Embedded LQR Controller Design for Self-Balancing Robot

Mustafa Engin
Department of Electronics Technology
Ege Higher Vocational School, Ege University İzmir, Turkey
mustafa.engin@ege.edu.tr

Abstract— **This study aims at designing an embedded controller that can stabilize custom designed self-balancing two-wheeled robot (TWSBR) using velocity feedback alone. A state-space model for the custom designed TWSBR is obtained through dynamic analysis using a Lagrangian method. System identification experiments are proposed and performed to identify the parameters of the model. Based on the identified model, an LQR controller, which is a combination of optimal estimation and optimal control, is designed and implemented that can successfully stabilize TWSBMR. At last, few tests are carried out to evaluate the performance of the controller, showing how the proposed LQR controller outperforms classical controller in many aspects.**

Keywords- embedded system; LQR controller; self-balancing two-wheeled robot; inertial measuring unit; complementary filter

I. INTRODUCTION

The TWSBR is one of the most worth-mentioning configuration of mobile robots that receiving special attention. They have some advantages over their four-wheeled counterparts that would make them more suitable for a variety of applications, such as being highly maneuverable, having low noise and being light weight. Due to its nonlinearity and intrinsically unstable dynamics property, a good controller is needed to keep the robot in upright position.

The main approach to TWSBR mathematical model development is Euler-Lagrange energy analysis [1]. The other common modelling approach is Kane's method which uses Newtonian free body derivations [2]. In [3], the yaw-axis dynamics are modelled and controlled by simulating partial feedback linearization, friction effects are neglected to simplify the feedback linearization transform. In [4], experimental results are obtained using a neural-fuzzy based approach. TWSBR balance has been accomplished using several feedback control strategies including root-locus PID gain assignments [5], feedback linearization [3], neural-fuzzy-logic compensation [4], adaptive PD control [6], LQR full state feedback [7-11], and a self-tuning PID neural network [12]. ad et al. proposed model predictive controller for position and tracking of TWSBR [13].

The main purpose of this paper is to design and construct a robot which will balance on two wheels and it should be able to carry a load. The robot should be able to work on its own without any supervisor. It should also be small so it doesn't require much space, is easy to carry and is flexible to display at the small space.

II. MATHEMATICAL MODELLING OF TWSBR

An accurate model will help us design a controller fit for the system, thus makes it possible to implement the controller on TWSBR. The schematic in Figure 1 shows TWSBR, where the pivot point can be manipulated by applying torque to the wheels in the direction of the acceleration of the center of mass. We use Lagrangian method to obtain equations of motion for the TWSBR. Take the center of mass of wheels as origin of a static coordinate system, X and Y direction are given as shown. θ and ϕ are angles of body and wheels, and both are positive in counterclockwise direction. The translational velocity of wheels is,

$$v_w = \dot{c}_w = \left(-r\dot{\emptyset}(t), 0\right) \tag{1}$$

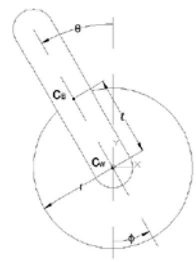

Figure 1. Simple mechanical drawing or robot.

Where: r is radius of wheels and cw center of mass of wheels. The translational velocity of body is,

$$v_B = \dot{c}_B = \left(-r\dot{\emptyset}(t) - l\cos\theta(t)\cdot\dot{\theta}(t), -l\sin\theta(t)\cdot\dot{\theta}(t)\right) \tag{2}$$

Where: c_B is center of mass of body and l is distance between two center of mass.

The kinetic energy of system can be described as [5]

$$K = \frac{1}{2}m_B \cdot {v_B}^2 + 2\cdot\frac{1}{2}m_W \cdot {v_W}^2 + \frac{1}{2}I_B \cdot \dot{\theta}^2 + 2\cdot\frac{1}{2}I_B \cdot \dot{\emptyset}^2 \tag{3}$$

Where: $m_B$is mass of body, $m_W$is mass of single wheel, I_m is motor armature inertia and I_B is inertia of body.

