LQR-Based Stabilization and Position Control of a Mobile Double Inverted Pendulum

D T Ratnayake1,*, and M Parnichkun1

1Industrial Systems Engineering, Asian Institute of Technology, Khlong Luang, Pathum Thani 12120, Thailand.

* Corresponding Author: dhanika.1004@gmail.com

Abstract
The double inverted pendulum is a system that can be used efficiently to test and validate the performance of control algorithms because of its instability and the many degrees of freedom system. The paper presents the design and development of a unique mobile double inverted pendulum system. LQR is applied for the balancing control of both pendulums and position control of the mobile platform of the system. Both simulations and experiments are conducted to evaluate the system performance.

1. Introduction
The inverted pendulum is a system that can be used to evaluate the control performance of control algorithms. This experimental apparatus has been brought into the tests since the mid-1960s [1]. There are several versions of this system such as double inverted pendulum, self-balancing robot, and so forth. The double inverted pendulum is unstable with many degrees of freedom system whereas the self-balancing robot can be considered as a mobile inverted pendulum.

There have been many controllers used to control the inverted pendulum systems which can be categorized as linear and non-linear controllers. The single inverted pendulum is simple. Linear control algorithms such as the Proportional-Integral-Derivative (PID) control and the state-space control have been successfully applied for the single inverted pendulum system [1]. Neural network (NN) [2] and the nonlinear controllers such as, an energy-based control technique [3] have been applied to control the system. A double inverted pendulum system is more complicated and has many degrees of freedom. Both linear controllers [4] and NN [5] have been applied to control the double inverted pendulum systems. The swinging up of this system was successfully conducted using an energy-based controller [6] and a sliding mode controller [7]. In addition to the inverted pendulum systems, self-balancing robots have been developed and applied to evaluate the performance of control algorithms as shown in [8]. Several control algorithms; such as PID, Linear Quadratic Regulator (LQR) [9] and Fuzzy Logic Controllers (FLC) [10] have been used to determine the self-balancing robots. Their control performances have been compared with the self-balancing robots as shown in [5].

There have been many controllers used to influence the inverted pendulum systems which can be categorized as linear and non-linear controllers. The single inverted pendulum is simple. Linear control algorithms such as the Proportional-Integral-Derivative (PID) control and the state-space control have been successfully applied for the single inverted pendulum system [1]. Neural network (NN) [2] and nonlinear controllers such as an energy-based control technique [3] have been applied to control the system. A double inverted pendulum system is more complicated and has many degrees of...
freedom. Both linear controllers [4] and NN [5] have been applied to control the double inverted pendulum systems. The swinging up of this system was successfully conducted using an energy-based controller [6] and a sliding mode controller [7]. In addition to the inverted pendulum systems, self-balancing robots have been developed and applied to evaluate the performance of control algorithms as shown in [8]. Several control algorithms such as PID, Linear Quadratic Regulator (LQR) [9] and Fuzzy Logic Controllers (FLC) [10] have been used to control the self-balancing robots. Their control performances have been compared with the self-balancing robots as shown in [5].

2. System Design

Figure 1a shows the mobile double inverted pendulum system developed. The frame of the self-balancing robot is fabricated from black acrylic. The wheels are mounted on the bottom plate of the frame by means of two rotary bearings (with friction coefficient less than 0.02 each) and a steel shaft. A single 28.9W DC motor is used to drive the system which is coupled to the wheels using two spur gears of similar diameter. Only the forward and backward motions are tested, thus, only a single actuator is sufficient for the system to drive. An incremental encoder with a resolution 0.18° is coupled with the rear end of the DC motor. An IMU unit with the output frequency of 100Hz is mounted on the body (frame) of the self-balancing robot to measure the leaning angle of the robot. The pendulum is fabricated using an aluminum rod with a rubber ball attached to the top of the rod. The bottom end of the pendulum is mounted on the self-balancing robot by means of a rotary bearing (with friction coefficient of less than 0.02) and an incremental encoder. This encoder is used to measure the angle between the pendulum and the body of the self-balancing robot. The motor is driven using a motor driver with a 12-bit resolution and 20kHz PWM signal. An Arduino due microcontroller is used to control the whole system. A 4-cell 2200mAh lithium-polymer battery is used to power the whole robot.

