Malaysian Journal of Mathematical Sciences 10(S) February: 265–275 (2016)
Special Issue: The 3rd International Conference on Mathematical Applications in Engineering 2014 (ICMAE'14)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES
Journal homepage: http://einspem.upm.edu.my/journal

Mathematical Modelling and Development of a Monoball Robot for Educational Purpose

Latif, M.H*[1], Md.Yusof, H.[1,2], and Toha, S.F[2]

[1] *Department of Mechatronics Engineering, International Islamic University Malaysia*

E-mail: hafizlatif736@gmail.com

ABSTRACT

This paper introduces the development of a LEGO based self-balancing robot for the purpose of teaching fundamental concept of mechatronics subjects. This is a low cost and portable experimental assembly which would be suitable to introduce the important engineering subject such as control systems, embedded processors, programming and modelling. These subjects are thought towards the end of the programme and seldomly incorporated together. In this paper, the mathematical modelling of a self-balancing robot is proposed. This self-balancing robot has the capability of balancing itself on a ball which is based on the inverted pendulum theory. The design of the robot was made by first, analysing the dynamics of the inverted pendulum and from there on, a mathematical model is developed using free body diagrams. To add to the challenge of developing this robot for students, the mathematical model of the robot to be developed and modelled must be based on the spherical wheel inverted pendulum. The mathematical model is then verified by comparing the behaviour of the constructed robot with simulated model. The effectiveness of the PID and the LQR controller is verified by simulations and experimental results. Finally the usefulness of the self-balancing robot using LEGO Mindstorms and Arduino controller as a medium of teaching and learning mechatronics subjects is demonstrated.

Keywords: LEGO, Self-balancing robot, Spherical wheel inverted pendulum, Education

1. Introduction

Research on self-balancing robot has gained interest from many researchers around the globe. Ballbot Model-Based Design, by Yorihisa Yamamoto (2009) Yamamoto (2009) which demonstrates control of a self-balancing robot on a ball developed using LEGO NXT Mindstorms kit has enticed the interest of robotic communities from all around the world. This kit has been widely utilised in many areas of education and research, covering from undergraduate projects to postgraduate researches Gawthrop and McGookin (2006), (Resnick and Silverman, 1996), (Talaga and Oh, 2009). Although LEGO Mindstorms was intended for youngsters over the ages of 10, it is the best way to begin with for a research of this kind. This is a hassle free, functional and cost effective way which could be translated into a prototype once satisfied with its features.

Monoball robot as referred to its name utilises a single ball for movement. This robot features enable it to execute its manoeuvre in a unique way, in the presence of external disturbances the robot has the ability to tilt spontaneously in any directions and balances itself. Monoball robot is kept in equilibrium over a single ball by means of two gyroscopes that provide the angular speed to measure the inclination and two motors that behave as actuators (Prieto et al., 2012). Based on a non-linear system, similar to the concept of inverted pendulum, this robot has a bright future to be developed much further because it offers reliable control and stability which make its manoeuvring more effective. Other than using Lego-based Monoball robot, there are also researches using two-wheeled coaxial robots which show outstanding advantages such as compact in mechanical structure, flexibility in motion and portability. Over the past decades, JOE (Grasser et al., 2002) and nBOT just to name a few and other similar robots have been reported over worldwide.

