PID Controller Design for Mobile Robot Using Bat Algorithm with Mutation (BAM)

Dwi Pebrianti
[1]*Magister of Computer Science, Universitas Budi Luhur*
Jakarta, Indonesia
[2]*Faculty of Electrical & Electronics Engineering*
Universiti Malaysia Pahang
Pahang, Malaysia
dwi.pebrianti@budiluhur.ac.id

Luhur Bayuaji
[1]*Magister of Computer Science, Universitas Budi Luhur*
Jakarta, Indonesia
[2]*Faculty of Computer Science & Software Engineering*
Universiti Malaysia Pahang
Pahang, Malaysia
luhur.bayuaji@budiluhur.ac.id

Yogesvaran Arumgam
Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang
Pahang, Malaysia

Indra Riyanto
Dept. of Electronics Engineering, Faculty of Engineering, Universitas Budi Luhur
Jakarta, Indonesia

Muhammad Syafrullah
Magister of Computer Science, Universitas Budi Luhur
Jakarta, Indonesia

Nurnajmin Qasrina Ann Ayop
Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang
Pahang, Malaysia

Abstract— **By definition, a mobile robot is a type of robot that has capability to move in a certain kind of environment and generally used to accomplish certain tasks with some degrees of freedom (DoF). Applications of mobile robots cover both industrial and domestic area. It may help to reduce risk to human being and to the environment. Mobile robot is expected to operate safely where it must stay away from hazards such as obstacles. Therefore, a controller needs to be designed to make the system robust and adaptive. In this study, PID controller is chosen to control a mobile robot. PID is considered as simple yet powerful controller for many kind of applications. In designing PID, user needs to set appropriate controller gain to achieve a desired performance of the control system, in terms of time response and its steady state error. Here, an optimization algorithm called Bat Algorithm with Mutation (BAM) is proposed to optimize the value of PID controller gain for mobile robot. This algorithm is compared with a well-known optimization algorithm, Particle Swarm Optimization (PSO). The result shows that BAM has better performance compared to PSO in term of overshoot percentage and steady state error. BAM gives 2.29% of overshoot and 2.94% of steady state error. Meanwhile, PSO gives 3.07% of overshoot and 3.72% of steady state error.**

Keywords—mobile robot, optimization algorithm, PID controller, Bat Algorithm with Mutation, Particle Swarm Optimization

I. INTRODUCTION

Research in mobile robot is expanding day by day. By definition, mobile robot is a type of robot that has capability to move in a certain kind of environment and generally used to accomplish certain tasks with some Degree of Freedoms (DoF). Mobile robot is mainly used in exploration, industry, military, security and entertainments. Nowadays, some specific mobile robots have been developed for example fire fighter robot, security robots and office robots.

In order to accomplish the task, mobile robot needs to have fundamental competences such as able to operate safely, able to avoid obstacles; able to work in hazardous area and at the same time must pose no risk to humans surround. In that case, there is a need of controller in order to control every movement of mobile robot and also to accomplish the work efficiently as required.

Accurate design and control of a mobile robot is not a simple task. It is well known that the operation of a mobile robot is essentially time-variant. This means that the operation parameters of the mobile robot, environment and the road conditions are always varying according to time. In order to achieve a robust and adaptive system and also improved dynamics and steady state performances, controller needs to be designed carefully.

Commonly used controller is Proportional-Integral-Derivative (PID) controller. PID controller is still a famous controller used nowadays due to its remarkable effectiveness, simplicity of implementation and broad applicability. Tuning process is the important part in the designing of PID controller.

Tuning of PID controller parameters is still an active research area for many years. The closed-loop system performance specifications, such as peak overshoot, settling time, rising time, and the robust performance of the control loop over different conditions are the main aim of PID tuning process. Recently, optimization method becomes an option to tune PID controller gain. Optimization methods, such as Genetic Algorithm (GA) [2], Particle Swarm Optimization (PSO) [3-4], Bat Algorithm [5-6], Simulated Kalman Filter (SKF) [7], Ant Colony Optimization (ACO) [8], Evolutionary Algorithm [9] and Teaching Learning Optimization (TLBO) [10] have been proven to give excellence result by improving the steady state characteristics and performance indices, compared to conventional Ziegler-Nichols (Z-N) method [1].