3. System identification
The schematic of the system used to derive the dynamics model is shown in Figure 1b. The subscripts $W$, $B$, and $P$ denote the wheels, body of the balancing robot and the pendulum, and $\alpha$, $\beta$, and $\gamma$ represent the rotation/leaning angles, respectively. Note that $\emptyset$ is the angle formed between the vertical and the radial axis in addition to $\emptyset$ when the body of the robot is tilted (This is due to the offset
between the radial axis and the point of action of CG of the robot body, \( X_{CG,B} \). Notations \( m, J, L, l, \) and \( r \) stands for the mass, inertia, total length, length to the CG from the axial axis and radius of the wheel, respectively. The derivation is initiated using the Euler-Lagrange equation, whereas the negligibly small friction components at bearings are disused in the calculation. Initially, the kinetic obtained as expressed in Equations (1) and (2).

\[
T_W = \frac{1}{2} m_W \dot{x}^2 + \frac{1}{2} J_W \dot{\gamma}^2 \tag{1}
\]

\[
T_B = \frac{1}{2} m_B ((\dot{x} + l_B \dot{\alpha}_a)^2 + (-l_B \dot{\alpha}_s)^2) + \frac{1}{2} J_B \ddot{\alpha}^2
\]

\[
T_P = \frac{1}{2} m_P ((\dot{x} + L_B \dot{\alpha}_a + l_B \beta \dot{c}_\beta)^2 + (L_B \ddot{\alpha}_s + l_B \beta \ddot{c}_\beta)^2) + \frac{1}{2} J_P \dot{\beta}^2
\]

Where \( l_B = \sqrt{X_{CG,B}^2 + Y_{CG,B}^2} \) and \( c_\beta = \frac{Y_{CG,B}}{l_B} \).

\[
V_W = 0, V_B = m_B g l_B \dot{c}_a, V_P = m_P g (L_B \dot{c}_a + l_P \dot{c}_\beta)
\]

(2)

Afterwards, the total potential energy is deducted from the total kinetic energy to form the Lagrangian. Consequently, the Euler-Lagrange equation outputs the nonlinear dynamics model of the system which is expressed in Equation (3).

\[
a_1 \ddot{x} + (a_2 \dot{c}_a + a_3 \dot{c}_{a+\phi}) \ddot{\alpha} + a_4 \ddot{c}_\beta \ddot{\beta} - (a_2 \dot{s}_a + a_3 \dot{s}_{a+\phi}) \dot{\alpha}^2 - a_4 \dot{c}_\beta \dot{\beta} = \frac{\tau}{r}
\]

\[
(a_2 \dot{c}_a + a_3 \dot{c}_{a+\phi}) \dot{x} + a_5 \ddot{\alpha} + a_6 \dot{c}_{a+\phi} \dot{\beta} + a_6 s_{a+\phi} \dot{\beta}^2 - (a_2 \dot{s}_a + a_3 \dot{s}_{a+\phi}) g = 0
\]

\[
a_4 \dot{c}_\beta \ddot{x} + a_6 s_{a+\phi} - \dot{\beta} + a_7 \dot{\beta} - a_6 s_{a+\phi} - \dot{\beta}^2 - a_4 \dot{c}_\beta g = 0
\]

Where,

\[
\begin{align*}
a_1 &= m_W + m_B + m_P + (J_W / r^2) & a_5 &= m_B l_B^2 + m_P l_B^2 + J_B \\
a_2 &= m_B l_B & a_6 &= m_P l_B l_P \\
a_3 &= m_P L_B & a_7 &= m_P l_B^2 + J_P \\
a_4 &= m_P l_P & \tau / r & \text{is the input force in the direction of } x
\end{align*}
\]