The potential energy of system can be described as

$$U = m_B g l \cdot \cos\theta \tag{4}$$

Lagragian, which is the difference between the kinetic and potential energy of the system is therefore:

978-1-5386-5683-9/18/$31.00 ©2018 IEEE

$$L = K - U \tag{5}$$

According to Lagrange equations of motion in Cartesian coordinates for a point mass subject to conservative forces, namely, [14]. Any non-conservative forces acting on the point mass would show up on the right hand side. So equation of motions are:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot\theta}\right) - \frac{\partial L}{\partial \theta} = -\tau(t) \text{ and } \frac{d}{dt}\left(\frac{\partial L}{\partial \dot\phi}\right) - \frac{\partial L}{\partial \phi} = \tau(t) \tag{6}$$

Where: τ is torque produced by motor and it can be calculated using motor parameters, motor speed ω_m and applied motor voltage $V(t)$. $\tau(t) = 2G_r(\bar{s}\cdot u - k\cdot\omega_m)$, Gr is gearbox ratio, $\omega_W = \dot\phi - \dot\theta = \omega_m/G_r$, u is normalized motor duty cycle and $u = V(t)/V_{max}$, V_{max} is maximum motor driver voltage, $\bar{s}$ is motor stall torque and k is motor constant. Substituting robot equations in to Lagrange equations

$$(I_B + m_B l^2)\ddot\theta(t) + m_B l\left(-g\sin\theta(t) + r\cos\theta(t)\cdot\ddot\phi(t)\right) = -\tau(t) \tag{7}$$

$$m_B r l\cos\theta(t)\cdot\ddot\phi(t) - m_B r l\sin\theta(t)\cdot\dot\phi(t)^2 + (2I_W + (m_B + 2m_W)r^2)\ddot\phi(t) = \tau(t) \tag{8}$$

To simplify, we set up following substitution rules. $a = 2I_W + (m_B + 2m_W)r^2$, $b = m_B rl$, $c = I_B + m_B l^2$, $d = m_B gl$, $e = 2G_r\bar{s}/V_{max}$, $j = 2{G_r}^2 k$. Solve for $\ddot\phi(t)$ and $\ddot\theta(t)$ in equations (8), then to obtain linearized model according to four states, equilibriums can be obtained by solving equations (8) when all angular speeds and accelerations are 0. There are two equilibriums of such an inverted pendulum system. Equilibrium 1 ($\theta(t) = 0$) is an unstable equilibrium, which we are interested in for control. While equilibrium 2, ($\theta(t) = \pi$), is a stable equilibrium can be achieved without any control. Thus, we linearized equations (8) at equilibrium 1 with notation rules. We plug in all the data which is taken from custom designed TWSBR.

$$\begin{bmatrix} \dot x_1(t) \\ x_2(t) \\ x_3(t) \\ x_{4(t)} \end{bmatrix} = \begin{bmatrix} -13.528 & 13.528 & 175.457 & 0 \\ 17.735 & -17.735 & -115.56 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_{4(t)} \end{bmatrix} + \begin{bmatrix} -90.456 \\ 118.585 \\ 0 \\ 0 \end{bmatrix} \cdot u(t) \tag{9}$$

$$\begin{bmatrix} y_1(t) \\ y_1(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_{4(t)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot u(t) \tag{10}$$

Equations (11) together with (12) are our linearized model. Where:

$$A = \begin{bmatrix} -13.528 & 13.528 & 175.457 & 0 \\ 17.735 & -17.735 & -115.56 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} -90.456 \\ 118.585 \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{11}$$

Therefore, we are ready to check the controllability and observability of TWSBMR model for further work.

The controllable matrix C_0 is of form

$$C_0 = [B \quad AB \quad A^2B \quad A^3B] \tag{12}$$

The observable matrix O_b is of form:

$$O_b = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} \tag{13}$$

Using Matlab command ctrb(A,B) and obsv(A,C) we know both of matrixes are of full rank, which implies the system is controllable as well as observable.

III. Controller Design

Two-wheeled self-balancing mobile robot is a single-input and multi-outputs (SIMO) system, which makes it hard to apply conventional approaches to control the system. However, we can directly deal with this SIMO system using modern control schemes such as LQR, which gives us a better understanding of stabilizing the whole system. Being familiar with classic controller for TWSBMR, which is a position feedback controller. We are seeking for a modern control strategy with velocity feedback.

A. Linear Quadratic Regulator

Linear Quadratic Regulator (LQR) problem is one fundamental problem of optimal control.

For a Linear Time Invariant (LTI) system:

$$\dot u(t) = Ax(t) + Bu(t) \tag{14}$$

Where x and u are the states and control.