The search of optimal PID controller by using GA has received great attention from researchers. However, it is known that the disadvantage of GA is its enormous computational effort. Therefore, the PID tuning process will consume a lot time that will make the mobile robot design takes a longer time [2]. Additionally, the nature of GA that is easily to be trapped in local minima will give inappropriate PID controller gain. When this inappropriate PID controller gain used in the system, the robust performance in terms of fast time response and also the low steady state error will not be achieved.

The rest of the paper will be written as follow. Section II will discuss on the modeling of mobile robot and PID controller design used in this study. Section III will be the explanation on Bat Algorithm with Mutation (BAM) and its implementation on optimizing the PID controller for mobile robot. Section IV will be the result and discussion. Section V, which is the final section, will be the conclusion of the study.

II. MOBILE ROBOT MODELING AND PID CONTROLLER DESIGN

A. *Modelling of Mobile Robot*

A mobile robot used in this study is a systems that uses electric machines (motors) as motion generators. In this system, voltage will be the input for the electric motor and the output will be the rotational speed of electric motor or the motion of the mobile robot. The Permanent Magnet DC (PMDC) motor acts as an electric actuator. A simple case single Degree of Freedom (DoF) of a mobile robot platform is used in this study and the equations of motion for the robot are generated for moving forward and reverse [11]. Here, a simplified model of the robot platform and its symmetric half are constructed. Fig.1 shows the layout of the mobile platform.

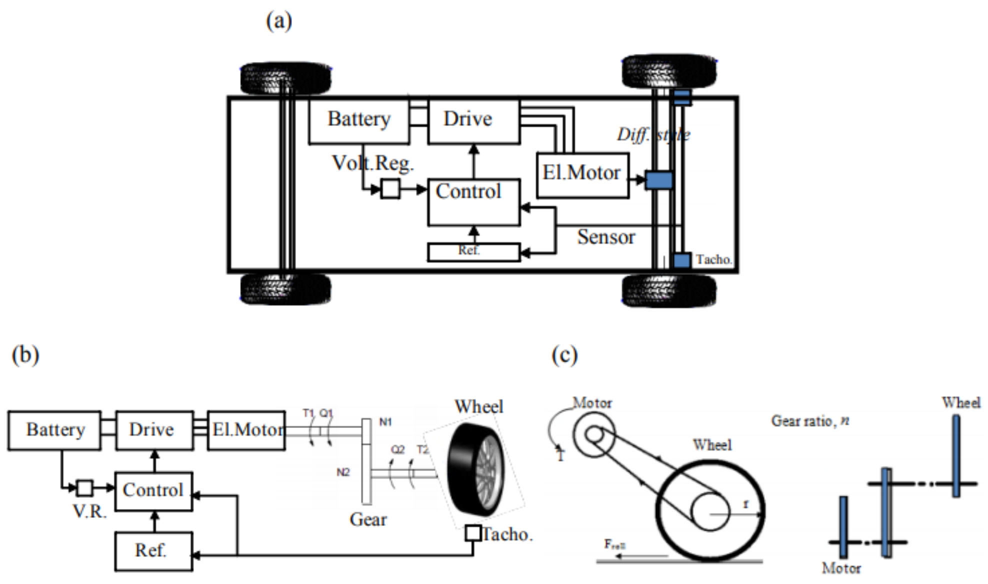

Figure 1. (a) Mobile platform and circuit, top view, (b) a simple model of half of the mobile platform, (c)gear ratio, pulley[11]

DC motor transfer function is generated as the transfer function of the mobile robot platform. In relation to the input voltage, $V_{in}(s)$ to the shaft angular velocity, $\omega(s)$, the Permanent Magnet DC (PMDC) motor transfer function without any load attached is given in Eq. (1).

$$G_{speed}(s) = \frac{\omega(s)}{V_{in}(s)}$$

$$G_{speed}(s) = \frac{K_t}{\{[(L_a s+R_a)(J_m s+b_m)+K_t K_b]\}}$$

$$G_{speed}(s) = \frac{K_t}{[(L_a J_m)s^2+(R_a J_m+b_m L_a)s+(R_a b_m+K_t K_b)]} \qquad (1)$$

where L_a is armature inductance, R_a is armature resistance, J_m is motor inertia and b_m is motor viscous damping. The equivalent mobile robot system open loop transfer function with load and gears attached, in terms of the input voltage, $V_{in}(s)$ and angular velocity, $\omega_{robot}(s)$ is given in Eq. (2).