Subsequently, the nonlinear model is linearized by assuming that \( q \approx 0, \sin(q) \approx 0, \cos(q) \approx 0 \) and \( \ddot{q}^2 \approx 0 \). The DC motor dynamics are rearranged neglecting the armature inductance and friction in the motor shaft to obtain the form is expressed in Equation (4).

\[
\tau = \frac{K_T}{R} (V - K_e \frac{\dot{x}}{r})
\]

(4)

Finally, motor input torque in equation (4) is substituted in equation (3) to obtain the linearized model that is shown in Equation (5).

\[
\begin{bmatrix}
a_1 & a_2 + a_3 c_\beta & a_4 \\
a_2 + a_3 c_\beta & a_5 & a_6 c_\beta \\
a_4 & a_6 c_\beta & a_7
\end{bmatrix}
\begin{bmatrix}
\ddot{x} \\
\ddot{\alpha} \\
\ddot{\beta}
\end{bmatrix}
\begin{bmatrix}
a_6 \dot{x} \\
\dot{\alpha} \\
\dot{\beta}
\end{bmatrix}
+ \begin{bmatrix}
-x \\
-(a_2 + a_3) g \\
-a_4 g \beta
\end{bmatrix}
= \begin{bmatrix}
b_1 V \\
0 \\
0
\end{bmatrix}
\]

(5)

Where,

\[
a_6 = K_e K_T / R r^2 \quad b_1 = K_T / R r
\]

The parameters of the experimental system are measured and calculated as shown in Table 1. The state-space model of the system is formed as expressed in Equation (6) by substituting parameters to Equation (5).
Table 1. Parameter Identification

| Coefficient | Description | Coefficient | Description | Coefficient | Description |
|-------------|-------------|-------------|-------------|-------------|-------------|
| $m_W$       | 150g        | $I_B$       | 100mm      | $\emptyset$ | 0.859°      |
| $J_W$       | 0.000Nm     | $L_B$       | 480mm      | $L_p$       | 420mm       |
| $r$         | 35mm        | $m_p$       | 150g       | $K_T$       | 0.155Nm/A   |
| $m_B$       | 2200g       | $J_p$       | 0.016Nm    | $K_e$       | 0.327V/rads$^1$ |
| $J_B$       | 0.022Nm     | $l_p$       | 330mm      | $R$         | 2.5Ω        |

Since $\emptyset$ is negligibly small, the offset in CG of the body in the tangential axis is approximated to be zero in the derivation of the linearized model.

$$X = [x \ \alpha \ \beta \ \dot{x} \ \dot{\alpha} \ \dot{\beta}]^T, \quad A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -8.151 & 0.504 & -8.170 & 0 & 0 \\ 0 & 80.782 & -7.864 & 34.127 & 0 & 0 \\ 0 & -46.395 & 19.819 & -12.438 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1207 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1.837 \end{bmatrix} \quad (6)$$

Matlab is used to identify the open-loop poles of the system, which deduce that the system is unstable. In addition, the controllability and observability confirm that the system is both controllable and observable.

4. Control Architecture

4.1. Input signal filtering for experimental system

The Signal from the gyro has noise in the angular velocity reading. Hence, a simple moving average filter is applied as expressed in the first line of the Equation (7). In this case, $\dot{a}_n$ represents original signal and $\dot{a}_T$ represents filtered output. Likewise, the differentiated output from the encoder is conditioned with a simple low pass filter with the gain $(K_{ip})$ of 1 and cut-off frequency $(T_{ip})$ at 10Hz. The second line of Equation (7) shows the filter transfer function for the two encoders connected to the motor and the pendulum. In this case, $\theta$ represents the original signal and $\theta_{filtered}$ represents the filtered output.

