Two-Wheeled LEGO EV3 Robot Stabilisation Control Using Fuzzy Logic Based PSO Algorithm

M. F. Maharuddin, N. M. Abdul Ghani and N. F. Jamin
Faculty of Electrical & Electronic Engineering,
Universiti Malaysia Pahang, 26600 Pekan Pahang
normaniha@ump.edu.my

***Abstract*—This paper presents a control system design to stabilise a two-wheeled Lego EV3 robot. This robot is developed based on the inverted pendulum. The mathematical modelling is derived based on this robot and using Euler Lagrange equation and represented in Simulink block diagram. The fuzzy logic controller is used to stabilise this robot with Particle Swarm Optimization algorithm for optimum performance of the system. The result of the fuzzy logic controller without optimisation is compared with the fuzzy logic controller with optimisation. Using the Simulink block diagram, the result of optimum tilt angle and control input signal are presented. The results show that the fuzzy logic controller with optimisation is able to improve the performance of the solution when compared to the fuzzy logic controller without optimisation.**

***Index Terms*—Lego EV3 Robot; Fuzzy Logic Control; Particle Swarm Optimization; Two-Wheeled Stabilisation.**

I. Introduction

An inverted pendulum system is very common in control engineering world. Research has been done to stabilise this inverted pendulum. This unstable system has their own uniqueness and can be applied to various applications of technology. Based on this inverted pendulum, researchers have introduced a lot of technology and solve problems. An inverted pendulum is a pendulum that has its centre of mass above its pivot point. It usually mounted on a cart that can move horizontally. The inverted pendulum is unstable and must be actively balanced to remain upright.

Nowadays, many problems can be solved based on this inverted pendulum system. People's life become easier, and it requires less energy. The two-wheeled system takes a little space compared to the four-wheeled system. Besides, the movement is smoother and faster. Many technologies use inverted pendulum system as their foundation such as Segway Transport. As a two-wheeled self-balancing transport and powered-up by battery, the Segway is invented by Dean Kamen. This Segway is controlled by body position of the user, include steering, acceleration and breaking [1]. The user will move the Segway forward or backwards by using their weight forward or backwards on the platform. The advantage of this transportation is that it can reduce fatigue that caused by walking. Besides, it is an eco-friendly machine and low operating cost because Segway does not use fuel. However, the disadvantages for Segway are the item is expensive, heavy and the user did not know how far he or she can travel based on his or her mass.

Another technology that adopted the two-wheeled inverted pendulum as their foundation is the two-wheeled wheelchair system. This is an upgraded version from four-wheeled wheelchair system. It has many advantages such as the user can move from wide space to narrow place. Besides, the position of the user and the eye contact with the normal person become same, and it will improve their self-motivation [2]. This wheelchair system is based on the two-wheeled inverted pendulum in term of stabilisation.

The two-wheeled robot system is a very complex nonlinear system, and it is difficult to control. Researchers have solved many problems based on the idea of inverted pendulum such as robotic wheelchairs, personal transport system and many more in recent years [3]. Previously, many optimisation algorithms have been used to optimise the control performance of the system such as Gravitational Search Algorithm, Particle Swarm Optimization algorithm and Simulated Annealing algorithm [4].

Optimization of fuzzy logic control has been done using Genetic Algorithm, GA for two-wheeled stabilisation [5]. The ant-colony optimisation can be used to control the ball beam system using fuzzy sliding mode control [6]. To further improve the control performance, an improved ant colony optimisation (ACO) is proposed to optimise the controller parameters. The proposed ACO algorithm has the enhanced capability of fuzzy 12 pheromones updating and adaptive parameter tuning. Particle Swarm Optimization, PSO has been used intensively in optimisation engineering system [7].

The objective of this paper is to optimise the scaling factors of the fuzzy controller for the inverted pendulum. The synthetic error and synthetic error variety are introduced to reduce the input varies and the control rule of the fuzzy controller without losing the information [7]. Then, PSO is used to search the scaling factor which adjusts the fuzzy controller to the system. It was shown that system is stable, and it takes a short time to get the good performance. The PSO has improved the result of the system. In addition, PSO also had been used successfully to stabilise the double-link inverted pendulum system of the two-wheeled wheelchair. The researcher applied PSO to optimise the LQR controller to get an optimum parameter of the controller [12]. Synthetic error and synthetic error variety also reduce the input varies, and the control rule of fuzzy control and PSO is introduced to optimise the parameters of the fuzzy controller. Besides, PSO is one of the simple optimisation to get an optimum parameter in the algorithm.