$$G_{speed}(s) = \frac{\omega_{robot}(s)}{V_{in}(s)}$$

$$= \frac{K_t/n}{[(L_a J_{equiv})s^2 + (R_a J_{equiv} + b_{equiv} L_a)s + (R_a b_{equiv} + K_t K_b)]} \qquad (2)$$

In the calculations of total equivalent inertia, the inertias of the gears and wheels have to be included. Linear velocity of the mobile platform is obtained by multiplying angular speed *ω* robot by wheel radius, *r*.

Table I shows the nominal values for various parameters of DC motor and mobile robot platform.

Substituting all the parameters values to the system transfer function and applying the total torque to electric motor equation will result in a mathematical model of mobile platform dynamics. Eq.(3) is the open loop transfer function, relating the armature input terminal voltage, $V(s)$ in to the output terminal voltage of the tachometer, $V_{tach}(s)$, with most corresponding load torque applied. This equation is generated based on the equations that describe the DC motor, system dynamics and sensor modelling.

$$G_{speed}(s) = \frac{\omega_{robot}(s)}{V_{in}(s)} = \frac{K_s K_t}{(L_a s+R_a)(J_m s+b_m)+(L_a s+R_a)T+K_b K_t} \qquad (3)$$

where *T* is the disturbance torque. The transfer function of torques including coulomb friction is shown in Eq. (4).

$$G_{open}(s) = \frac{2K_sK_t}{As^2+Bs+C} \quad (4)$$

with

$A = 2B_{equiv}L_a$

$B = (r^2ML_a + 2B_{equiv}R_a + r^2MR_a + C_rL_a + 2J_{equiv}L_a)$

$C = 2K_bK_t + C_rR_a + 2J_{equiv}R_a$

B. PID Controller Design

Proportional-Derivative-Integral controller or commonly known as PID controller is widely used controller. PID is first in 1913 and stills a popular controller until today. The main advantages of PID are its simplicity, reliability and robustness.

TABLE I. MOBILE ROBOT PARAMETERS VALUE

Numerical Symbol	Description	Numerical Value
V_{in}	Input voltage	12 V
K_t	Motor torque constant	1.188 Nm/A
R_a	Armature resistance	0.156 Ω
L_a	Armature inductance	0.82 H
I_m	Geared-motor inertia	0.271 kgm^2
B_m	Geared-motor viscous damping	0.271 Nms
K_b	Motor back EMF constant	1.185 rad/s/V
n	Gear ratio	3
r	Wheel radius	0.075 m
h	Robot height	0.920 m
b	Robot width	0.580 m
d	Distance between wheels centres	0.4 m
J_{equi}	Total equivalent inertia	0.275 kgm^2
B_{equi}	Total equivalent damping	0.392 Nms
ω	Angular velocity	6.667 rad/s

PID controller is designed so that the time response of a system in terms of low overshoot percentage and a short settling time can be achieved [12].

Fig.2 shows the block diagram of a PID controller in a closed loop system. In this figure, *r(t)* is the user reference, *e(t)* is the error signal which is the differences between the measurement value *c(t)* and the reference value *r(t)*. The equation of PID controller which is expressed by *u(t)* in Fig.2 is shown in Eq. (5).

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e d\tau + K_d \frac{de}{dt} \quad (5)$$

where K_p, K_i and K_d are proportional gain, integral gain and derivative gain, respectively.

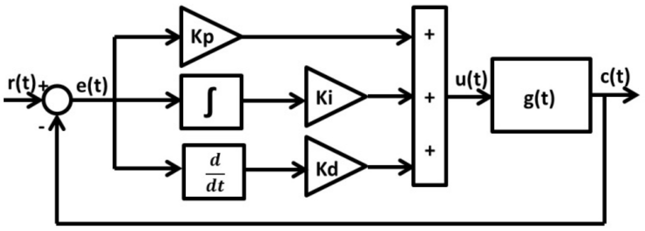

Figure 2. PID controller used in a closed loop system

Theoretically, by increasing the value of K_p, the system will have higher overshoot. Higher overshoot is unacceptable for certain system since it will decrease the life time of the system. K_i gain will counteract the offset. Increasing K_i will give faster response to the system.