$$\dot{\alpha}_T = \sum_{n=T-29}^{T} \frac{\dot{a}_n}{50} \quad (7)$$

4.2. LQR

LQR is used to control the robot position, the leaning angle of the body of the self-balancing robot, and the pendulum at the upright position. LQR is an optimal controller that minimizes the cost function expressed in equation (8). The state depending matrix $Q$ and input weighting matrix $R$ have to be adjusted according to the desired cost function. The control law of the LQR follows $u = -Kx(t)$ where $K$ is the optimal gain matrix. $K$ can be obtained from the Algebraic Ricatti Equation (ARE).

$$J = \int_{T}^{x'}(t)Qx(t) + u'(t)Ru(t) \ dt \quad (8)$$

4.3. Position control of the system

Figure 2 shows the complete control block diagram of the system. The input $u$ that is obtained using the control law is fed to the system to obtain the outputs $x$, $\alpha$, and $\beta$. $\alpha$ and $\beta$ are used to estimate $\dot{\alpha}$ and $\dot{\beta}$ while $\dot{x}$ is also estimated. The estimations are followed up by a series of filters.
The external step input applied to change the position of the robot \( x_{set} \) is converted to a linear input using a linear trajectory generator (LTG), summed with the \( x \), and the six states are inputted to obtain the control law. This loop is repeated at 250Hz to maintain the stability and the position of the system.

![Control block diagram](image)

**Figure 2.** Control block diagram

### 5. Results

#### 5.1 Simulation results

Simulation is conducted on Matlab/Simulink to evaluate the performance of the controller. \( Q \) and \( R \) matrices are defined to achieve the desired control conditions. \( Q \) matrix is defined in such a way that higher weight is given on the position states rather than the velocity states. The input weighting matrix is defined as such that the input does not exceed the limitation of the driving DC motor. Firstly, the gains for balancing the robot and the pendulum without robot position control are obtained by defining lower weight for the wheel position (Case I). Then, \( Q \) matrix is defined as \( \text{diag}(10000, 5000, 5000, 5, 5, 5) \) and \( R \) as \([1, 0] \) which focuses on both stabilizing and position maintaining.

The simulations are conducted with the initial condition at \((0, 4, 1, 0, 0, 0)\) (in degrees and degrees per second) and the optimal gain obtained as \((100.00, -525.87, 527.09, 99.05, 64.24, 133.09)\). Figure 3 shows the simulation results (in generalized coordinates) for Case I, Case II (the robot position is controlled at zero position), and Case III (the robot position is controlled by altering the target position with a 12cm distance). Figure 3a shows that the self-balancing robot and the pendulum reach the upright position within 3s, however, the robot position does not stay at zero position. In Figure 3 b,
the position is controlled at zero position within 2.4\(s\) while the pendulum takes 2.7\(s\) to be balanced at the upright position. The result of tracking a step position command of 12\(cm\) distance is shown in Figure 3c. This results in the system reaching the new position within 3.5\(s\).

5.2 Experimental results

Figures 4a and 4c show the signal after the low pass filter from the two encoders. It is observed that the differentiation of the motor encoder contains the noise of 750°\(s^{-1}\) the amplitude at 3Hz. This noise is removed by a low pass filter. However, this filtering causes a delay of 16\(ms\) to the original signal. A similar result is observed from the pendulum encoder signal with a delay of 14\(ms\) to the original signal. The averaging filter acting on the gyro signal removes the amplitude noise and causes an 8\(ms\) delay to the signal, as depicted in figure 4b, the plot in grey contained the actual signals whereas the colored plots represent the filtered signals.

Next, three experiments are conducted. The first experiment is conducted to balance the robot and the pendulum without the robot positioning control (Case IV). The weight matrix is defined so that higher priority is given for the leaning angle of the self-balancing robot and the pendulum, than the robot position. Then, optimal gain matrix is obtained as \((105.0, -525.0, 540.0, 60.9, 60.4, 39.8)\), which corresponds to both balancing and robot positioning control. Thereafter, all the positions are controlled at zero (Case V). In the final experiment, both leaning angles of the robot and the pendulum are controlled at zero and the robot position is controlled at different set positions (Case VI).