In this paper, a two-wheeled Lego EV3 Robot which inspired by the inverted pendulum on two wheels system is controlled using Fuzzy logic based PSO algorithm. The PSO algorithm is used for fuzzy logic control parameter tuning. The selection of PSO algorithm is chosen due to easy implementation and has very good performance efficiency. This will provide the better result to stabilise and control the two-wheeled robot system. Then, the results are compared between fuzzy logic without optimisation and fuzzy logic with Particle Swarm Optimization algorithm.

II. Two-Wheeled Robot

The robot that used in this project is Lego EV3 balancing robot which mimics the two-wheeled inverted pendulum as sown in Figure 1. It has only one link to stabilise the system; the robot must capable to move forward and backwards. The control input signal that will move the robot is in voltage, and the output for the robot is tilt angle. The angle must be in 0 degrees to make the robot stable in the upright position. The modelling of the robot is derived based on the Lego EV3 robot using Euler Lagrange Equation.

Based on the derivation, Ψ is the body pitch angle that uses gyro value, θ is the wheel angle that use encoder value and ∅ is the yaw of bthe ody that use encoder difference. The input motor voltage of both motor is same due to the stabilization system. The parameters of the robot have been tabulated in Table 1. To derive the modelling, the Euler Lagrange equation was used [3]:

$$T_1 = \frac{1}{2}M(\dot{x}_b^2 + \dot{z}_b^2) + \frac{1}{2}m(\dot{x}_m^2 + \dot{z}_m^2)$$
$$T_2 = \frac{1}{2}J_w\dot{\theta}^2 + \frac{1}{2}J_\Psi\dot{\Psi}^2 + \frac{1}{2}J_m(\dot{\theta} - \dot{\Psi})^2$$
$$U = mgz_m + mgz_b$$

T_1 = translational kinetic energy
T_2 = rotational kinetic energy
U = gravitational potential energy

Lagrangian L_g and Lagrange equations are given as follows:

$$L = T_1 + T_2 - U$$
$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}}\right] - \frac{\partial L}{\partial \theta} = F_\theta$$
$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\Psi}}\right] - \frac{\partial L}{\partial \Psi} = F_\Psi$$

$The\ state\ x(t) = [\theta \quad \Psi \quad \dot{\theta} \quad \dot{\Psi}]$

So, the state space representation of the system is:

$$E_p\dot{x} = A_px + B_pu, \qquad y = C_px$$

$$E_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & (2m+M)R_W^2 + 2J_W + 2J_m & MLR - 2J_m \\ 0 & 0 & MLR - 2J_m & ML^2 + J_\Psi + 2J_m \end{bmatrix}$$

$$A_p = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -2f_W - 2\beta & \beta \\ 0 & ML_g & \beta & -\beta \end{bmatrix}$$

$$B_p = \begin{bmatrix} 0 \\ 0 \\ 2\alpha \\ -2\alpha \end{bmatrix}$$

$$C_p = [0 \quad 1 \quad 0 \quad 0]$$

Or it can be written as:

$$\dot{x} = A_x + B_u, \qquad y = C_x$$

Figure 1: Two-wheeled balancing EV3 LEGO robot

Table 1
Parameter of the Two-Wheeled EV3 LEGO [3]

Symbol	Quantity	SI unit	Name
g	9.81000	[m/sec^2]	Gravity constant
m	0.02200	[kg]	Mass of wheels
R	0.02900	[m]	Radius of wheels
M	0.61000	[kg]	Body weight
W	0.15600	[m]	Body width
D	0.05000	[m]	Body depth
H	0.25000	[m]	Body height
L	H/2	[m]	Distance of the centre of mass from the wheel axle
J_φ	M L^2/3	[kgm^2]	Body pitch inertia moment
J_m	0.00001	[kgm^2]	DC motor inertia
R_m	6.69000	[Ω]	DC motor resistance
K_b	0.46800	[Vsec/rad]	DC motor back emf constant
K_t	0.31700	[Nm/A]	DC motor torque constant
n	1.00000		Gear ratio
f_m	0.00220		Friction coefficient between body and DC motor
f_w	0		Friction coefficient between wheel and motor

III. Fuzzy Logic Based PSO

Fuzzy control system is a control system based on fuzzy logic. It is a mathematical system that analyses analogue input in terms of logical variables. The term fuzzy refers to the fact that the logic involved can deal with concepts that cannot be expressed as “true” or “false” but rather as “partially true” [8]. This fuzzy logic has advantages that the solution to the problem can be cast in terms that humans can understand. A fuzzy controller is easily modified and can use multiple input and output. The collection of rules is called a rule-based which like if-then format. Figure 2 shows the fuzzy block diagram that is used in this project. This will represent the system in feedback type. In the fuzzy controller block, there are three control parameter gains. The function of gain is to convert real value before entering fuzzy logic and convert the value after entering fuzzy logic into real value. The PSO algorithm will optimise this value, and it will affect the output of the system. These gains in this Simulink block diagram which are for control the settling time, control the control input signal and control the oscillation of the tilt angle graph. Figure 3 shows the diagram for gain.

