1st International Mediterranean Science and Engineering Congress (IMSEC 2016)
Çukurova University, Congress Center, October 26-28, 2016, Adana / TURKEY
Pages: 4916-4920, Paper ID:1417

Modeling and LQR control of a wheeled self-balancing robot

Abdullah Çakan[1], Fatih Mehmet Botsalı[2]

[1]Selcuk University, Turkey; acakan@selcuk.edu.tr
[2]Selcuk University, Turkey; fbotsali@selcuk.edu.tr

Abstract

In this paper, developing and control of a wheeled self-balancing robot is described. Model of the wheeled self-balancing robot is created by using a computer aided design software and exported to Matlab/Simmechanics. The model is linearized for deriving state (A) and input (B) matrices and derived matrices are used while designing the controller. LQR (Linear-quadratic regulator) controller is designed and implemented in order to use in body angle and wheel position control simulations by using Matlab/Simulink. The control results shows that the LQR controller can successfully achieve body angle and wheel position control of the wheeled self-balancing robot. Controlled position results are given in the form of the graphics.

Keywords: Modeling, Simulation, Control, LQR Controller, Balancing, Robot

1. INTRODUCTION

Wheeled self-balancing robots has the same basics with the inverted pendulum systems. And this system is a model commonly used in control system studies to compare control and optimization algorithms as a benchmark problem. It is unstable without control that is why the robot body is acting like a pendulum and easily fallen over if the wheel is not moved to balance it. The wheeled self-balancing robots is a nonlinear dynamic system aim of the control system is to balance the body as an inverted pendulum by applying the torque.

Self-balancing systems include multiple links, cart or wheels to be commanded and wheeled self-balancing robots have one or two wheels in literature [1-4]. There are different type control algorithms such as PID and LQR, neural network based fuzzy logic control used in both experimental and numerical studies [5-6]. Variety of optimization algorithms such as Genetic Algorithm, The Bees algorithm and PSO algorithm [7-9]. Technology of the Segway PT which is a self-balancing transportation device is an another example of wheeled self-balancing system that is basic of this paper [10-11].

This paper presents modeling, body angle control and wheel position control of a wheeled self-balancing robot. For this aim; a CAD software is used for modeling and Matlab software is used for both linearization and controller design. Thus, these types mechanical systems can be modeled and controlled without mathematical model. A LQR controller is designed in Matlab/Simulink for body angle and wheel position control and LQR controller Q and R matrices are determined by trial error method. Control simulation is realized and controller performance is given in the form of the graphics.

2. MODELING AND CONTROL

In this section modeling and control of the wheeled self-balancing robot is described. Model of the wheeled self-balancing robot is created by using a computer aided design software and exported to Matlab/Simmechanics that is multi-body modeling toolbox of the Matlab. After linearization, LQR controller designed and implemented to the model using Matlab/Simulink software. Simmechanics model of the system shown in Figure 1.

Modeling and control strategy of this paper is given in Figure 2. In this study mathematical model of the wheeled self-balancing robot is not used and all system dynamics are obtained by using Matlab/Simulink software. Also control simulations are realized with this way.

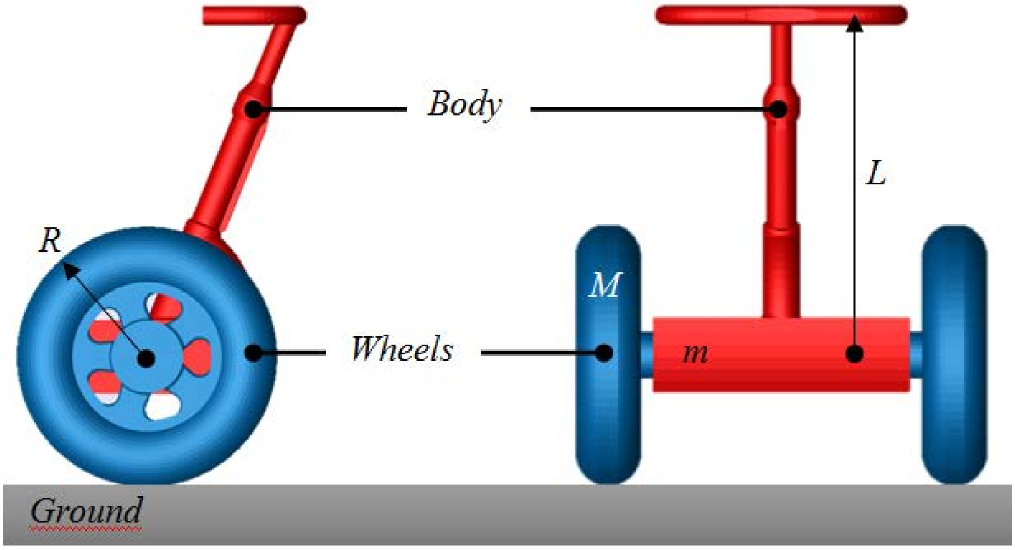

Figure 1 : Simmechanics model of the wheeled self-balancing robot

Figure 2 : Working strategy of the paper.

