

LQG Control Design for Balancing an Inverted Pendulum Mobile Robot

R. EIDE, P.M. EGELID, A. STAMSØ AND H.R. KARIMI

Department of Engineering, Faculty of Engineering and Science, University of Agder,
Grimstad, Norway (e-mail: hamid.r.karimi@uia.no)

May 16, 2011

Abstract

The objective of this paper is to design linear quadratic controllers for a system with an inverted pendulum on a mobile robot. To this goal, it has to be determined which control strategy delivers better performance with respect to pendulum's angle and the robot's position. The inverted pendulum represents a challenging control problem, since it continually moves toward an uncontrolled state. Simulation study has been done in MATLAB Simulink environment shows that both LQR and LQG are capable to control this system successfully. The result shows, however, that LQR produced better response compared to a LQG strategy.

1 Introduction

Inverted pendulum has been the subject of numerous studies in automatic control since the forties. The inverted pendulum is a typical representative of a class of high-order nonlinear and non-minimum phase systems [1]. Since the system is inherently nonlinear, it is useful to illustrate some of the ideas in nonlinear control.

Wheeled mobile robots have in the recent years become increasingly important in industry, since they provide a large degree of flexibility and efficiency with respect to transportation, inspection, and operation. In many cases, however, it is the control, or lack of knowledge in control, which limit the area of application. In this system, an inverted pendulum is attached to a robot equipped with a motor that drives it along a horizontal track (*robot* will in the following mean an autonomous robot provided with the *Robotics Starter Kit* from National Instruments). The user is able to dictate the position and velocity of the robot through the motor. The pendulum is characterized by an unstable equilibrium point, and its behavior can be used in the analysis and stability control of many similar systems.

Many different control methods are proposed for the inverted pendulum problem. The Proportional-Integral-Derivative (PID) and Proportional-Derivative (PD) controllers [2] and [3], Model Predictive Control (MPC) [4], and fuzzy control [5] to mention a few. However one of the obstacles by using the PID and PD controllers are that they alone cannot effectively control all of the pendulum state variables (modes) since they are of lower order than the pendulum itself. Hence, they are usually replaced by a full-order controller [3]. A linear state feedback controller based on the linearized inverted pendulum model can instead be used, and may also be extended with a disturbance observer (Kalman filter) to improve the disturbance rejection performance.

The proposed method is to balance an inverted pendulum placed on top of a mobile robot by use of LQR/LQG control methods. Our solution implements an LQG controller with comparison to a simple LQR controller. The controller found by means of a more analytical approach will be tested with implementation of the controller in the MATLAB/Simulink environment.

With varying input forces the goal is to design

a controller capable of meeting the following requirements

- settling time, T_s , less than 5 seconds
- maximum overshoot of 10 degrees (0.175 radians)
- rise time T_r less than 0.5 s

We will in part one describe a simplified mathematical model of the problem. Then, when having a state space model of the system, we can in part two go on and determine the controllers and observers. Finally, the results from simulations will then be presented and compared in part three.

2 Derivation of the Mathematical Model

The main objective in this part is to come up with a valid model that can be used as a basis for the control design. The overall system can be described as Figure 1 depicts. The mass of the robot and the pendulum is here denoted as M and m , respectively, together with the applied force F and the angle θ referred to the vertical axis.

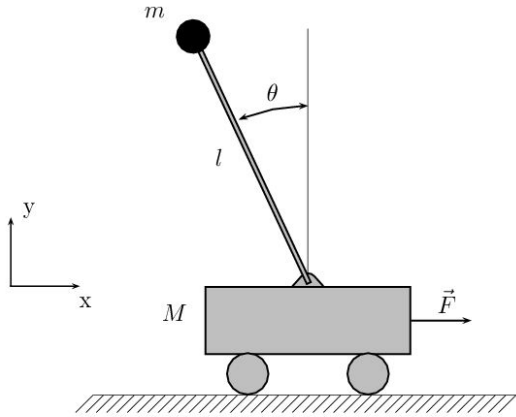


Figure 1: *The inverted pendulum placed on a simplified mobile robot*

There are a lot of aspects to take into consideration when modeling this system, of which some are more crucial than others. Without losing the generality of the model, the authors have considered the following assumptions as good

approximations for the sake of simplifying the model:

