

Improve Linear Quadratic Regulator by Particle Swarm Optimization Algorithms for Two Wheeled Self Balancing Mobile robot

Dr.Ekhlash.H.Karam

Computer Engineering
University of Mustansiriyah
Iraq/Baghdad
ek_karam@yahoo.com

Noor.M.Mjeed

Computer Engineering
University of Mustansiriyah
Iraq/ Baghdad
noormjeed@yahoo.com

Abstract The aim of this paper is to suggest a methodical smooth control method for improving the stability of two wheeled self-balancing robot under effect disturbance. To promote the stability of the robot, the design of linear quadratic regulator using particle swarm optimization (PSO) method and adaptive particle swarm optimization (APSO). The computation of optimal multivariable feedback control is traditionally by LQR approach by Riccati equation. Regrettably, the method as yet has a trial and error approach when selecting parameters, particularly tuning the Q and R elements of the weight matrices. Therefore, an intelligent numerical method to solve this problem is suggested by depending PSO and APSO algorithm. To appraise the effectiveness of the suggested method, The Simulation result displays that the numerical method makes the system stable and minimizes processing time.

Keywords —ASO, LQR, PSO, two wheeled self-balancing mobile robot.

I. INTRODUCTION

A two wheeled self-balancing robot is a special case of robots ,its balancing problem is a hot research topic, and the main reason for this is unstable, nonlinear , multivariable and strong coupled so that it has become traditional device used to test the processing abilities of different control methods. The ingrained intricacy related with the control of this platform makes it the most convenient test platform for the development and design of control systems of automobiles, crafts, space, robots, the rockets, and even martial transport. Can also be used to improve the means of personal transport, and would show to be very helpful as it would authorize people to travel small distances on a small style of transport and would help decrease air pollution [1]. Engineers and scientists are hard working to improve and develop methods of control unit to make the system stable and more powerful; different mechanisms are Designed and implemented on two wheeled self-balancing robot. design

conventional controller (PID) It is much used in many fields in industry, with tuning of just three gains [2], Pole placement produces minimum overshoots but high settling time [3], LQR produces more overshoots than pole placement but good settling time [3], LQR with neural [4], PD control with fuzzy [4], LQR and PID are combined to improve a stabile [5], Design optimal LQR controller based on Particle swarm optimization algorithm [6, 7].

The simplest method to design state feedback control, mostly for multi outputs system. Can get it by linear quadratic (LQR) structured method by solving algebraic recombinate equations. However, this method as yet has a trial and error approach when selecting parameters. Particularly adjust the Q and R elements of the weight matrices

The goal of the paper is to solve the problem of trial and error approach of selection parameter in the traditional methods, by using intelligent algorithm (PSO, APSO) to tuning feedback gains. In this paper we offer as follows: part II

characterize the Mathematical Model of The Two Wheeled Self Balancing robot, part III characterize Linear Quadratic Regulator .PSO and APSO algorithms are lead in part IV. Result of Simulation is discussed in part V the conclusion in part.

II. MATHEMATICAL MODEL OF TWO WHEELED SELF BALANCING

Two wheeled self-balancing robot is mainly made up of two wheels and body, and the robot is the coaxial two wheels, driven by an independent engine. Parameters of quality are considered, radius and moment of inertia of the two wheels are the same, so the center of object gravity is inverted over the axletree and it maintains equilibrium over sports. This section describes the dynamic model of the system by introducing differential equations that describe the dynamics of a two-wheeled vehicle. The model derivative on the basis of Newton's equations of motion. The mechanical model of two wheeled self-balancing mobile robot consists of two separate sub-models, namely: a linear model of direct current motor activating the wheels and nonlinear model of the inverted pendulum. The dynamics of vehicle the medium body and two wheels are analyzed separately and this will lead to the following two nonlinear equations of motion [8].

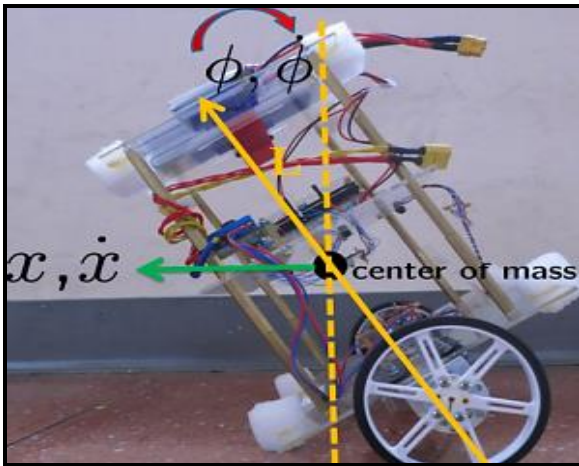


Fig. 1 Diagram of the chassis two wheeled self-balancing robot [9].