However, if the response is too fast, the process will be prone to be unsteady. K_d gain is needed to keep the system to be under control. This shows that the selection of an appropriate K_p, K_i and K_d gains becomes critical in PID tuning process.

For analyzing the performance of PID controller, here are three commonly used performance criteria, including the integral square error (*ISE*), integral absolute error (*IAE*), and integral time absolute error (*ITAE*). In a MIMO system, they are defined by

$$\text{ISE} = \int_0^\infty (e_1^2(t) + e_2^2(t) + \cdots + e_n^2(t))dt \quad (6)$$

$$\text{IAE} = \int_0^\infty (|e_1(t)| + |e_2(t)| + \cdots + |e_n(t)|)\, dt \quad (7)$$

$$\text{ITAE} = \int_0^\infty t \cdot (|e_1(t)| + |e_2(t)| + \cdots + |e_n(t)|)\, dt \quad (8)$$

where e(t) is the error generated from the difference between reference and measurement values.

In this study, Integral Square Error(ISE) will be used as the performance index.

III. BAT ALGORITHM WITH MUTATION (BAM)

Bat Algorithm with Mutation (BAM) is a descendent of Bat Algorithm (BA) optimization which is combined with a mutation operator. BA is different from others swarm intelligence optimization techniques with the existence of loudness and pulse emission rate values in the updating value of BA parameter's equations. The nature of microbats which communicate by using the echolocation principle has been inspired the development of BA as an optimization algorithm.

BA algorithm performs better at the exploitation of the solution, but the explorations of BA were relatively poor. As a solution for the exploration problem in BA, mutation was inducing in Differential Evolution (DE) into BA to solve the optimization problem of PID controller for mobile robot.

Implementing mutation technique to BA will give two improvements. First is the improvement on exploration phase, where the new search space by the mutation of the DE algorithm can be achieved. Second is the exploitation phase, where the population information with BA will be improved.

Therefore, this approach can overcome the lack of the exploitation of the DE algorithm.

In BA, for every bat, its velocity and position in a d-dimensional search space can be written as follows:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{9}$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \tag{10}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{11}$$

where β in the range of [0, 1] is a random vector drawn from a uniform distribution and x_* is the current global best location. Every bat is randomly assigned a frequency which is drawn uniformly from $(fmin, fmax)$. A random walk around the current best solutions is known as the local search. Where a novel solution for every bat can be generated locally by

$$x_{new} = x_{old} + \varepsilon A^t \tag{12}$$

where $\varepsilon \in [-1, 1]$ is a scaling factor which is a random number, while $A^t = \langle A_i^t \rangle$ is the average loudness of all bats at time step. As the iterations proceed, the loudness and the rate of pulse emission will be updated accordingly by using Eq. (13).

$$A_i^{t+1} = \alpha A_i^t, \quad r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{13}$$

where α and γ are constants. In order make it to simple $\alpha = \gamma = 0.5$.

A main modification was done by adding mutation operator to the BA. Two minor modifications will be made. These modifications are made with the aim of speeding up convergence. By speeding up the convergence, the method will be more practical for a wider range of applications but without losing the attractive features of the original method. The 1st modification is fixing the frequency f and loudness A instead of various frequencies fi and amplitude Ai.

Similar to the original BA, in BAM, each bat is defined by its position x_i^t, velocity v_i^t, the emission pulse rate p_i^t and the fixed frequency f, loudness A in a d-dimensional search space. Eq. (10) and (11) are the new solutions x_i^t and velocities v_i^t at time step t, respectively.