- no friction in the hinge between the pendulum and the robot
- no friction between the wheels and the horizontal plane
- small angle approximations, i.e the pendulum does not move more than a few degrees
- sensors for measuring all the states are when considering LQR control assumed to be available

By applying the Lagrange's equations with respect to the position of the robot, x , and the pendulum deflection angle θ , and taking into account the moments around the center of mass, the following nonlinear dynamic equations of the pendulum system are obtained [6]:

$$(I + ml^2)\ddot{\theta} - mgl\sin\theta = -ml\ddot{x}\cos\theta \quad (1)$$

where I is moment of inertia of the pendulum, θ is the angle in counterclockwise direction with respect to the vertical line (see figure 1), and l is the distance to the center of mass (equal to $L/2$). Furthermore, the robot is governed by an equation of motion relating the forces applied by the pendulum on the robot. Then, according to Newton's law of motion, the equation of motion of the robot in horizontal direction can be given as

$$(M + m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \quad (2)$$

where F can be derived out from the physical properties of the motor to be [7]

$$F = \frac{\eta_t K_m K_t}{Rr} V - \frac{K_m^2 K_g^2}{Rr^2} \dot{x} \quad (3)$$

Here K_m , K_t , η_t , K_g , R , and r are coefficients dependent on the physical properties of the motor and gear. The different parameters for (3) is not known for the robot used in this project. However, [8] provide values which will be used in the following. Based on the previous derivation, we can now define four state variables, given as the *state vector*

$$\vec{x} = [x_1 \ x_2 \ x_3 \ x_4]^T = [x \ \dot{x} \ \theta \ \dot{\theta}]^T \quad (4)$$

Equations (1), (2), and (3) need to be linearized in order to represent the system on state space form

$$\dot{x} = Ax + BV \quad (5)$$

$$y = Cx \quad (6)$$

where A , B , and C are matrices needed for the control design and V is the (input) voltage applied on the motor. The linearizing point of interest is the unstable equilibrium point

$$\vec{x} = [0 \ 0 \ 0 \ 0]^T$$

The assumption of small angle approximation gives $\sin\theta = \theta$, $\cos\theta = 1$, and $\dot{\theta} = 0$. This lead to the following linearized equations of motion (note that (3) will be the same)

$$\ddot{\theta} = \frac{(mgl\theta - ml\ddot{x})}{(I + ml^2)} \quad (7)$$

$$\ddot{x} = \frac{(F - ml\ddot{\theta})}{(M + m)} \quad (8)$$

We want to make (7) and (8) more convenient for the state space representation by expressing $\ddot{\theta}$ and \ddot{x} in the following way

$$\ddot{\theta} = \frac{K_m^2 K_g^2}{Rr^2(M_t L - ml)} \dot{x} + \frac{g}{L - ml} \theta - \frac{K_m K_g}{Rr(M_t L - ml)} V \quad (9)$$

$$\ddot{x} = -\frac{LK_m^2 K_g^2}{Rr^2(LM_t - ml)} \dot{x} - \frac{mlg}{(LM_t - ml)} \theta + \frac{LK_m K_g}{Rr(LM_t - ml)} V \quad (10)$$

where for simplicity $L = \frac{I+ml^2}{ml}$ and $M_t = M + m$

Based on these equations and the following relation

$$\vec{x} = [x_2 \ \ddot{x} \ x_4 \ \dot{\theta}]^T$$

we can now represent (9) and (10) as a first order system. The system on state space form then be

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{K_m^2 K_g^2}{Rr^2(M_t L - ml)} & \frac{g}{L - ml} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{LK_m^2 K_g^2}{Rr^2(LM_t - ml)} & -\frac{mlg}{(LM_t - ml)} & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ \frac{LK_m K_g}{Rr(LM_t - ml)} \\ 0 \\ -\frac{K_m K_g}{Rr(M_t L - ml)} \end{bmatrix} V \quad (11)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \vec{x} \quad (12)$$

Now, by using the values for the different constants given in [8], the matrix equations becomes

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -15.14 & -3.04 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 37.23 & 31.61 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 3.39 \\ 0 \\ -8.33 \end{bmatrix} \vec{V}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \vec{x}$$

When having the system on state space form, the following step is to design a LQR controller (assuming that all the states are known and can be measured). Then, by assuming no sensor available for measuring the angle, we will continue with an observer design. Finally, an optimal LQR (LQG controller) will be proposed by implementing a Kalman filter.