After calculation of all the equations of the structure, this model is combined with the direct current (dc) motor model to get the final nonlinear equations of the robot are given below. Nonlinear dynamic equations are rearranged as follows in here [8].

$$(I_p + M_p l^2) \ddot{\theta}_p = M_p l \ddot{x} \cos \theta_p + \frac{2K_m K_e}{Rr} \dot{x} + M_p g l \sin \theta_p = -\frac{2K_m}{R} V_a \quad (1)$$

$$(2M_w + \frac{2I_w}{r^2} + M_p) \ddot{x} + M_p l \ddot{\theta}_p \cos \theta_p + \frac{2K_m K_e}{Rr^2} \dot{x} - M_p l \dot{\theta}_p^2 \sin \theta_p = \frac{2K_m}{Rr} V_a \quad (2)$$

Nonlinear equations must be Linearization to get state equations. Suppose that $\theta_p = \pi + \phi$, hence linearized equations are obtained.

$$\ddot{\phi} = \frac{M_p l}{(I_p + M_p l^2)} \ddot{x} + \frac{2K_m K_e}{Rr(I_p + M_p l^2)} \dot{x} - \frac{2K_m}{R(I_p + M_p l^2)} V_a + \frac{M_p g l}{(I_p + M_p l^2)} \quad (3)$$

$$\ddot{x} = \frac{M_p l}{(2M_w + \frac{2I_w}{r^2} + M_p)} \ddot{\phi} - \frac{2K_m K_e}{Rr^2(2M_w + \frac{2I_w}{r^2} + M_p)} \dot{x} + \frac{2K_m}{R(2M_w + \frac{2I_w}{r^2} + M_p)} V_a \quad (4)$$

Using the above two linear equations are obtained the state space model, mathematical modeling is required to design a controller [9].

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{2K_m K_e (M_p l r - I_p - M_p l^2)}{Rr^2 \alpha} & \frac{M_p g l^2}{\alpha} & 0 & 0 \\ 0 & 0 & \frac{2K_m K_e (r\beta - M_p l)}{Rr^2 \alpha} & \frac{M_p g l \beta}{\alpha} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2K_m (-M_p l r + I_p - M_p l^2)}{Rr \alpha} \\ 0 \\ \frac{2K_m (-r\beta + M_p l)}{Rr \alpha} \end{bmatrix} u \quad (5)$$

$$\beta = 2M_w + \frac{2M_w}{r^2} + M_p, \alpha = I_p\beta + M_p l^2 (M_w + \frac{I_w}{r^2}) \quad (6)$$

TABLE I
THE PARAMETER VALUES OF THE TWO
WHEELED SELF BALANCING SYSTEM [8].

Symbol	Definition	Parameter
l	Length to the body's center of mass	0.07 m
r	Radius of wheel	0.051 m
M_p	Mass of body	1.13 kg
M_w	Mass of wheel	0.03 kg
g	Gravity	9.81 m/s^2
K_m	Motor torque constant	0.006123 Nm/Amp
K_e	Back EMF constant	0.00608 Vs/rad
I_p	Inertia of the body	0.0041 kg.m^2
I_w	Inertia of the wheel	0.000039 kg.m^2
R	Nominal Terminal Resistance	3Ω

After applied of values in the Table 1, it is obtained.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0097 & 11.1594 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.0293 & 172.1160 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.0815 \\ 0 \\ 0.2456 \end{bmatrix} u \quad (7)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (8)$$

III. LINER QUADRATIC OPTIMAL CONTROL

Optimal control LQR is the design most mature controller a method in the improvement of recent control theory [10]