LQR essentially generated an automated way of finding appropriate feedback controller:

$$u(t) = Kx(t) \tag{15}$$

Not only to stabilize the system, but also minimize a quadratic cost function, which of the form:

$$J = \int_0^\infty \left(x(t)^T Q x(t) + u(t)^T R u(t)\right) dt \tag{16}$$

Where Q and R are weighting matrix. The gain is K given as:

$$K = -R^{-1} B^T X \tag{17}$$

Where X is the solution to an Algebraic Riccati Equation (ARE)

$$A^T X + XA - XBR^{-1}B^T X + Q = 0 \tag{18}$$

Some assumptions here are: $Q > 0$, $R > 0$ and (A, B) is stabilizable. Q and R are weighting matrix that penalize the states and control work, respectively. Tuning elements of Q will change performance of the states correspondingly. In the same way, we can tuning R to determine the ability of control for each input. The relative ratio of Q and R elements will emphasize whether we care more about control or system performance. So it is very important to understand how to tradeoff between performance and control, and choose nice weighting matrix in a LQR design. Using MATLAB lqr command control law is obtained and results are shown in (21) and the scaled version in (22)

$$K = [1 \quad 483.6133 \quad 10.0038 \quad 20.3053] \tag{19}$$

$$K_1 = [0.0175 \quad 8.4406 \quad 10.1746 \quad 0.3544] \tag{20}$$

LQR controller Simulink model is shown Figure 2.

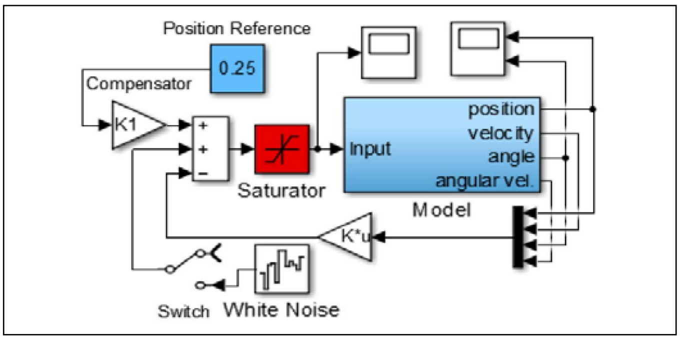

Figure 2. LQR Controller Simulink block.

B. Complementary Filter

The inertial measurement unit (IMU), MPU-6050 IMU was used, which incorporates a gyroscope and an accelerometer and is accessible using the I^2C communication bus. Because of accelerometer output is affected by high frequency oscillation due to the movements of the robot, and gyroscope output is affected by integration drift. Therefore, we need appropriate sensor fusion technique to combine measured data from accelerometer and gyroscope to obtain accurate result. As shown in Figure 3, the low pass filter eliminates high frequency components of noise coming from accelerometer. Similarly, the high pass filter eliminates the low frequency components of noise. Then, the outputs of the low pass and high pass filters are combined together to form tilt information which is free of oscillation noise and drift. Figure 3 illustrates the structure of the complementary filter. The mathematical representation of the complimentary filter is described through the following equation.

$$Y(k) = a(Y(k-1) + X_{GY}(k) * dt) + (1-a)X_{ACC}(k) \qquad (21)$$

Where: Y is output of filter, this time angle, X_{ACC} and X_{GY} are input of filter from accelerometer and gyroscope sensors outputs, a is filter coefficient, it depends on sample time, dt, and time constant, τ. Hence, $a = (\tau/(\tau + dt))$ and time constant important for filter. It defines cut-off frequency of filter. It can be calculated experimentally, by choosing acceptable maximum integral drift in the gyroscope angle. In this study time constant is defined $\tau = 0.005$ sec for sampling rate 100 Hz.

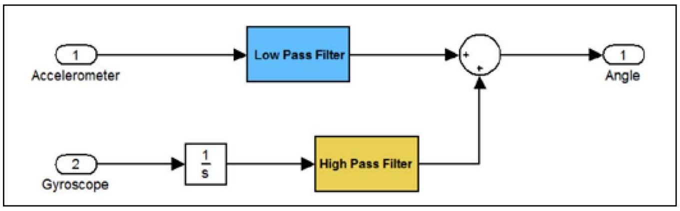

Figure 3. Complementary filter configuration.