Rezero is an example of the advancement of the inverted pendulum robots exhibited by Fankhauser and Gwerder (2010) (Fankhauser and Gwerder, 2010),from Swiss Federal Institute of Technology Zurich. Their research have demonstrated a working simulation and unique non-linear controller which suitable for their Ballbot stabilization purposes since it is capable of stabilising the system in high performance situations. The research incorporates a number of sophisticated components involving power electronics, laser range sensor, Inertia Measurement Unit (IMU), omniwheels, brushless DC motors with encoders, and rubber coated ball with ball rack. The usage of IMU in their research has many advantages compared to the use of single accelerometer and gyroscope especially for the tilt and angular velocity measurement. This is because IMU comes with a complete built-in 3 gyroscopes, 3 accelerometers and a filter to compensate sensor drifts. Another research on the same interest has been performed in University of British Columbia by (Balasubramaniam and Lathiff, 2011). Their research employed a complementary filter algorithm using an embedded processor known as Cricket. The controller is built-in with 3-Axis accelerometer and 3-Axis rate gyroscope. Powered by ATmega-644 processor with a built-in dual 1.5A H-bridges allows the integration of both gyroscope and accelerometer to achieve the vertical balance of the robot's body. Additional complementary filter is used in this project provides a single usable value obtained from the synthesis of both data from gyroscope and accelerometer sensors. The output from the filter is intentionally designed to be dependent on the gyroscope data solely. The error values are calculated based on the output of the filter which is the pitch angle of the robot. These values are then fed into the PID controller. This research focuses on the motion control of a self-balancing robot. It is anticipated that the proposed features of the robot will be beneficial for lower level undergraduate students in Mechatronics engineering. This self-stabilizing robot could also be used as a medium for educational purposes in introducing Mechatronics System Design and also Control System and Robotics for IIUM engineering foundation students before going into depth of the related subjects such as control system, robotics, actuators and sensors and microprocessors.

2. Conceptual Design

Figure 1 illustrated the proposed design of the self-balancing robot rendered using Solidworks software. The real design might subject to minor changes. The dimension for the Monoball robot is 20cm x 20cm x 25cm (base area x height). The diameter of the ball is 14cm. The design was modelled in Solidworks environment prior to prototyping stage to have a clear visualization of the end product.This robot has two sensors; IMU and ultrasonic sensor. The IMU sensor has built-in 3-axis Gyroscope, 3-axis accelerometer and compass which will be used to measure inclination angle by means of angular speed provided. The Ultrasonic sensor is uses for obstacle avoidance purposes. For the drive system, two DC motors were used with 2A Motor shield controlled by an Arduino Uno microcontroller. The main body of the robot was constructed using LEGO block found in LEGO Mindstorms set.

3. Mathematical Modelling

To obtain the mathematical model of the plant, we adopted a conventional modelling procedure. The free body diagram is shown in Figure 2 below where ψ is the rotational angle of the body,$\theta$is the rotational angle of the wheels,

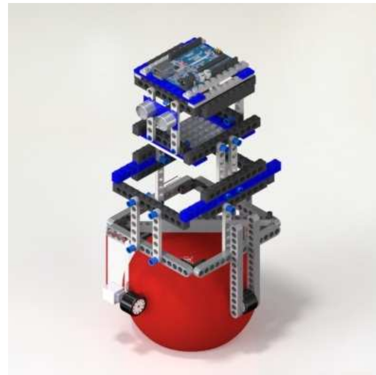

Figure 1: Monoball robot.

(x_b,z_b) is the position of inertia of the body, (x_s,z_s) is the position of the wheel center. R is the radius of the wheel and L is the distance between center of the wheel and the center of the inertia of the body. The main assumption made is the motion in XZ and YZ planes are decoupled and the equations that govern the motion in these planes are identical. The physical parameters are shown in Table 1.The equation of motion for the spherical wheel inverted pendulum system can be derived based on the coordinate system presented in Figure 2 as proposed by Yorihisa Yamamoto (2009). In order to establish the equation of motion for the system, the Lagrangian method will be used. Let say, value of θ= 0 at t=0, each coordinate in Figure 2 are given as shown below .

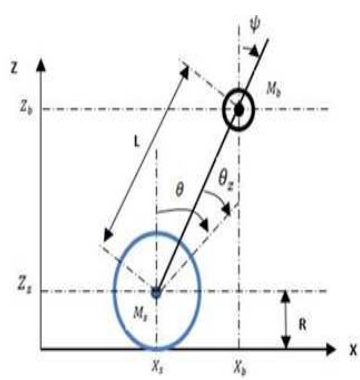

Figure 2: Spherical wheel inverted pendulum coordinate system.

