrical engineering from the Royal Melbourne Institute of Technology (RMIT) University, Melbourne, Australia, in 1997, and the Ph.D. degree in electrical and electronic engineering from the University of Melbourne, Australia, in 2002. She is currently an Associate Professor of the Department of Mechatronics Engineering at the International Islamic University Malaysia. Previously, she was a Lecturer at RMIT University (2001–2004) and at Monash University (2004–2008). She has been the chair of IEEE Instrumentation and Measurement Society,Malaysia Chapter, since 2007. Her main research interests include nonlinear control systems theory and applications, intelligent systems and image processing.