3 Linear Quadratic Regulator (LQR) design

There are different methods, or procedures, to control the inverted pendulum. One is the pole placement procedure having the advantage of giving a much clearer linkage between adjusted parameters and the resulting changes in controller behavior. However, one disadvantage with this method is that the placing of the poles at desired locations can lead to high gains [8]. In this section a linear quadratic regulator (LQR) is proposed as a solution. The principles of a

LQR controller is given in Figure 2. Here is the state space system represented with its matrices \mathbf{A} , \mathbf{B} , and \mathbf{C} with the LQR controller (shown with the $-\mathbf{K}$).

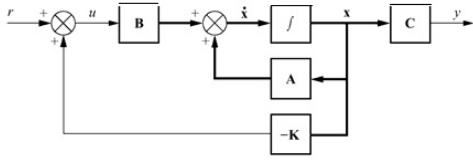


Figure 2: Schematic representation of state space control using a LQR controller

The LQR problem rests upon the following three assumptions [9]:

1. All the states ($\mathbf{x}(t)$) are available for feedback, i.e it can be measured by sensors etc.
2. The system are *stabilizable* which means that all of its unstable modes are controllable
3. The system are *detectable* having all its unstable modes observable

To check whether the system is controllable and observable, we use the functions `obsv(A,C)` and `ctrb(A,B)` and find this to be true.

LQR design is a part of what in the control area is called *optimal control*. This regulator provides an *optimal control law* for a linear system with quadratic performance index yielding a cost function on the form [10]

$$J = \int_0^{\infty} \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t) dt \quad (13)$$

where $\mathbf{Q} = \mathbf{Q}^T$ and $\mathbf{R} = \mathbf{R}^T$ are weighting parameters that penalize the states and the control effort, respectively. These matrices are therefore controller tuning parameters. It is crucial that \mathbf{Q} must be chosen in accordance to the emphasize we want to give the response of certain states, or in other word; how we will *penalize* the states. Likewise, the chosen value(s) of \mathbf{R} will penalize the control effort u . Hence, in an optimal control problem the control system seeks to maximize the return from the system with *minimum cost*. In a LQR design, because of the quadratic performance index of the cost function, the system has a mathematical solution that yields an

optimal control law

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \quad (14)$$

where \mathbf{u} is the control input and K is the gain given as $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S}$. It can be shown (see [10]) that S can be found by solving the algebraic Riccati Equation

$$\mathbf{S}\mathbf{A} + \mathbf{A}^T\mathbf{S} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} = \mathbf{0} \quad (15)$$

The process of minimizing the cost function therefore involves to solve this equation, which will be done with the use of MATLAB function `lqr`. In this project the parameters in \mathbf{Q} was initially chosen according to *Bryson's Rule* (see [11] for details) to be

$$\mathbf{Q} = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 32.65 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

and the control weight of the performance index \mathbf{R} was set to 1.

Here we can see that the chosen values in \mathbf{Q} result in a relatively large penalty in the states x_1 and x_3 . This means that if x_1 or x_3 is large, the large values in \mathbf{Q} will amplify the effect of x_1 and x_3 in the optimization problem. Since the optimization problem are to minimize J , the optimal control \mathbf{u} must force the states x_1 and x_3 to be small (which make sense physically since x_1 and x_3 represent the position of the robot and the angle of the pendulum, respectively). This values must be modified during subsequent iterations to achieve as good response as possible (refer to the next section for results). On the other hand, the small \mathbf{R} relative to the max values in \mathbf{Q} involves very low penalty on the control effort \mathbf{u} in the minimization of J , and the optimal control \mathbf{u} can then be large. For this small \mathbf{R} , the gain \mathbf{K} can then be large resulting in a faster response. In the physical world this might involve instability problems, especially in systems with saturation [8].