In the wheeled self-balancing robot modeling, the input of the system is torque τ that applied to the wheels and the outputs are body angle θ and position of the wheel x. The body and wheels are assumed to be rigid. The parameters of the wheeled self-balancing robot are given in Table 1.

Table 1 Wheeled self-balancing robot parameters

M	Mass of the wheels	10 kg
m	Mass of the body	40 kg
L	Body length	0.46 m
R	Wheel radius	0.18 m
τ	Torque applied to the wheel	Nm
x	Wheel position coordinate	m
θ	Pendulum angle	deg

Table 1 Wheeled self-balancing robot parametersIn this system control torque is used for rotational motion of the wheels, and the wheels are moving horizontally to the x direction on the ground to balance body angle. Control block diagram is shown in Figure 3. Wheel position and body angle are controlled using LQR controller

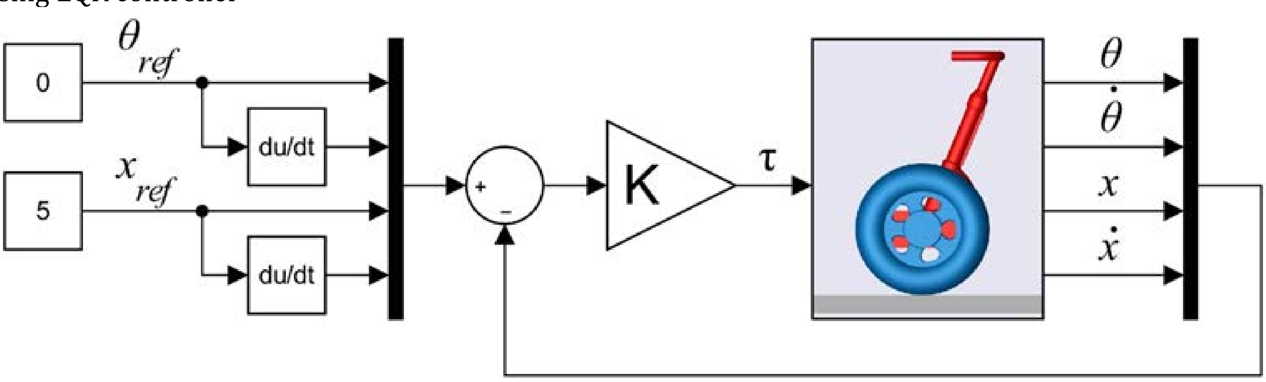

Figure 3 : Control block diagram

Linear Quadratic Regulator (LQR) is a control method that is appropriate to control wheeled self-balancing robot body angle and wheel position. Equations of motion of the system can be described in the form Eq. (1) in this equation A and B are the state and input matrices of the system, the LQR determines the control signal u to minimize of objective function J shown in Eq. (2).

$$\dot{x} = Ax + Bu \tag{1}$$

$$J = \int_0^\infty \left(x_{ref} - x(t)\right)^T Q \left(x_{ref} - x(t)\right) + u(t)^T R u(t)\, dt \tag{2}$$

The matrices Q and R remove the error between desired input and the control response. In this paper state vector x defined in the Eq. (3) and A and B matrices are generated by linearization of the wheeled self-balancing system using Matlab. Generated A and B matrices shown in Eqs. (4) - (5)

$$x = [\theta \quad \dot{\theta} \quad x \quad \dot{x}]^T \tag{3}$$

$$A = \begin{bmatrix} -38.7316 & 12.8794 & 5.6979 & 9.9588 \\ -104.7609 & 39.2818 & 18.0856 & 30.5764 \\ 83.2656 & -30.5537 & -13.2043 & -23.5357 \\ -44.2548 & 16.7696 & 6.3255 & 12.6540 \end{bmatrix} \tag{4}$$

$$B = \begin{bmatrix} -0.7188 \\ -2.0030 \\ 1.5985 \\ -0.8379 \end{bmatrix} \tag{5}$$

The next step in control design is to find the vector of state-feedback control gains K matrix. A, B, Q and R matrices are needed to find the K using Matlab with"lqr" command. Q and R matrices are determined by trial and error and shown in Eq. (6). The command is used in the form of Eq. (7) and K is determined.