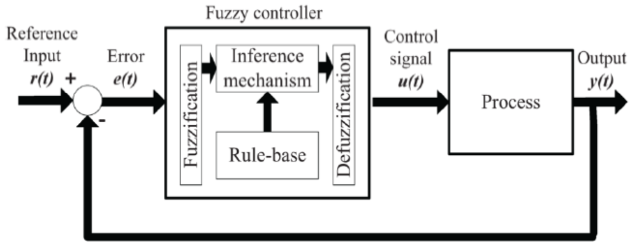

Figure 2: Fuzzy logic control diagram

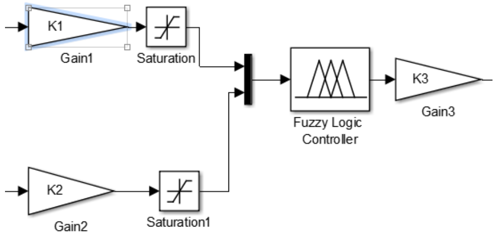

Figure 3: Control parameter gains

A. Fuzzy Rule

Fuzzification is a process where the values are changed or transformed into grades of membership for linguistic terms of fuzzy sets. This is a very important step to get the correct value.

The value of input 1 and input 2 is the error and change of error of tilt angle. In this fuzzy controller, 5x5 membership function is utilized. This system used five fuzzy variables which is negative big (NB), negative small (NS), Zero (ZE), positive small (PS) and Positive Big (PB). The fuzzy variable is same for input 1 and input 2. This system uses five fuzzy variables because it is more precise. The Gaussian Curve membership is selected for input 1, input 2 and output 3 because it is more precise compared to another shape. Then, the range and parameter of membership in fuzzy logic control are selected based on the previous paper that has been studied.

In FLC, the Rule base uses IF-THEN instruction. The example of instruction is “IF INPUT THEN OUPUT”. Input is the value that has been entered before. Fuzzy logic mimics the human thinking to make a conclusion from given data and focus on decision-relevant information. The operator can be OR, AND or XOR.

Defuzzification is the process where it produces a result in fuzzy logic when the fuzzy sets and corresponding membership degree is executed. The rules will transform some variables into a fuzzy result. In this project, the centroid is chosen for defuzzification process. The result from the rules must be added together. If the rules touched at the centre of the Gaussian shape, the top of the Gaussian would be removed, and the remaining portion will be taken. The remaining shape is added and calculated by using a formula. The value after using the formulae is the defuzzification value. For this stabilisation of Lego EV3 robot, Gaussian-type is chosen, and the parameter can be edited whether the user wants to increase or decrease the area of the shape. Finally, the range of the membership function can be edited, and for this project, the range is between -1 until 1. The surface must be smooth, so it indicated that the system functions smoothly with the good decision of rule inference.

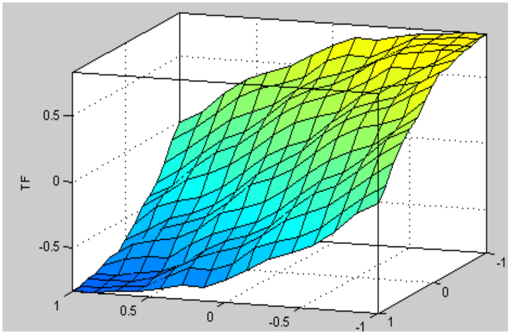

Figure 4: Surface Viewer

B. PSO Algorithm

This project uses particle swarm optimisation as an algorithm that will optimise the value to the optimum level. This algorithm has applied in a lot of paper to solve many problems such as data mining, signal processing and many more. Particle Swarm Optimization is inspired by the behaviour of a swarm of birds and schools of fish, and it is swarming intelligence meta-heuristic. Imagine a group of birds that circling an area when they can smell food. Birds that situated near the food will chirp the loudest, and the other birds will come near to the sound. Then, if other birds come close to the food than the first bird, it will chirp loudest. This pattern will continue until one of the birds found the food. It is a population-based method, and this optimisation will be modified to the best value until it reacts to the termination criteria.