After having specified the initial weighting factors, one important task is then to simulate to check if the results correspond with the specified design goals given in the introduction. If not, an

adjustment of the weighting factors to get a controller more in line with the specified design goals must be performed. However, difficulty in finding the right weighting factors limits the application of the LQR based controller synthesis [?]. By an iterative study when changing Q values and running the command $K=lqr(A,B,Q,R)$ $K=$ $\begin{bmatrix} -10.0000 & -12.6123 & -25.8648 & -6.7915 \end{bmatrix}$

The simulation of time response with this controller will be shown in the next section.

3.1 State Estimation

As mentioned for the case of the LQR controller, all sensors for measuring the different states are assumed to be available. This is not a valid assumption in practice. A void of sensors means that all states (full-order state observers), or some of the states (reduced order observer), are not immediately available for use in any control schemes beyond just stabilization. Thus, an observer is relied upon to supply accurate estimations of the states at all robot-pendulum positions. The schematics of the system with the observer is shown in figure 3 below.

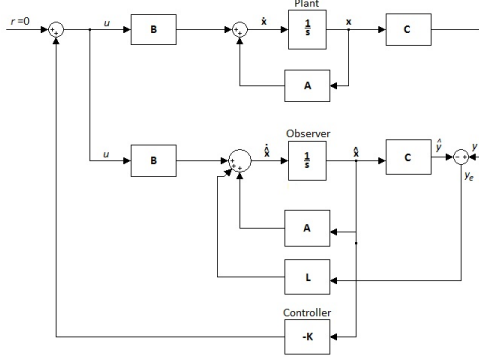


Figure 3: Schematic of state space control using an observer where L is the observer gain and K is the LQR gain matrix

As can be seen from figure 3, the observer state equations are given by

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \quad (17)$$

$$\hat{y} = C\hat{x} \quad (18)$$

where \hat{x} is the estimate of the actual state x . Furthermore, equations (17) and (18) can be re-

written to become

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly \quad (19)$$

This, in turn, is the governing equations for a *full order observer*, having two inputs u and y and one output, \hat{x} . Since we already know A , B and u , observers of this kind is simple in design and provides accurate estimation of all the states around the linearized point. From Figure 3 we can see that the observer is implemented by using a duplicate of the linearized system dynamics and adding in a correction term that is simply a gain on the error in the estimates. Thus, we will feed back the difference between the measured and the estimated outputs and correct the model continuously. The proportional observer gain matrix, L , can be found by pole placement method by use of the `place` command in MATLAB. The poles were determined to be ten times faster than the system poles. These were found to be $eig(A - B * K) = [-18.0542, -5.4482, -3.4398, -2.4187]^T$, which yields the gain matrix

$$L = \begin{bmatrix} 34.4 & 0.2 \\ 91.0 & 1.1 \\ 30.4 & 52.5 \\ 1103.6 & 726.4 \end{bmatrix} \quad (20)$$

When combining the control-law design with the estimator design we can get the compensator (see Section 6 for results).

4 Kalman filter design

In the previous design of the state observer, the measurements $y = Cx$ were assumed to be noise free. This is not usually the case in practical life. Other unknown inputs yielding the state equations to be on the general *stochastic state space form*

$$\begin{aligned} \dot{x} &= Ax + Bu + Gd \\ y &= Cx + Hd + n \end{aligned} \quad (21)$$

where the matrices G can be set to an identity matrix, H can be set to zero, d is stationary, zero mean Gaussian white process noise, and n is sensor noise of the same kind.

In this section we will show how the Kalman filter can be applied to estimate the state vector, \hat{x} and the output vector \hat{y} by using the known inputs u and the measurements y . Block schematics of a Kalman filter connected to the plant is depicted in Figure 4.