Table 1: Monoball physical parameters

Description	Unit	Parameter
Gravity acceleration	m/sec^2	g=9.81
Mass of spherical wheels (Ball)	kg	M_S= 0.0516
Radius of spherical wheels	m	R_S=0.0146
Inertia moment of spherical wheel	kgm^2	J_S =7.333(10^{-6})
Motor wheel mass	kg	M_W =0.2155
Motor wheel radius	m	R_W = 0.0062
Motor wheel height	m	L_W =0.0124
Inertia moment of motor wheel	kgm^2	J_W =1.657(10^{-5})
Mass of the body	kg	M_b=0.4362
Width of the body	m	W=0.11
Depth of the body	m	D=0.125
Body center- spherical wheel center (dist.)	m	L=0.0508
DC motor resistance	Ω	R_m =17.647
DC motor back-EMF constant	Vsec/rad	K_b =0.63
DC motor torque constant	Nm/A	K_t =0.103
Body pitch inertia moment	kgm^2	J_ψ = 3.322(10^{-4})
Motor inertia moment	kgm^2	J_M =1 (10^{-5})
Friction coeff. (body- motor)		f_m =0.0020
Friction coeff. (spherical wheel-floor)		f_s =0

$$(x_s, z_s) = (R_s\theta, z_s) \tag{1}$$

$$(\dot{x}_s, \dot{z}_s) = (R_s\dot{\theta}, 0) \tag{2}$$

$$(x_b, z_b) = (x_s + L\sin\psi, z_s + L\cos\psi) \tag{3}$$

$$(\dot{x}_b, \dot{z}_b) = (R_s\dot{\theta} + L\dot{\psi}\cos\psi, -L\psi\sin\psi) \tag{4}$$

For this plant, the Lagrangian mechanics used as follows

$$L = T_1 + T_2 - U \tag{5}$$

Where T_1 is the translational kinetic energy,T_2 is the rotational kinetic energy and U represents the potential energy of the plant. The derivation of T_1, T_2

and U are given as the following

$$T_1 = \frac{1}{2}M_s(\dot{x}_s^2 + \dot{z}_s^2) + \frac{1}{2}M_b(\dot{x}_b^2 + \dot{z}_b^2) \tag{6}$$

$$\begin{aligned} T_2 &= \frac{1}{2}J_s\dot{\theta}_s^2 + \frac{1}{2}J_\psi\dot{\psi}^2 + \frac{1}{2}J_w\dot{\theta}_m^2 + \frac{1}{2}J_m\dot{\theta}_m^2 \\ &= \frac{1}{2}J_s\dot{\theta}_s^2 + \frac{1}{2}J_\psi\dot{\psi}^2 + \frac{1}{2}(J_m + J_w)\frac{R_s^2}{R_w^2}(\dot{\theta} - \dot{\psi})^2 \end{aligned} \tag{7}$$

$$U = M_s g z_s + M_b g z_b \tag{8}$$

By combining equations (6), (7) and (8), the resulting expression for L parameter is shown below

$$\begin{aligned} L = {} & \frac{1}{2}M_s(\dot{x}_s^2 + \dot{z}_s^2) + \frac{1}{2}M_b(\dot{x}_b^2 + \dot{z}_b^2) + \frac{1}{2}J_s\dot{\theta}_s^2 + \frac{1}{2}J_\psi\dot{\psi}^2 \\ & + \frac{1}{2}(J_m + J_w)\frac{R_s^2}{R_w^2}(\dot{\theta} - \dot{\psi})^2 - M_s g z_s + M_b g z_b \end{aligned} \tag{9}$$

Set θ and ψ as the generalized coordinate system. Then, apply the Lagrange equations

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = F_\theta \tag{10}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}}\right) - \frac{\partial L}{\partial \psi} = F_\psi \tag{11}$$