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R = 0.5 \tag{6}$$

$$K = lqr(A, B, Q, R) \tag{7}$$

$$K = [175.6516 \quad 15.0764 \quad -6.1976 \quad -32.8910] \tag{8}$$

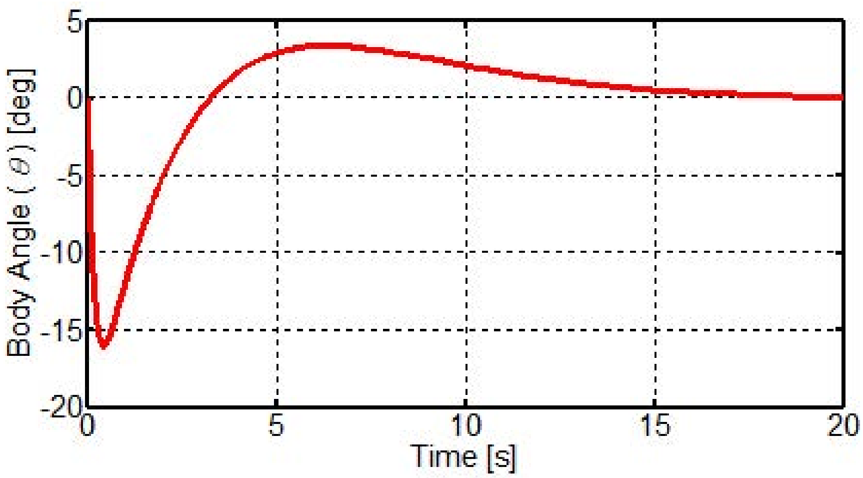

Figure 4 : Body angle of wheeled self-balancing robot.

3. RESULTS AND DISCUSSION

Desired body angle is 0o and desired wheel position is 5m. LQR controlled body angle response is shown in Figure 4. Moreover wheel position of the wheeled self-balancing robot system is given in Figure 5. From these results it can be said body angle and wheel position are reached to desired values. From this result, body angle and wheel position of the wheeled self-balancing robot system are achieved successfully using LQR controller. In addition LQR controlled torque change is given in Figure 6. As a result designed LQR (Linear-quadratic regulator) controller can be used effectively in this kind systems.

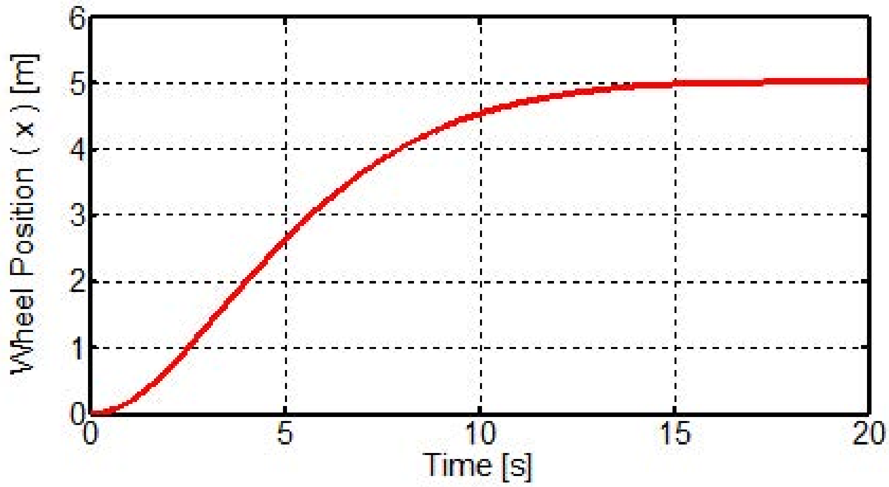

Figure 5 : Wheel position of wheeled self-balancing robot.