C. Wheel Velocity

To measure angular velocity, angular position sensors MLX90316 are attached to each wheels of the robot. In fact these sensors measures the angular position of the wheel between 0° and 360°. Each sensor produce voltage in the range of 0.1Vref- 0.9Vref. The angle of the wheel can be calculated with ADC measurements using equation (24).

$$\varphi = (ADC_{READ} - ADC_{offset}) \cdot \alpha \qquad (22)$$

α is constant which depends on sensor and ADC. The angular velocity of the wheel can be calculated equation (25).

$$\omega = \frac{(\varphi - \varphi_{old})}{T_s} \qquad (23)$$

Where: T_s is sample time.

D. Software

The program was structured as a task dispatcher with four different periodic tasks, the controller itself, the wheel velocity estimator, Bluetooth communication and the robot body angle estimator. In order to create a periodic dispatcher systic timer interrupts in the Tiva board was setup with interrupt. Controller sample time was set to Ts=0.01s. The complementary filter sample time set to Ts=0.005s. Because, the estimators are running at a higher frequency. Complementary filter need certain number of iterations in order to get a reliable result. Wheel velocity estimator sample time was set to same value with complimentary filter. In fact it is not compulsory, since the maximum output frequency of wheel velocity sensors are 15 Hz this value appropriate for them. Main variables and their precision with rages are given in table 1. Tilt angle, θ, angular velocity $\dot{\theta}$, and wheel velocity, ω, are defined floating point and control signal is defined integer.

TABLE-1. MAIN VARIABLES AND RAGES, PRECISIONS.

Variable	Range	Type	Precision
θ	[-1, 1] rad	Floating point	5.3×10^{-3} rad/s
$\dot{\theta}$	[-3.5, 3.5] rad/s	Floating point	1.22×10^{-5} rad/s
ω	[-70, 70] rad/s	Floating point	3.91×10^{-3} rad/s
u	[-2048, 2048]	integer	1

Communication with the inertial measurement sensor is started using the I^2C bus and the accelerometer and gyroscope data are obtained on all three axes. After receiving this data, the complementary filter estimate tilt angle and body angular velocity. Bluetooth communication is checked. If data has been received two different operations will be performed. If the data is a number, the coefficients of the LQR controller will change, but if the data is a character, a movement direction is defined.

IV. RESULTS AND DISCUSSION

The frame was constructed and assembled by mechatronic engineering students according to the given custom design. The completed assembly can be seen in Figure 5, whereby wiring terminals were added for convenience and the signal and power wires are fed underneath the components in the gaps between the frames to limit the amount of wiring clutter. The design is entirely modular. The layer heights can be adjusted by choosing spacers of different lengths and the box for loads can be removed. The weight distribution of components on the frame was very close to being symmetrical over the axis perpendicular with the ground.

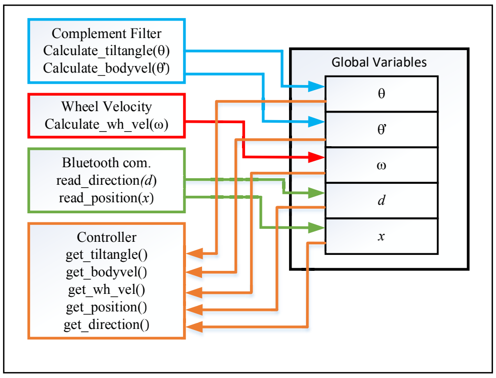

Figure 4. Communication between tasks.

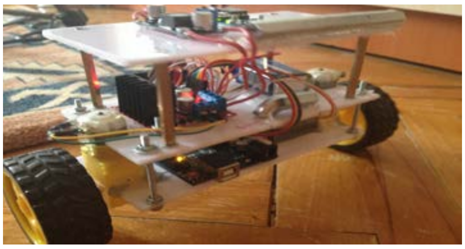

Figure 5. Completed Physical Assembly.

A. Moving Test

Moving test demonstrate the ability of the TWSBR to move forward and backward according to our command, so we can remote control the TWSBR rather than just balancing it. The idea of controlling moving is setting reference signal of ϕ to a non-zero value, a positive value drives TWSBR moving forward and a negative value drives TWSBR moving backward.

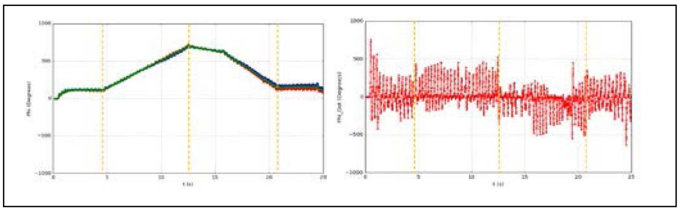

Figure 6. Φ and $\dot{\emptyset}$ responses of moving test.