Here $x(t), u(t)$ are represent input vector, $y(t)$ represents the output and $u(t)$. The LQR approach constructs a linear state feedback law expressed by

$$u(t) = -ku(t) \quad (9)$$

Where K represent gains matrix. K design is the exchange between transient response and control efforts. The optimal control method in this design is to look for the control vector in Equation (10), which reduces the performance index in the below:

$$J = \int_0^{\infty} x^T(t)Qx(t) + u^T(t)Ru(t)dt \quad (10)$$

Where R and Q are a positive definite and semi-definite weight matrix of state matrix $u(t)$ and input matrix $x(t)$. The control gain matrix k is obtained by solving

$$K = R^{-1}B^T P \quad (11)$$

where p is the solution of the Algebraic Riccati equation by:

$$A^T P + PA - PB R^{-1}B^T P + Q = 0 \quad (12)$$

It is required that (A, C) is detectable and (A, C) is stabilizable. Nevertheless, the method as yet has a trial and error approach when selecting parameters, particularly tuning the Q and R elements of the weight matrices. When the system has a greater degree, the way trial and error would be impossible. And also for simpler systems, it is work-intensive and it takes time. For that we use optimization method adaptive particle swarm (APSO) and (PSO) to adjust Q and R matrix matrices.

IV. PARTICLE SWARM OPTIMIZATION

PSO is one of a strong optimization method with high efficiency compared with other methods. It is a random evolutionary arithmetic technique based on the movement and intelligence of swarms. The PSO technique is designed with a number of randomized solutions and the search for optima by updating generations. In a D-dimensional search space A

swarm of N particles is formed that moves around. Each particle is set position and speed, which is updated, based on social cognitive factors and personal realize the best solution. The velocity of each particle its towers the global best and its own personal best Applying the following equation [10, 11]

$$V_i^{k+1} = w * V_i^k + c_1 * r_2 * rand(X^b - X_i^k) + c_2 * r_2 * rand * (X_i^g - X_i^k) \quad (13)$$

where, X_i is the present particle position , V_i is the particle velocity, w , c_1, c_2 are constants and r_1, r_2 are random variables within the range between 0 or 1, and all positions are updated based on their movement in serch space according to equation of update:

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (14)$$

Performance is calculated for each particle is weighted by fitness function (performance index), which is attached to the problem to be resolved.to reduce the integral of squared error (ISE) and is The aim of the optimization.

$$F = ISE = \int e^2(t)dt \quad (15)$$

V.ADAPTIVE PARTICALE SWARM OPTIMIZATION

APSO one of All the suggsted modifications is basically to improve exploration and exploitation or even both exploration and exploitation capability of the PSO, we extend the idea of adaptive inertia weight strategy to solve the optimal problem of LOR. To implement the weight-inertia adjustment strategy, it is important to evaluate the position of the swarm during each repeat step. Thus, the success rate (SP) of particles is utilized to update the adaptively velocity .a small value of SP means that the particles fluctuate around the optimum value with very little improvement. However, Large value

of SP Refers that the particles have arrived the best value and the particles progress slowly towards optimum, while. Thus, the success rate can be used to adjust the inertia weight adaptively. As follows below the SP value is calculated and used to arrive at the adaptive Inertia weight parameter (AIWP) as follow in(16) [13].

$$W(t) = W_{\max} - W_{\min})SP + W_{\min} \quad (16)$$

VI. RESULT OF SIMULATION

The PSO and APSO methods are utilized to obtain the optimal values of the weighting matrices of LQR algorithm is executed in MATLAB 2016b. Table2 Displays the variables used in Particle Swarm Optimization and adaptive Particle Swarm Optimization algorithms. Furthermore, the number of repetition(i), particle size and cognitive acceleration constants in both PSO and APSO are same except the inertia weight(AIWP).It was also used parameter Called it dimension(d) in algorithms it was chosen 5, $(q_{11}, q_{22}, q_{33}, q_{44}, R)$ representing the diagonal elements to be improved in the LQR design . According to the fitness function ISE, the optimization algorithms are implemented the best is obtained from global particles, the weights of LQR for a specific number of repetitions .Results obtained in table2 after applied intelligent algorithms and get optimal values Q,R, The two methods were compared in table4 and has been shown APSO better than PSO, through(ts,mp,tr).