The state space equation of the plant could be obtained by linearizing the Lagrangian equation (10) and (11) at standing position

$$\begin{aligned} F_\theta = {} & [(M_b + M_s)R_s^2 + J_s + \frac{R_s^2}{R_w^2}(J_m + J_w)]\ddot{\theta} \\ & + [M_b L R_s \cos(\psi) - \frac{R_s^2}{R_w^2}(J_m + J_w)]\ddot{\psi} - M_b L R_s \dot{\psi}^2 \sin(\psi) \end{aligned} \tag{12}$$

$$\begin{aligned} F_\psi = {} & [M_b L R_s \cos(\psi) - \frac{R_s^2}{R_w^2}(J_m + J_w)]\ddot{\theta} \\ & + [M_b L^2 + J_\psi + \frac{R_s^2}{R_w^2}(J_m + J_w)]\ddot{\psi} - M_b g L \sin(\psi) \end{aligned} \tag{13}$$

Equation (12) and (13) can be further simplified by letting $k = R_s/R_w$ which results in equation (14) and (15)

$$\begin{aligned} F_\theta = {} & [(M_b + M_s)R_s^2 + J_s + k^2(J_m + J_w)]\ddot{\theta} \\ & + [M_b L R_s \cos(\psi) - k^2(J_m + J_w)]\ddot{\psi} - M_b L R_s \dot{\psi}^2 \sin(\psi) \end{aligned} \tag{14}$$

$$\begin{aligned} F_\psi = {} & [M_b L R_s \cos(\psi) - k^2(J_m + J_w)]\ddot{\theta} \\ & + [M_b L^2 + J_\psi + k^2(J_m + J_w)]\ddot{\psi} - M_b g L \sin(\psi) \end{aligned} \tag{15}$$

We assume that the body pitch is small enough and $\sin\psi \to \psi$; $\cos\psi \to 1$ and $\dot{\psi} \to 0$. This will result in

$$F_\theta = [(M_b + M_s)R_s^2 + J_s + k^2(J_m + J_w)]\ddot{\theta} + [M_b L R_s - k^2(J_m + J_w)]\ddot{\psi} \tag{16}$$

$$F_\psi = [M_b L R_s - k^2(J_m + J_w)]\ddot{\theta} + [M_b L^2 + J_\psi + k^2(J_m + J_w)]\ddot{\psi} - M_b g L \psi \tag{17}$$

Considering the torque of the DC motor and viscous friction, generalized force F_θ and F_ψ are as follows.

$$F_\theta = K_t i - f_m \dot{\theta_m} - f_s \dot{\theta} \tag{18}$$

$$F_\psi = K_t i - f_m \dot{\theta_m} \tag{19}$$

where i is the DC motor current and $\dot{\theta_m}$=$k(\dot{\theta}$ - $\dot{\psi})$ is the DC motor angular velocity. However, the DC motor current cannot be used directly to control the DC motor because it is based on the PWM which is the voltage based control. Hence, the relationship between i and v must be considered using DC motor equation. The internal friction of the motor is assumed to be zero thus negligible. The general motor model is adopted as in equation (20) where v is the DC motor voltage.

$$L_m i = v - K_b \dot{\theta_m} - R_m i \tag{20}$$

The motor inductance L_m is small enough to be negligible so

$$i = \frac{v - K_b k(\dot{\theta} - \dot{\psi})}{R_m} \tag{21}$$

Substituting equation (21) into equation (18) and (19) , the generalized force can be expressed in term of motor voltage

$$F_\theta = \gamma v - (\beta + f_s)\dot{\theta} + \beta\dot{\psi} \tag{22}$$

$$F_\psi = -\gamma v + \beta\dot{\theta} - \beta\dot{\psi} \tag{23}$$

where $\gamma = \frac{K_t}{R_m}$ and $\beta = k(\frac{K_t K_b}{R_m} + f_m)$ These equations can be expressed in matrix form shown in equation (24)

$$Tv = Q\begin{bmatrix}\ddot{\theta}\\ \ddot{\psi}\end{bmatrix} + R\begin{bmatrix}\dot{\theta}\\ \dot{\psi}\end{bmatrix} + S\begin{bmatrix}\theta\\ \psi\end{bmatrix} \tag{24}$$