The population of the solution is called a swarm, and the feasible solution is called particles. The particles will move around the space to find the optimum value [10]. PSO does not change the population from generation to generation, but they will be updating their position and velocity. The member of the population will influence each other. This algorithm will keep track of three global variable which is the target value, Global best value and stopping value. Imagine in one space; there are five particles. For the first iteration, all particles will move around and calculate the fitness value. Each particle will have a value of fitness. Then, for the second iteration, the particles will move around, and they will find the new value. The value for each particle is call pBest. If the value of pBest is better than before, it will assign current fitness as new fitness. If not, the value of the previous pBest will be kept. Then, among the pBest, one value will be taken to assign as gBest. gBest is the best value among pBest. Then, velocity will be calculated, and the particles will move to the target based on the velocity that has been calculated. Particles that far for the target will require a lot of velocities to get to the target.

The function of inertia weight is to control the exploration and exploitation of the particles. At the beginning of the iteration, the particles will explore all over the space, and after some time, the particles will focus on the places that near to the target. Suitable selection of inertia value will give balances between global and local exploration. If inertia value is too small, the ability of the particles to explore will be eliminated. As originally developed, inertia weight often decrease linearly from about 0.9 to 0.4 during the run and is set according to the following equation [9].

$$\omega = \frac{\omega_{max} - \omega_{min}}{iter_{max}} \; x \; iter$$

The velocity in the PSO algorithm will be updated for each iteration [11]. The particles will move according to the value of the velocity. The formulae for the velocity is:

velocity = inertia x velocity + c1 x (R1. x (local_best_position- current_position)) + c2 x (R2. x (globl_best_position- current_position)

where R1 and R2 is random number between 0 and 1 while c1 and c2 is coefficient and local_best_position is equal to pBest while globl_best_position is gBest. The PSO parameters used are shown below:

<u>Parameters for PSO Algorithm</u>
n = 300
Iteration = 300
Dimension = 3
C1 = 2
C2 = 2
Wmax = 0.9
Wmin = 0.4
K1 = 74.4841
K2 = 0.7279
K3 = 4.1128
Convergence curve = 14.6959

IV. RESULTS

Previously, people try a method that called trial and error method. They will put any number that will stabilise the system. This method requires a lot of time and patience. Besides, this method is not systematic and relevant. In this work, an optimisation method is used to compare with the Fuzzy Logic Controller with trial and error method. The output observed in this project is the tilt angle and the control input signal. The tilt angle must be at zero to make sure the robot is stable and not falling.

The optimisation will minimise the error in this system and will give the optimum value for gain to get the best result. The optimisation method for the Fuzzy Logic Controller is PSO algorithm. For Figure 5, the convergence curve is approaching 0. The starting value is very big, and after 300 iterations, it became 14.6959. For the tilt angle in Figure 6, at 0.5 seconds, the graph is approaching 0, and it means that the robot is already stable. Then, for the control input signal in Figure 7, it only needs 3V for the voltage to run the motor.

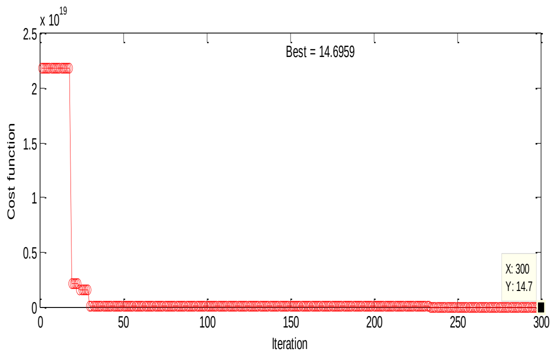

Figure 5: Convergence Curve

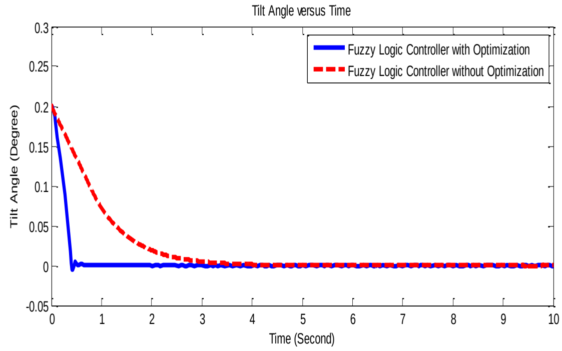

Figure 6: Tilt Angle versus Time

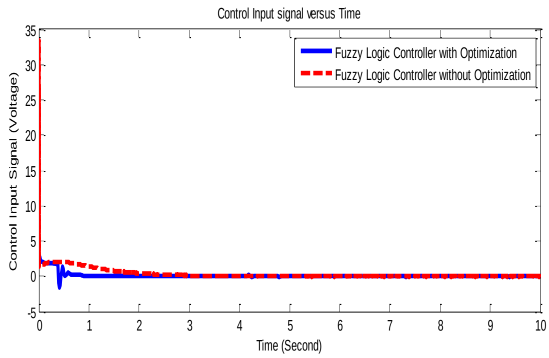

Figure 7: Control Input Signal versus Time

Based on the graph in Figure 6 and Figure 7, it is clearly seen that performance fuzzy logic controller with PSO algorithm is much better than fuzzy logic controller without optimisation. The settling time is faster for a controller with optimisation. Then, the robot needs less energy to move the motor, which is only 3 volt, compare to without optimisation that needs 35 volts.

