Design and Control of Two-Wheeled and Self-Balancing Mobile Robot

Conference Paper · September 2021

DOI: 10.1109/ELMAR52657.2021.9550938

4 authors:

Jasmin Velagic
University of Sarajevo
115 PUBLICATIONS **1,172** CITATIONS
SEE PROFILE

Imran Kovac
University of Sarajevo
2 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Adis Panjevic
University of Sarajevo
3 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Adnan Osmanović
University of Sarajevo
13 PUBLICATIONS **71** CITATIONS
SEE PROFILE

Design and Control of Two-Wheeled and Self-Balancing Mobile Robot

Jasmin Velagić, Imran Kovač, Adis Panjević, Adnan Osmanović
Faculty of Electrical Engineering/University of Sarajevo
Sarajevo, Bosnia and Herzegovina
jasmin.velagic@etf.unsa.ba

***Abstract*—The paper focuses on the design and development of a two-wheeled and self-balancing robot as well as its control. The problem is equivalent to the inverted pendulum principle of balancing robots. Dynamic model based PD controller and empirical controller were designed. These controllers are used in closed loop system to provide the robot balance, even when robot is slightly pushed, which normally causes it to fall. The equations of robot motion were derived using Lagrangian and mapped to a transfer function in the complex s-domain. The controller parameters are initially tuned using PID Tuner in Simulink. Then, the zero-order-hold discretization method was applied to implement this control on the Arduino microcontroller. Furthermore, the controller parameters are additionally adjusted through experiments in order to exhibit better control performance. Moreover, the effect of an unexpected disturbance on the robot was taken into consideration. The effectiveness of the designed controllers was verified experimentally.**

***Keywords*—Self balancing robot; Equations of motion; PD empirical controller; Lagrangian method; ZOH discretization**

I. Introduction

In the recent time, a two-wheeled and self-balancing robot has been commonly used for a human transportation [1]. This type of the vehicle is inherently strongly coupled, nonlinear and unstable system. It represents the mobile robot with two wheels on the right and left sides, where is not possible to balance it without an adequate control. When unexpected disturbances affect the robot, the robot maintains its balance with movements of the wheels and tilting of the body. These are provided by the controller that also ensures good stability behaviour of the robot but does not properly guarantee its safety. This means that the robot cannot stay in one place while responding to an external force. Finally, it is evident that there is a trade-off between the robot stability and its safety and control of this type of vehicle is a challenging task.

The main advantage of the two-wheeled and self-balancing robots over other mobile robots is their increased maneuverability due to the reduction of turning radius to zero. Therefore, the robot can achieve smother rotations while changing the course of motion. It allows easy navigation on various terrains, rotation in place to instantly change its motion direction, turning sharp corners, stable running and high energy utilization [2]. These make them very useful for the broad applications, both in the military and civilian fields. The most popular self-balancing robots, Segway and Ninebot, were widely used for patrol transportations [3]. Also, these robots are employed in many industries such as inside factory floors or for tourism in the park. The Anybots QB self-balancing robot was exploited as a service robot platform [2], [4]. The self-balancing robots are also used as robotic assistance platforms for people with disabilities [5], transporting means of baggage [6] and transport of objects to reduce human efforts in workplaces [7], domestic applications or hazardous work environments [8], etc.

Since the two-wheeled and self balancing robot is inherently unstable system, it is needed to be actively controlled to maintain its upright position. In order to fulfill this requirement, its wheels are independently driven to provide robot balance. For this purpose, various linear and nonlinear controllers were designed. The used linear controllers are mainly based on PID (Proportional-Integral-Derivative), LQR (Linear Quadratic Regulator), LQG (Linear-Gaussian Control) and Pole-Placement methods, as well as their combinations [9]-[12]. Also, nonlinear controllers have been employed for the two wheeled and self-balancing robots. Commonly used nonliner controllers include backstepping and sliding mode control [13], [14], fuzzy logic control [15], [16], neural networks [17], Lyapunov control function [18], nonlinear optimal control [19] and reinforcement learning [20]. These controllers ensure the tracking control and stabilization of the two-wheeled robot despite the unknown terrain inclination and the presence of friction and external disturbances. The stability of the sliding mode controllers, backstepping and Lypunov based control functions are established through a Lyapunov analysis.

The main objectives of this paper are development of the two-wheeled and self-balancing robot and design of a controller that allows the robot stand upright and not fall even on inclines and when it slightly pushed. To meet these requirements it is necessary to overcome the instability and cross coupling effects of the robot. For this purpose, we designed and implemented the PD controller based on dynamic model and empirical controller. Dynamic model of the proposed robot is derived using Lagrangian method, while the empirical controller relays on method trial and error. Effectiveness and quality of the designed controllers are verified through both simulations and experiments.

The paper is organized as follows. Section II describes main components of the proposed robot and its control system structure. In Section III the equations of robot motion are derived. The design of PD controller based on dynamics of the two-wheeled and self-balancing robot is explained in Section

IV. The simulation and experimental results are given in Section V, as well as a design of empirical controller. Finally, the conclusions are drawn in Section VI.

II. System Description

The design of the two-wheeled self-balancing robot is based on a single inverted pendulum principle. Our custom made self-balancing robot, named BOSBAR, is shown in Fig. 1.

The mechanical structure of the BOSBAR is composed of the robot body with two wheels driven by two stepper motors (SKU955965) coupled to each of them. The motors are controlled with electrical signals generated by Arduino Uno microcontroller (PWM signals) and driven by the L298H drivers (H-bridge circuits). The control signals are produced on basis measured data acquired by an inertial measurement unit (MPU 6050 IMU). The IMU provides the inclination reading, current angular velocity and acceleration around x,y and z axes data. The data for x axis is used to compute a roll angle of the robot. In order to balance a two-wheeled inverted pendulum robot it is necessary to have accurate information received from the IMU. The electrical wiring diagram that contains mentioned components of the BOSBAR self-balancing robot is shown in Fig. 2.