A mutation operator is added to the original BA as the modification, in an attempt to increase diversity of the population. This step is taken in order to improve the search efficiency and speed up the convergence to optima. Once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk by Eq. (12). This is in condition that ξ is larger than pulse rate r, $\xi > p$, where $\xi \in [0, 1]$ is a random real number drawn from a uniform distribution. On the other hand, if $\xi \le p$, Eq. (14) that includes the mutation operator in DE is used to find the new solution to increase diversity of the population to improve the search efficiency.

$$x_{new} = x_{r1}^t + F(x_{r2}^t - x_{r3}^t) \tag{14}$$

where, F is the mutation weighting factor, while, r_1, r_2 and r_3 are uniformly distributed random integer numbers between 1 and NP.

The pseudocode of BAM optimization is as follow:

```
Begin
  Initialization. Objective function $f(x)$, $x = [x_1, x_2, \dots, x_d]^T$; Bat population NP contains of bats $x_i (i = 1,2,\dots,n)$ and $v_i$; Pulse frequency $f_i$ at $x_i$; Pulse rates p, the loudness $A_i$ and weighting factor F.
  Evaluate the fitness function for each bat in P(Eq.6)
  While (t < Max number of iterations )
     Sort the population of bats P from best to worst by order of fitness for each bat
     for i=1:NP(all bats) do
        Select uniform randomly $r_1 \neq r_2 \neq r_3 \neq i$
        $r_4 = [NP * rand]$
        $v_i^t = v_i^{t-1} + (v_i^t - x_*) \times Q$
        $x_i^t = x_i^{t-1} + v_i^t$
        if (rand > r) then
           $x_u^t = \boldsymbol{x_*} + \boldsymbol{\alpha\varepsilon^t}$
        else
           $x_u^t = x_{r1}^t + \boldsymbol{F}(x_{r2}^t - x_{r3}^t)$
        end if
        Evaluate the fitness for the offsprings $x_u^t$, $x_i^t$ & $x_{r4}^t$
        Select the offsprings $x_k^t$ with the best fitness among the offsprings $x_u^t$, $x_i^t$ & $x_{r4}^t$
        if (rand < A) then
           $x_{r4}^t = x_k^t$
        end if
     end for i
     Evaluate the fitness function using new offsprings
     Sort the population of bats P from best to worst by order of fitness for each bat
     $t = t + 1$
  end while
  Post-processing the results and visualization
End
```

Fig.3 shows the block diagram of PID optimization using BAM for mobile robot.

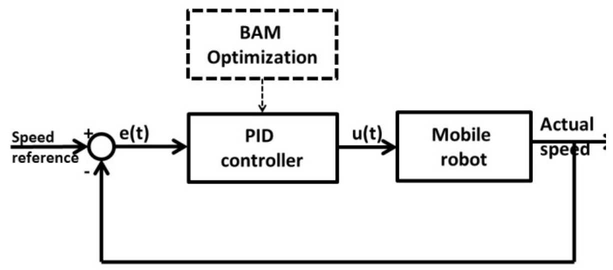

Figure 3. PID Controller optimization using BAM for mobile robot.

IV. Result and Discussion

This section will provide the explanation on the result obtained in this study. The simulation was conducted to evaluate the performance of BAM in optimizing the PID controller gain in mobile robot.

BAM parameters used in this study are shown in Table II. By utilizing these parameters, the optimized PID controller gain is $K_p = 135.92$, $K_i = 20.85$ and $K_d = 15.75$. Fig.4 shows the time response of the system. It is seen from the graph that the system has about 2.29% overshoot. Even though, overshoot exists in the response, the value is still considered to be small. This will not give severe effect to the system. In terms of steady state error, optimized PID gain gives 0.046 of Integral Square Error (ISE) which makes the system to have 2.94% steady state error. Additionally, the settling time is about 4 seconds.

TABLE II. BAM PARAMETERS VALUE

Parameters	Values
Number of population *(NP)*	150
Maximum number of iteration *(i)*	100
Pulse freqency *(f_i)*	0.5
Pulse rate *(p)*	0.5
Loudness *(A)*	0.5
Weighting factor *(F)*	1.7