V. CONCLUSION

For the conclusion, this project is based on the concept of the two-wheeled inverted pendulum. The optimisation technique is proposed to the controller to minimise the error. In this project, MATLAB software is used with Simulink block diagram to represent the system. The modelling of the two-wheeled Lego EV3 robot is derived based on the Euler Lagrange equation. The state space is obtained before proceeding to other works. To stabilise the robot, the tilt angle must be in 0 degrees. The initial method to stabilise the system is by using trial and error method. This method takes a long time and is not efficient. The Fuzzy Logic Control based on PSO algorithm is successfully designed and simulated. The PSO algorithm can improve the stabilisation of the two-wheeled Lego EV3 robot system to match the performance of the inverted pendulum system. Based on the result, the performance for a fuzzy logic controller with optimisation is better than fuzzy logic controller without optimisation. Finally, the result between controllers that use optimisation is compared with a controller without using the optimisation. Result shows that controller with optimisation performs much better compared to the controller without optimisation.

For future work, it is recommended that the optimisation is used for a two-link-system such as the two-wheeled wheelchair. The stabilisation can be implemented by using a fuzzy logic controller on hardware and in turns validates the output obtained from the simulation. Besides, other optimisation can be adopted and compared with PSO algorithm to evaluate which optimisation is better to stabilise the system.

ACKNOWLEDGEMENT

The work presented in the paper has been supported by Research Grant PGRS170344 from the Research and Innovation Department, Universiti Malaysia Pahang.

REFERENCES

[1] N. Ghani, N. Yatim, and N. Azmi, "Comparative assessment for two wheels inverted pendulum mobile robot using robust control," *Control Automation and Systems (ICCAS), 2010 International Conference on*, pp. 562–567, 2010.

[2] S. Ahmad and M. O. Tokhi, "Forward and backward motion control of wheelchair on two wheels," *2008 3rd IEEE Conf. Ind. Electron. Appl.*, pp. 461–466, 2008.

[3] Mathew, Sheelu Treesa, and S. J. Mija. "Design of H_2 controller for stabilization of two-wheeled inverted pendulum." In *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, pp. 174-179, 2014.

[4] R. E. Precup, R. C. David, E. M. Petriu, M. B. Rădac, S. Preitl, and J. Fodor, "Evolutionary optimization-based tuning of low-cost fuzzy controllers for servo systems," *Knowledge-Based Syst.*, vol. 38, pp. 74–84, 2013.

[5] S. Ahmad, M.O Tokhi and S.F Toha, "Genetic Algorithm Optimisation for Fuzzy Control of Wheelchair Lifting and Balancing", *Proceeding of Third UKSim European Symposium on Computer Modeling and Simulation*, 2009.

[6] Y.-H. Chang, C.-W. Chang, C.-W. Tao, H.-W. Lin, and J.-S. Taur, "Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3624–3633, 2012.

[7] D. Wang, G. Wang, and R. Hu, "Parameters optimization of fuzzy controller based on PSO," *2008 3rd Int. Conf. Intell. Syst. Knowl. Eng.*, pp. 599–603, 2008.

[8] D. T. Pham, a. H. Darwish, and E. E. Eldukhri, "Optimisation of a fuzzy logic controller using the Bees Algorithm," pp. 475–480, 2009.

[9] S. B. Chandra Debnath, P. Chandra Shill, and K. Murase, "Particle Swarm Optimization Based Adaptive Strategy for Tuning of Fuzzy Logic Controller.," *Int. J. Artif. Intell. Appl.*, vol. 4, no. 1, pp. 37–50, 2013.

[10] Z.-L. Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 384–391, 2004.

[11] Q. Bai, "Analysis of Particle Swarm Optimization Algorithm," *Comput. Inf. Sci.*, vol. 3, no. 1, pp. 180–184, 2010.

[12] A. Aula, S. Ahmad and R. Akmeliawati, "PSO-Based State Feedback Regulator for Stabilizing a Two-Wheeled Wheelchair in Balancing Mode," *10th Asian Control Conference, IEEE,* 2015.