Finally the matrices for Q, R, S and T are obtained

$$Q = \begin{pmatrix} (M_b + M_s)R_s^2 + J_s + k^2(J_m + J_w) & M_bLR_s - k^2(J_m + J_w) \\ M_bLR_s - k^2(J_m + J_w & M_bL^2 + J_\psi + k^2(J_m + J_w) \end{pmatrix}$$
$$R = \begin{pmatrix} \beta + f_s & -\beta \\ \beta & \beta \end{pmatrix}; S = \begin{pmatrix} 0 & 0 \\ 0 & -M_bgL\beta \end{pmatrix}; T = \begin{pmatrix} \gamma \\ -\gamma \end{pmatrix} \tag{25}$$

Let the input *u(t)* to the system as *u(t)*=v and the plant states,X=*x(t)*= $[\theta, \psi, \dot{\theta}, \dot{\psi}]^T$ Based on the equation (24), the state space representation of the system can be derived

$$\dot{x} = Ax + Bu \tag{26}$$
$$y = Cx + Du \tag{27}$$

where;

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A(3,2) & A(3,3) & A(3,4) \\ 0 & A(4,2) & A(4,3) & A(4,4) \end{bmatrix}; B = \begin{bmatrix} 0 \\ 0 \\ B(3) \\ B(4) \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{28}$$

where;

$$\begin{aligned} A(3,2) &= -M_bgLQ(1,2)/det(Q) \\ A(3,3) &= -[(\beta + f_s)Q(2,2) + \beta Q(1,2)]/det(Q) \\ A(3,4) &= \beta[Q(2,2) + Q(1,2)]/det(Q) \\ A(4,2) &= -M_bgLQ(1,1)/det(Q) \\ A(4,3) &= [(\beta + f_s)Q(1,2) + \beta Q(1,1)]/det(Q) \\ A(4,4) &= -\beta[Q(1,1) + Q(1,2)]/det(Q) \\ B(3) &= \gamma[Q(2,2) + Q(1,2)]/det(Q) \\ B(4) &= -\gamma[Q(1,1) + Q(1,2)]/det(Q) \\ det(Q) &= Q(1,1)Q(2,2) - Q(1,2)^2 \end{aligned} \tag{29}$$

These values from matrices A, B, C and D will be used in the controller design

4. Controller Design

In this paper, we test the plant with conventional controllers. Although there are many possibilities and choices, the controllers that we have considered are the PID and LQR controller. The implementation of LQR controller is done by utilized the weighting factor obtains through experimental data (trial and error). The LQR controller is being used due to its applicability in MIMO systems. The effect of optimal control depends on the selection of weighting matrices Q and R. A discrete LQR controller can be designed in Matlab using the command: K=lqr(A,B,Q,R) which will return the gain matrix K accordingly which will affect the close loop response. The easiest way to determine the matrix R and Q are by following assumption Q= C x C' and R = 1 since only angle of the motor that will be controlled to stabilize the robot.The complete design of LQR with observer base controller is illustrated in Figure 3 .The

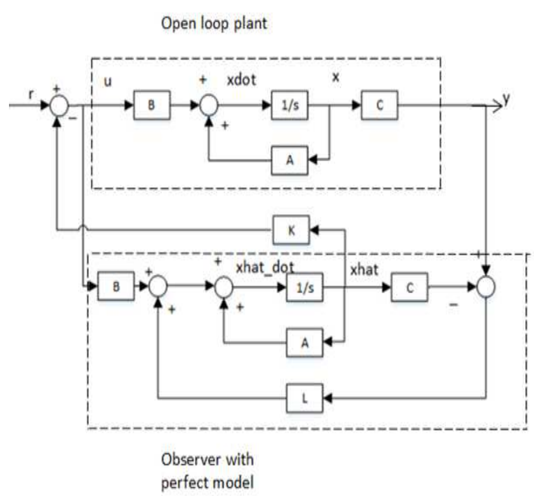

Figure 3: LQR with observer base control.