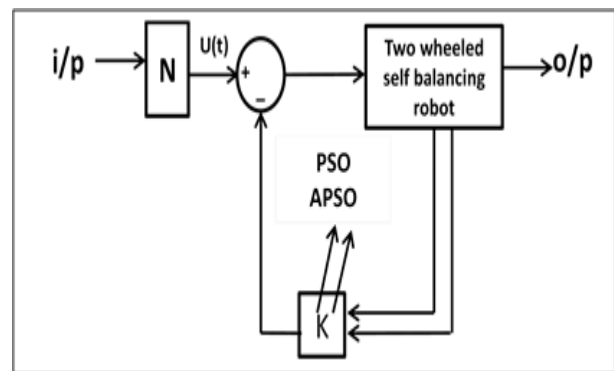


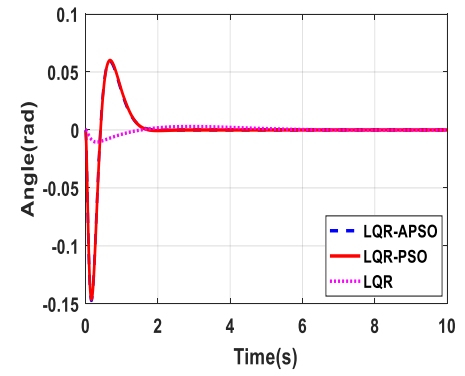
Fig. 2 Represent LQR control system.

TABLE 2
PARAMETERS USED IN THE ALGORITHM.

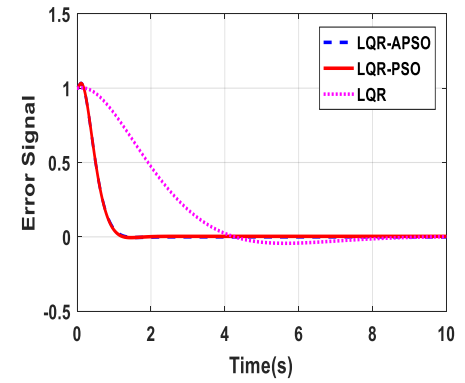
initialization Parameters	PSO	APSO
No.of iterations(i)	100	100
birds step	30	30
Dimensions(d)	5	5
C_1	0.5	0.5
C_2	2	2
weight(w)	0.3	AIWP

TABLE 3
AFTER APPLIED THIS INTELLIGENT ALGORITHM
(PSO, APSO) RESULTS ARE OBTAINED

Optimization algorithm	Weight matrices	Controller gain
ASO	Q=diag 10^4 *[2.0524 0.1000 0.001 0.001] R=0.22	K= 10^3 *[-0.3054 -0.1661 2.4325 0.1948]
PSO	Q=diag 10^4 *[2.3663 0.100 0.0010 0.001] R=0.27	K= 10^3 *[-0.2960 -0.1597 2.3896 0.1912]
ISE	0.038	0.022

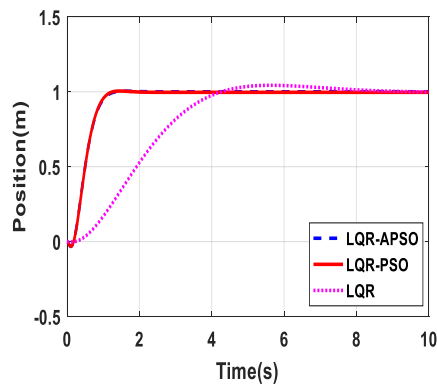


(b)

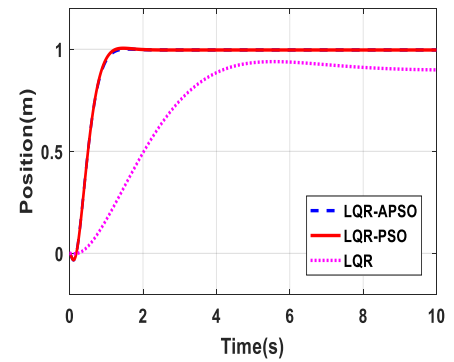


(c)

Fig. 3 Simulation result of the Robotic system with state feedback controller. (A) Position (B) Angle (C) Error signal.



(a)



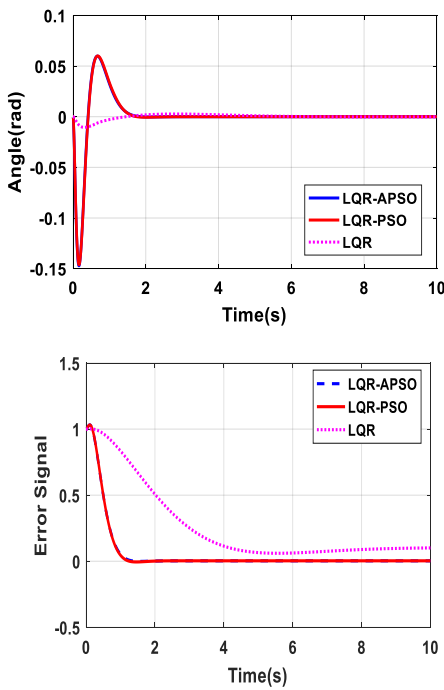


Fig. 4 Simulation result of the Robotic system with state feedback controller with step disturbance.