Figure 6 is the ϕ and $\dot{\emptyset}$ responses, which are the wheel position and angular velocity responses. There are three lines here in blue, green and red. They stand for the two information for left wheel, right wheel and the average of wheels, separately. It is reasonable that they assembled each other since they share the same movement when moving forward and backward. We can tell that TWSBR stabilized itself in the first 5 seconds. Then from 5s to 12s, our command was sent to the processor and TWSBR moved forward. Subsequently, it went backward from 16s to 21s.

V. CONCLUSION

In this paper, we firstly extracted state-space model of two-wheeled self-balancing mobile robot based on a Lagrangian method. Then we managed to design a second-order and stable LQR controller based on velocity control alone. The controller has an ARM Cortex-M4 core microcontroller with floating point capability serving as processing unit. The outcome of producing a battery powered self-balancing robot that can be controlled and tuned wirelessly, and can recover from fairly large disturbances introduced to the system.

REFERENCES

[1] A. M. Almeshal, K. M. Goher, and M. O. Tokhi, "Dynamic modelling and stabilization of a new configuration of two-wheeled machines," *Robotics and Autonomous Systems,* vol. 61, pp. 443-472, 2013.

[2] S. Kim and S. Kwon, "Dynamic modeling of a two-wheeled inverted pendulum balancing mobile robot," *International Journal of Control, Automation and Systems,* vol. 13, pp. 926-933, 2015.

[3] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization," *IEEE Transactions on robotics,* vol. 21, pp. 505-513, 2005.

[4] K.-H. Su, Y.-Y. Chen, and S.-F. Su, "Design of neural-fuzzy-based controller for two autonomously driven wheeled robot," *Neurocomputing,* vol. 73, pp. 2478-2488, 2010.

[5] M. H. Khan, M. Chaudhry, T. Tariq, and U. Izhar, "Fabrication and modelling of segway," in *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*, 2014, pp. 280-285.

[6] A. Wasif, D. Raza, W. Rasheed, Z. Farooq, and S. Q. Ali, "Design and implementation of a two wheel self balancing robot with a two level adaptive control," in *Digital Information Management (ICDIM), 2013 Eighth International Conference on*, 2013, pp. 187-193.

[7] L. J. Pinto, D.-H. Kim, J. Y. Lee, and C.-S. Han, "Development of a Segway robot for an intelligent transport system," in *System Integration (SII), 2012 IEEE/SICE International Symposium on*, 2012, pp. 710-715.

[8] S. Nawawi, M. Ahmad, and J. Osman, "Real-time control of a two-wheeled inverted pendulum mobile robot," *World Academy of Science, Engineering and Technology,* vol. 39, pp. 214-220, 2008.

[9] J. Villacres, M. Viscaino, M. Herrera, and O. Camacho, "Controllers comparison to stabilize a two-wheeled inverted pendulum: Pid, lqr and sliding mode control," *moment,* vol. 2, p. 12, 2016.

[10] Y. Kim, S. H. Kim, and Y. K. Kwak, "Dynamic analysis of a nonholonomic two-wheeled inverted pendulum robot," *Journal of Intelligent & Robotic Systems,* vol. 44, pp. 25-46, 2005.

[11] M. Han, K. Kim, and J. Lee, "Implementation of unicycle segway using unscented Kalman filter in LQR control," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2013 10th International Conference on*, 2013, pp. 695-698.

[12] T.-J. Ren, T.-C. Chen, and C.-J. Chen, "Motion control for a two-wheeled vehicle using a self-tuning PID controller," *Control Engineering Practice,* vol. 16, pp. 365-375, 2008.

[13] H. S. Zad, A. Ulasyar, A. Zohaib, and S. S. Hussain, "Optimal Controller Design for Self-Balancing Two-Wheeled Robot System," in *Frontiers of Information Technology (FIT), 2016 International Conference on*, 2016, pp. 11-16.

[14] A. Ghaffari, A. Shariati, and A. H. Shamekhi, "A modified dynamical formulation for two-wheeled self-balancing robots," *Nonlinear Dynamics,* vol. 83, pp. 217-230, 2016.