system has been proven to be controllable and observable through controllability and observability test. Hence, an observer based control can be used. The property of observability determines whether or not based on the measured outputs of the system we can estimate the state of the system. The error will approach zero if the matrix A-LC is stable. For the case for control, the speed of convergence depends on the poles of the estimator which is the eigenvalues of A-LC. To use state estimate as the input, the state estimate should converge faster that is desired from overall close-loop. That is, the observer poles to be faster than the controller poles. A common guideline is to make the estimator

poles 4-10 times faster than the slowest controller pole. The slowest pole has found to be -5.1 therefore the poles for state estimator must be 10 times faster than this slowest pole. The values for all parameters P= [-95000 -84 -84 -51]; Q=diag([8000 100 10 300]); R=1 ; K=[-89.4427 -610.1048 -36.133 -102.6363]
From Figure 4, the response of the LQR controller can be observed. The blue

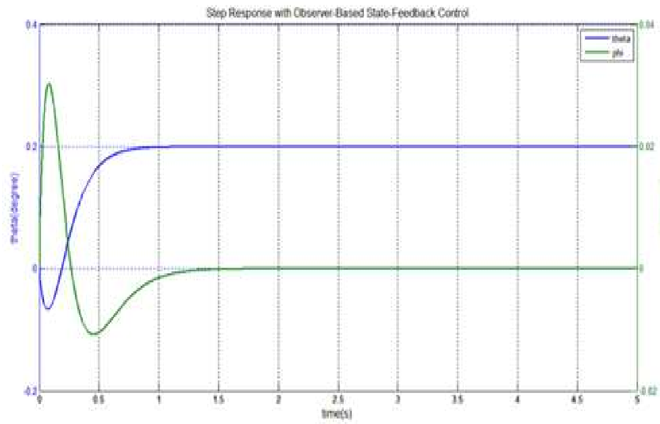

Figure 4: LQR with observer based control step response

line represented the Theta state (motor angle) and the green line represented the Phi state (body pitch angle) of the robot. The response of the plant shows significant improvement in term of settling time and percentage overshoot. The LQR controller then can be further implemented in Arduino microcontroller through coding implementation as all responses from testing shows satisfactory results.

Acknowledgments

I would like to take this opportunity to express my profound gratitude and deep regard to Dr. Hazlina Md. Yusof, for her exemplary guidance, valuable feedback and constant encouragement throughout the duration of the project. Her valuable suggestions were of immense help and her perceptive criticism kept me working to make this project in a much better way.

References

Balasubramaniam, P. and Lathiff, M. (2011). Self-balancing robot. *Engineering Physics Project Lab*, pages 1–23.

Fankhauser, P. and Gwerder, C. (2010). Modelling and control of a robot. Online reference, http://students,asl,ethz.ch/uplpdf/.

Gawthrop, P. and McGookin, E. (2006). Using lego in control education in pro. 7th ifa symp. *Advance in Control Education*, pages 31–38.

Grasser, F.and D'Arrigo, A., Colombi, S., and Rufer, A. C. (2002). Joe: A mobile, inverted pendulum. *IEEE Trans. Ind. Electron 29*, (1):107–114.

Prieto, S., Navarro, T., Plaza, M., and Rodrigues, O. (2012). A monoball robot based on lego mindstorm. *IEEE Control System Magazine*, pages 71–83.

Resnick, M., M. F. S. R. and Silverman, B. (1996). Programmable bricks: Toys to think with. *IBM Syst.K.35*, pages 443–452.

Talaga, P. and Oh, J. (2009). Combining aima and lego mindstorms in an artificial intelligence course to build real world robots. *J.Comput. Sci. Collage Arch.24*, (3):56–64.

Yamamoto, Y. (2009). Nxt ballbot model based design:control of self-balancing on a ball built with lego mindstorm nxt. *Cybernet Systems Co., Ltd*, (1):1–61.