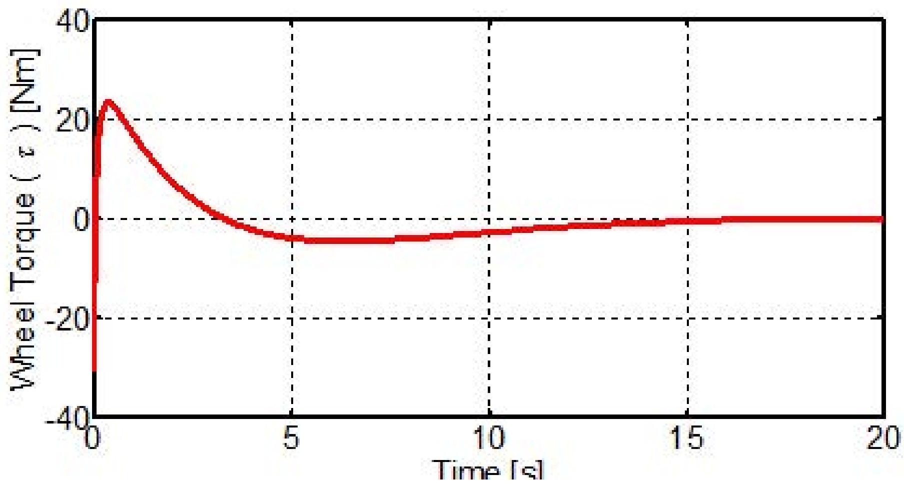

Figure 6 : Control torque applied to the wheel of wheeled self-balancing robot.

3. CONCLUSION

This paper presents modeling, body angle control and wheel position control of a wheeled self-balancing robot. For this aim; a CAD software is used to modeling and Matlab software is used for both linearization and controller design. Thus, these types mechanical systems can be modeled and controlled without mathematical model by using two different engineering software.

A LQR controller is designed in Matlab/Simulink for body angle and wheel position control. LQR controller Q and R matrices are determined by trial error method and A and B matrices generated by linearization in Matlab. Control simulation is realized and controller performance is given in the form of the graphics.

As a result of the paper, accuracy of proposed modeling technique is verified by simulations. Also controller performance and effectiveness are investigated and examined; body angle control and wheel position results of the proposed system are presented in the form of graphics.

The main contribution of the paper to the literature is that different type modeling and linearization approaches are implemented and LQR controller is designed and used in control methodology.

4. ACKNOWLEDGEMENTS

This study has supported by the Coordinatorship of Selçuk University Scientific Research Projects.

REFERENCES

[1] Tinkir, M., Kalyoncu, M., Onen, U. and Botsali, F. M., (2010) "Pid and interval type-2 fuzzy logic control of double inverted pendulum system", Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on, Singapore, pp. 117-121.

[2] Zhao, J., & Spong, M. W. (2001) "Hybrid control for global stabilization of the cart–pendulum system", Automatica, vol.37(12), pp.1941-1951.

[3] Lee, J. H., Shin, H. J., Lee, S. J., & Jung, S. (2013) "Balancing control of a single-wheel inverted pendulum system using air blowers: Evolution of Mechatronics capstone design", Mechatronics, vol.23(8), pp.926-932.

[4] Wu, J., Wang, X., & Wang, H. (2016) "Research on Two-Wheeled Self-Balancing Robot Control Strategy Based on LQR-Fuzzy Algorithm. International Journal of Control and Automation, vol.9(2), pp.31-40.

[5] Aswathy, A., & Sumathi, S. (2016) "A Study of Control of Self Balancing Robot System", Middle-East Journal of Scientific Research, vol.24(3), pp.564-570.

[6] Tayefi, M., & Geng, Z. (2016, May) "A constructive self-balancing controlled Lagrangian for wheeled inverted pendulum", In Control and Decision Conference (CCDC), 2016 Chinese (pp. 1776-1781). IEEE.

[7] Fang, J. (2014). "The LQR controller design of two-wheeled self-balancing robot based on the particle swarm optimization algorithm", Mathematical Problems in Engineering.

[8] Omatu, S., & Deris, S. (1996, November). "Stabilization of inverted pendulum by the genetic algorithm", In Emerging Technologies and Factory Automation, 1996. EFTA'96. Proceedings., 1996 IEEE Conference on vol.1, pp.282-287. IEEE.

[9] Sen, M. A., & Kalyoncu, M. (2015). "Optimisation of a PID Controller for an Inverted Pendulum Using the Bees Algorithm", Applied Mechanics & Materials.

[10] https://en.wikipedia.org/wiki/Inverted_pendulum

[11] https://en.wikipedia.org/wiki/Segway_PT