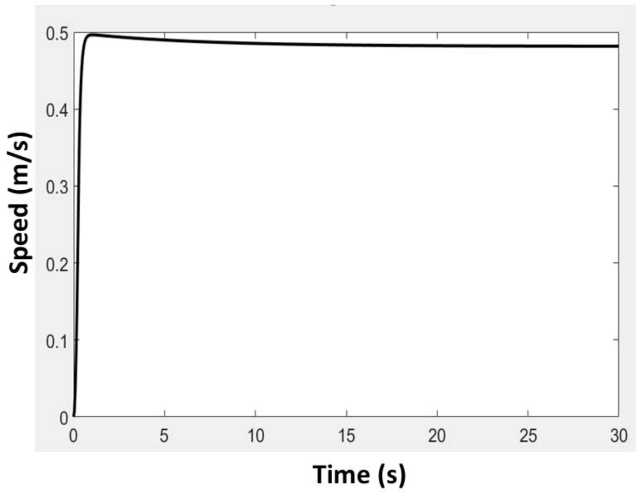

Figure 4. Time response of mobile robot with PID optimized using BAM

Next simulation is on the performance of BAM with different number of population. 50, 100 and 150 are chosen for this analysis. Fig.5 shows the response of the system by using these three different numbers of population. Referring to Fig.5, it is seen that 150 populations gives the best result. In 50 populations, the optimized PID gain gives 3.7% steady state error, 19.13% overshoot. This high overshoot is not preferable since it can reduce the life time of the mobile robot and possibility crash with other nearby objects. In 100 populations, the system has 3.58% steady state error and about 8.92% overshoot. Steady state error is not much different with the one obtained by using 150 populations. However, the overshoot percentage is much higher than 150 populations.

In order to compare the performance of BAM, Particle Swarm Optimization (PSO) is used. PSO is a well-known optimization algorithm that was developed by Kennedy and Eberhart in 1995. The social behaviour of a school of fish or a flock of birds, called the *swarm* is the working principle of PSO [13]. The parameters setting for PSO is listed in Table III.

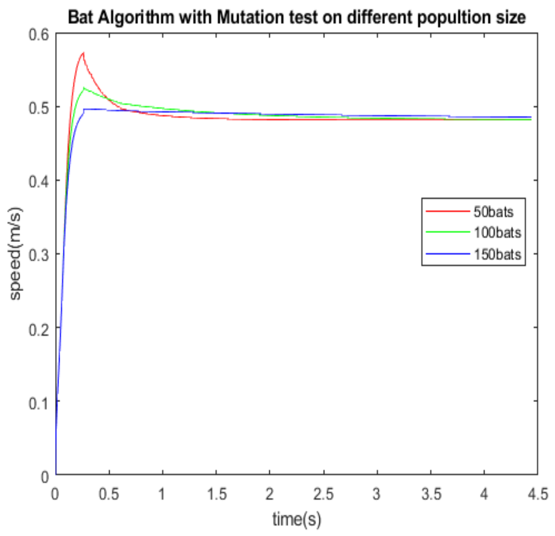

Figure 5. Time response of mobile robot with PID optimized using BAM with 50, 100 and 150 populations.

TABLE III. PSO PARAMETERS VALUE

Parameters	Values
Number of Population *(NP)*	150
Maximum number of iteration *(i)*	100
Weight inertia *(w)*	0.5
Learning factors, *c1* and *c2*	2

Fig.6 shows the comparison of time performance between system optimized using BAM and PSO. From Fig.6, it is seen that PSO has 3.07% overshoot and 3.72% steady state error. Meanwhile, BAM gives 2.29% overshoot and 2.94% steady state error. By looking at this result, there is no significant difference between PID gain tuned using PSO and BAM. This result shows that BAM and PSO are comparable and can be applied for tuning PID controller in mobile robot.