TABLE 4 COMPARATIVE BETWEEN OF METHODS.

	Settling Time(ts)			Rise Time(tr)			PeakOvershoot(mp)		
	APSO,PSO,LQR			APSO,PSO,LQR			APSO,PSO,LQR		
Position	1.2	1.3	8.4	0.62	0.63	2.85	0	0	10
Angle	1.7	1.7	5.12	0.17	0.17	0.87	0.06	0.063	0.03

I. CONCLUION

This paper proposed an optimal control method based on optimization algorithm (PSO, APSO) and Linear Quadratic Regulator(LQR) to stabilize the two-wheeled mobile robot. The parameters R and Q are adjusted by (LQR) weight matrices using an intelligent algorithm adaptive particle swarm algorithm. This numerical way removes the effort and takes a

long time "trial and error" way commonly implemented to tune the weighting matrices. Simulation was performed to compare the performance of (PSO, APSO) tuned LQR with traditional LQR. using (PSO, APSO) control were tested for position and angle, it has the ability to reject disturbance. But LQR without (PSO, APSO) making the system susceptible to disturbance. The results display that this approach improve the performance of the system by small or no overshoot as well as faster settling and reduces tuning time.

RFERENCES

- [1]Liu Kun, Bai Ming, Ni Yuhua, "Two-wheel self-balanced Car based on Kaman Filtering and PID Algorithm" 2011 IEEE 18Th International Conference on Industrial Engineering and Engineering Management (IE&EM), vol.Part 1, Sept. 2011.
- [2] D. Jin, "Development of a Stable Control System for a Segway," Royal Institute of Technology, 2013.
- [3] J. Wu and W. Zhang, "Research on Control Method of Two-Wheeled Self-Balancing Robot," Proceedings of the 4th International Conference on Intelligent Computation Technology and Automation (ICICTA '11), pp. 476–479, Shenzhen, China, March 2011.
- [4] C. Sun, T. Lu, and K. Yuan, "Balance Control Of Two-Wheeled Self-Balancing Robot Based on Linear Quadratic Regulator And Neural Network," in International

- Conference on Intelligent Control and Information Processing, pp. 862–867, 2013.
- [5] Liang Sun, Jiafei Gan “Researching of Two-Wheeled Self-Balancing Robot Base on LQR Combined with PID, Intelligent Systems and Applications” (ISA), 2010 2nd International Workshop on , vol., no., pp.1-5. , May. 2010.
- [6] Jian, F., & Liu, J. Y, “PSO in Two-Wheeled Self-Balancing Robot Control Research ” Advanced Materials Research Zug-, pp.534-537, January 2014.
- [7] Fang, Jian. , “The LQR Controller Design of Two-Wheeled Self-Balancing Robot Based on the Particle Swarm Optimization Algorithm”, *Mathematical Problems in Engineering*, pp.1-6, 2014.
- [8] Jamil, O., Jamil, M., Ayaz, Y., Ahmad, K., " Modeling, Control of a Two-Wheeled Self-Balancing Robot”, International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE), pp.191-199. April 2014.
- [9] Juang, H.-S., Lurr, K.-Y, “Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller Board” 2013 10th IEEE International Conference on Control and Automation, (ICCA), pp. 634-639. August 2013.
- [10] Vinodh K. E., Jovitha J., " Robust LQR Controller Design for Stabilizing and Trajectory Tracking of inverted Pendulum", *Procedia Engineering*, Vol. 64, pp. 169-178, 2013.
- [11] Y. del valle et al., “Particle Swarm optimization: Basic Concepts Variants and Applications in Power Systems,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.
- [12] S. Panda and N. P. Padhy, “Comparison of Particle Swarm Optimization and Genetic Algorithm for FACTS-Based Controller Design,” *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1418–1427, Sep. 2008.
- [13] A. Nickabadi, M.M. Ebadzadeh, R. Safabakhsh, A novel particle swarm optimization algorithm with adaptive inertia weight, *Appl. Soft Comput.* 11 (4) (2011) 3658–3670