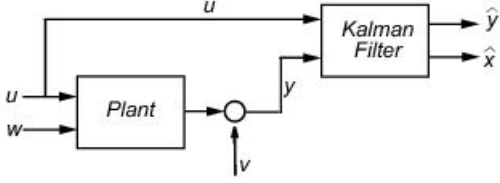


Figure 4: Kalman filter used as an optimal observer

It can be shown that this will result in an optimal state estimation. The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance when some presumed conditions are met [12]. The mean square estimation error is given by

$$J = E[(x(t) - \hat{x}(t))^T(x(t) - \hat{x}(t))] \quad (22)$$

where

$$E[(x(t) - \hat{x}(t))^T y(t)] = 0 \quad (23)$$

The optimal Kalman gain is given by

$$L(t) = S_e(t)C^T R^{-1} \quad (24)$$

where $S_e(t)$ is the same as J given in (22). Furthermore, when $t \rightarrow \infty$, the algebraic Riccati equation can be written as $\dot{S}_e(t) = 0$

$$0 = S_e A^T + A S_e + Q_n - S_e C^T R_n^{-1} C S_e \quad (25)$$

where Q_n and R_n are the process and measurement noise covariances, respectively. Tuning of the Kalman filter are required if these are not known. Note that when controlling the robot, a quadrature encoder will be used for measurement of the position. Since this device does not involve any noise, we can in the following set $R_n = 0$. Finally, the sub-optimal Kalman gain for a steady state Kalman filter can be expressed as $L = S_e C^T R^{-1}$.

5 LQG controller

The LQG controller is simply the combination of a Kalman filter with a regular LQR controller. The separation principle guarantees that these can be designed and computed independently [13]. LQG controllers can be used both in linear time-invariant systems as well as in linear time-variant systems. The application to linear time-variant systems enables the design of linear feedback controllers for non-linear uncertain systems, which is the case for the pendulum-robot system. The schematics a LQG is in essence similar to that depicted in figure 3 in Section 3.1. Hence, the observer gain matrix L in this figure can now be defined as the *Kalman gain*. However, we also assume disturbances in form of noise, such that the system in compressed form can be described as in Figure 5

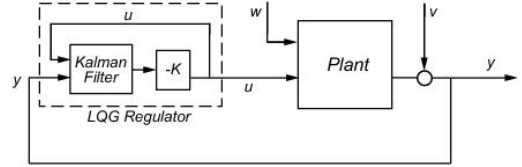


Figure 5: LQG regulator

From the system given by (21), the feedback control law given by (14), and the full state observer equation given by (19) defined previously, we can by combing these have the following output-feedback controller to be

$$\begin{aligned} \dot{\hat{x}} &= (A - LC - BK)\hat{x} + Ly \\ u &= -K\hat{x} \end{aligned} \quad (26)$$

which is the equations the software is based upon when calculating the state estimates. Since the optimal LQR controller for this system is already found to have the feedback gain matrix $K = [-10.0000 \quad -12.6123 \quad -25.8648 \quad -6.7915]$ this will be used directly in the LQG design. The remaining is to find the Kalman gain matrix. We first assume the system as given in (21). After the measurement and disturbance noise covariance matrices are determined, the MATLAB function `kalman` was used to find the optimal Kalman gain L and the covariance matrix P . Measurement noise covariance matrix is

determined out from an expected noise on each channel. The Kalman gain was found as

$$L = \begin{bmatrix} 4.9544 & 5.0879 \\ -0.7797 & -0.8612 \\ 5.0879 & 5.4689 \\ 25.9957 & 27.8975 \end{bmatrix} \quad (27)$$

6 Simulations

Figure 6 gives the time response of the system with a step input simulated by use of `lsim` function in MATLAB.

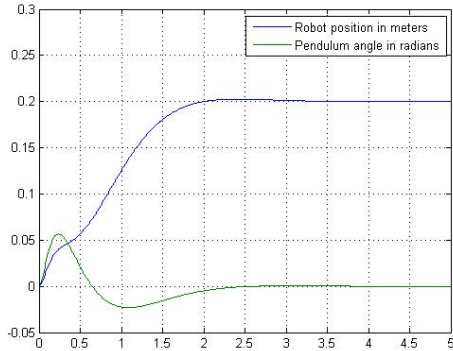


Figure 6: Time response of the system using a LQR controller

Note that a steady state error was reduced by a scaling factor `N` after the reference. This factor was calculated by a `rscale` function. Figure 8 below is the response when using an observer to estimate the states. Note that the simulation in this case was done in the Simulink environment with an impulse function as input.