Figure 5 shows the experimental results (in generalized coordinates) from the three cases. Figure 5a shows the result of Case IV. It can be seen that the robot position does not stay at zero position while the body and the pendulum leaning angles are controlled near zero degrees. However, high-frequency noises can be observed from both measurement signals. The body leaning angle has a standard deviation of 4.58° whereas the pendulum shows a standard deviation of 2.29° from the upright
position. It also can be seen that when the robot position is not controlled, the robot continuously moves to balance the body and the pendulum. In this experiment, oscillation is less compared with the other two cases. Figure 5b shows the result from Case V. It can be observed that the robot position is controlled at the starting position (zero position) with oscillation. The oscillation has a standard deviation of 0.84 cm while the mean is 0.14 cm away from the desired position. The standard deviations of the leaning angle of the self-balancing robot and the pendulum are reduced to 2.28° and 0.71°, respectively. The result of a single position change from Case VI is shown in Figure 5c. The offset position is applied as shown in the dotted lines, whereas the actual desired position is shown in a solid line. The position is changed for 30 cm at the speed of 10 cm s⁻¹. The robot position oscillates when the offset position is changed linearly. It can be seen that the robot moves to the new position at the end of the ramp, however, it still oscillates. The oscillation has a standard deviation of 0.74 cm with a shift of 0.78 cm as observed from the 30 cm mark. The result from the step change of 75 cm shows a similar result. The oscillation is seen after reaching the new position with a standard deviation of 2.05 cm and a shift of 3.12 cm from the desired position.

5.3 Comparison between the Two Test Methods
Firstly, the gain matrices from simulations and experiments show a similarity in each gain except for those corresponding to wheel and pendulum angular velocities. This is due to the undiminished noises at low amplitudes from the low pass filters. This phenomenon results in constraining the two individual gains to a lower value than those from the simulations. The minor differences are mainly caused due to the negligibly small friction parameters uncompensated during simulations. The result of the analysis conducted to obtain the maximum recovery angles shows that the angles recovered by the body of the balancing robot are comparatively higher in simulations, where the recovery angles of the pendulum are higher in experiments.

6. Conclusion
This paper described the development and control of a mobile double inverted pendulum. Both simulations and experiments were conducted to evaluate the performance of the system in balancing and position control. The results showed successful balancing of body and pendulum at the upright position and the position control at the desired positions, using LQR.

References
[1] Phelps F and Hunter J 1965 American Journal of Physics 33 285-295
[2] Williams V and Matsuko K 1991 Proc. 1991 IEEE International Joint Conference on Neural Networks (Singapore) 18-21
[3] Maeba T Deng M Yanou A and Henmi T 2010 Proc. 2010 International Conference on Modelling, Identification and Control (Okayania)
[4] Cheng F Zhong G Li Y and Xu Z 1996 Fuzzy Sets and Systems 79 315-321
[5] Bogdanav A 2004 Oreg. Heal. Scie. Univ CSE-04 006
[6] Yamakita M Iwashiro M Sugahara Y and Futura K 2002 Proc.1995 American Control Conference (Seattle)
[7] Henmi T Deng M and Inoue A 2004 Proc. 2004 American Control Conference (Boston)
[8] Chee O Y Shukri M and Abidin Z 2006 Proc. 2006 4th Student Conference on Research and Development, IEEE 2007 (Selangor)
[9] Xu C Li M and Pan F 2011 Proc. 2011 International Conference on Electrical and Control Engineering, IEEE 2011 (Yichang)
[10] Bature A A Buyamin S Ahmed M N and Muhammad M 2014 International Journal of Mechanical Mechatronics Engineering 14(3) 62-68