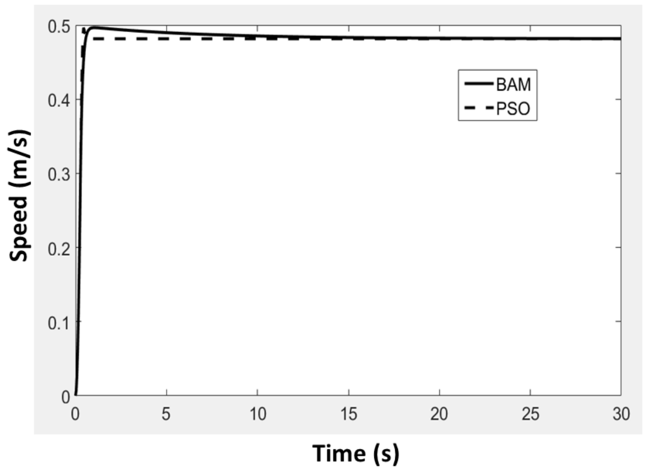

Figure 6. Comparison between BAM and PSO

V. CONCLUSION

PID controller design for mobile robot by using Bat Algorithm with Mutation (BAM) has been presented in this paper. BAM is an improved algorithm inspired from Bat Algorithm (BA). Mutation operator is introduced in the original BA to overcome the performance in exploration phase. Implementing BAM to optimized the PID controller gain in mobile robot gives good performance in term of low overshoot percentage (2.29%), low steady state error (2.94%) and fast settling time (4 seconds). Additionally, BAM is also comparable with the well-known optimization algorithm, Particle Swarm Optimization (PSO). The result from PSO gives 3.07% overshoot, 3.72% steady state error and 0.71 seconds of settling time.

ACKNOWLEDGMENT

This works is supported by Universiti Malaysia Pahang (UMP), under Universiti Malaysia Pahang Research Grant RDU 1703142 and RDU 170378.

REFERENCES

[1] P.M. Meshram, R.G. Kanojiya, "Tuning of PID Controller using Ziegler-Niclos method for speed control of DC motor", *Proceeding of International Conference on Advance in Engineering, Science and Management pp:117-122, 2012.*

[2] T. T. T. Kawabe, "A real coded genetic algorithm for matrix inequality design approach of robust PID controller with two degrees of freedom," *IEEE International Symposium on Intelligent Control,* pp. 119-124, 1997.

[3] M. S. K. S. S. a. B. Dashti, "Tuning of digital PID controller using particle swarm optimization," *In Proceedings of the 29th Chinese Control Conference,* pp. pp. 3383-3389, 2010.

[4] M. T. L. a. K. Solihin, "Tuning of PID controller using particle swarm optimization (PSO)," *International Journal on Advanced Science, Engineering and Information Technology,* no. 1(4), pp. pp.458-461, 2011.

[5] P. S. L. a. S. Dash, "Automatic generation control of multi area thermal system using Bat algorithm optimized PD–PID cascade controller," *International Journal of Electrical Power & Energy Systems,* vol. 68, pp. pp.364-372., 2015.

[6] N. K. P. a. N. Katal, "Optimal PID controller for coupled-tank liquid-level control system using bat algorithm," *International Conference on Power, Control and Embedded Systems (ICPCES),* pp. pp. 1-4, 2014, December.

[7] B. Muhammad, D. Pebrianti, N.A. Ghani, N.H.A Azis, N.A.A. Azis, M. S. Mohamad, M. I. Shapiai, Z. Ibrahim, "An application of Simulated Kalman Filter optimization algorithm for parameter tuning in Proportional-Integral_derivative Controller for Automatic Voltage Regulator System", *Proceeding of International Symposium on Control System 2018,* 2018.

[8] J. Kaliannan, A. Baskaran, N. Dey, A. S. Ashour, "Ant colony optimization algorithm based PID controller for LFC of single area power system with non-linearity and boiler dynamics", *World Journal of Modelling and Simulation,* Vol.12, No.1, pp:3-14, 2016.

[9] R. Muniraj, M. S. W. Iruthayarajan, R. Arun, "Tuning of Robust PID controller with filter for SISO system using Evolutionary Algorithms", *Studies in Informatics and Control,* Vol.26, No.3, pp:277-287, 2017.

[10] S. Chatterjee, V. Mukherjee, "PID controller for automatic voltage regulator using Teaching-Learning Based Optimization technique", *International Journal of Electrical Power & Energy System,* Vol.77, pp:418-429, 2016.

[11] F. Salem, "Refined models and control solutions for mechatronics design of mobile robotic platforms," *Estonian Journal of Engineering,* vol. 19(3), 2013.

[12] K.J. Astrom, T. Hagglund, "The future of PID control", *Control Engineering Practice,* Vol.9, No.11, pp:1163-1175, 2001.

[13] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", *Proceeding of IEEE International Conference on Neural Network. IV,* pp: 1942-1948, 1995.