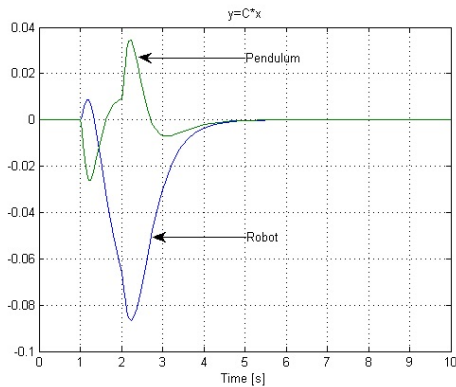


Figure 7: Time response of the system with LQR and observer

The reason for the difference between the plots is due to the different simulation methods. When simulating the LQG, we made it be that only the pendulum was subjected to an impulse force. This resulted in the following plot

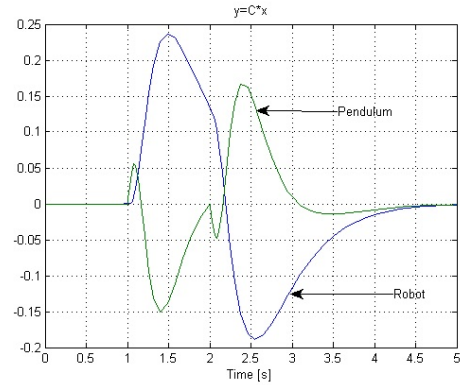


Figure 8: Time response of the system with LQG and impulse force on pendulum

We actually see here that the LQR gave better results.

Conclusion

This report has shown that by manipulating the state/control weightings and noise covariance matrices properly, both LQR and LQG will give satisfactory results. Although the results have been somewhat limited, it has provided us with some useful knowledge concerning differences between LQR and LQG control. It could have been interesting to see how other cost functions would have affected the results, and how the system would have reacted if some of the input values had been changed.

References

- [1] T. Žilic, D. Pavkovic, D. Zorc, Modeling and control of a pneumatically actuated inverted pendulum. *ISA Transactions*, Vol. 48, No. 3, July 2009, pp. 327-335.
- [2] J. Ngamwiwit, N. Komine, S. Nundrakwang, T. Benjanarasuth. Hybrid controller for swinging up inverted pendulum system. *Proceedings of the 44th IEEE conference on*

- decision and control, and the European control conference, 2005.*
- [3] J.F. Hauser, A. Saccon. On the driven inverted pendulum. *Proceedings of the 2005 fifth international conference on information, communications and signal processing, 2005.*
- [4] R. Balan and V. Maties. A Solution of the Inverse Pendulum on a Cart Problem Using Predictive Control. *Technical University of Cluj-Napoca, 2005.*
- [5] K. Tanaka, T. Ikeda, H.O. Wang, Fuzzy regulators and fuzzy observers : Relaxed stability conditions and lmi-based designs. *IEEE Transactions on Fuzzy Systems, Vol. 6, No.2, 1998, pp. 250-265.*
- [6] C. Xu and X. Yu. Mathematical modeling of elastic inverted pendulum control system. *Journal of Control Theory and Applications, Vol. 2, No. 3, 2004, pp. 281-282.*
- [7] J. Lam. Control of an Inverted Pendulum. *Georgia Tech College of Computing, 2009.*
- [8] A. E. Frazhol. The Control of an Inverted Pendulum. *Purdue University, School of Aeronautics and Astronautics, 2007.*
- [9] M. Athans. The Linear Quadratic LQR problem. *Massachusetts Institute of Technology, 1981.*
- [10] R. S. Burns. Advanced Control Engineering. *Butterworth Heinemann, 2001.*
- [11] J. Hespanha. Undergraduate Lecture Notes on LQG/LQR controller design. 2007.
- [12] G. Welch and G. Bishop. An Introduction to the Kalman Filter. *University of North Carolina, 2001.*
- [13] H. Morimoto. Adaptive LQG Regulator via the Separation Principle. *IEEE Transactions on Automatic Control, Vol. 35, No. 1, 1991, pp. 85-88.*