The feedback control system for the two-wheeled and self-balancing robot is shown in Fig. 3. The input of controller is an error, defined as a difference of desired roll angle of $0°$ and current roll angle. The output voltage of the controller is in the range of 0 to 5 [V], which is then scaled to range 0-12 [V] by using the motor driver. The motor converts a voltage into angular speed and contact force with the ground. The modelled dynamic system is represented by the robot transfer function. This transfer function takes a contact force of the robot with the ground as an input and gives roll angle of the robot as an output. The sensor transfers data about roll angle with small delay.

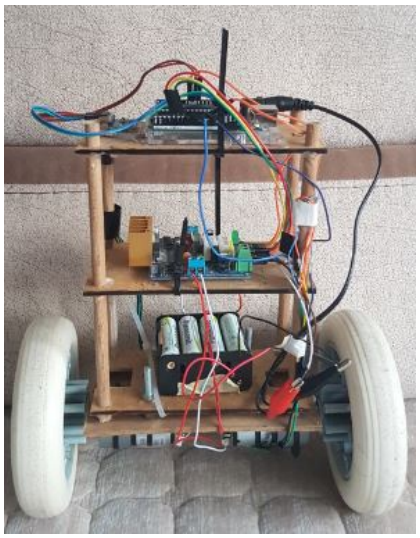

Figure 1. Self-balancing robot BOSBAR.

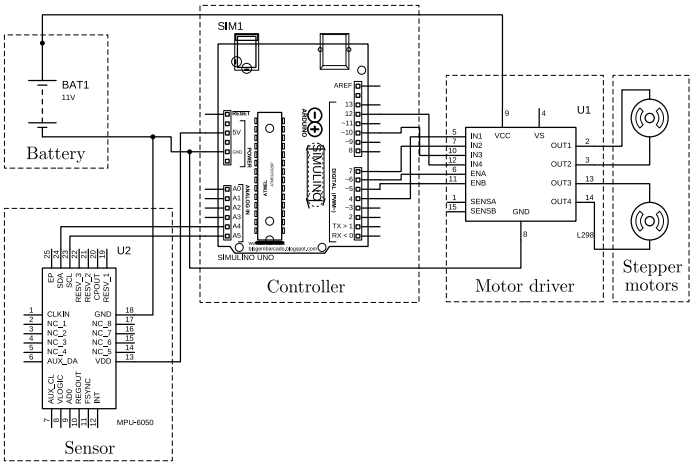

Figure 2. Electrical wiring diagram of the BOSBAR.

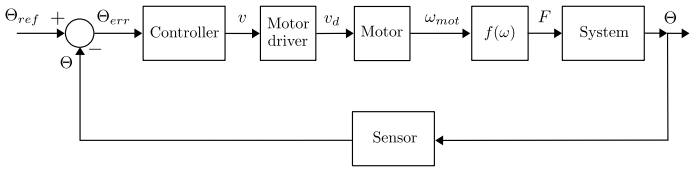

Figure 3. Control system for the self-balancing robot.

III. Dynamic Model of Self-Balancing Robot

The dynamic model of the used robot is mainly based on the inverted pendulum, where its center of mass is above the pivot point (Fig. 4).

The Lagrangian method is used to derive the dynamic equations of robot motion. We have chosen x and Θ as generalized coordinates of a system, where x is a position of the robot (linear motion) and Θ describes rotation of the robot around pitch axis. Then, the Lagrangian is computed using the following term:

$$L = T - V \tag{1}$$

where T and V are sum of kinetic energy of the system and sum of potential energy of the system, respectively.

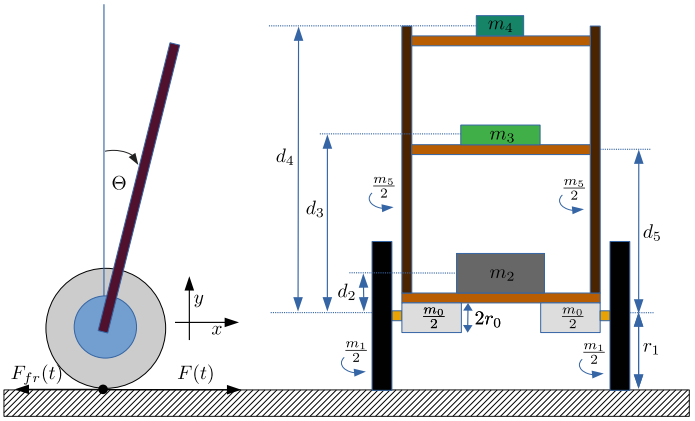

Figure 4. Model of two-wheeled and self-balancing robot.

The potential and kinetic energies of the whole system are given by:

$$V = m_2 g d_2 \cos\Theta + m_3 g d_3 \cos\Theta + m_4 g d_4 \cos\Theta + m_5 g d_5 \cos\Theta \tag{2}$$

$$T = \frac{m_0 \dot{x}^2}{2} + \frac{I_0 \dot{\Theta}^2}{2} + \frac{m_1 \dot{x}^2}{2} + \frac{I_1 \dot{\Theta}^2}{2} + \frac{m_2 \dot{x_2}^2}{2} + \frac{I_2 \dot{\Theta}^2}{2} + \frac{m_3 \dot{x_3}^2}{2} + \frac{I_3 \dot{\Theta}^2}{2} + \frac{m_4 \dot{x_4}^2}{2} + \frac{I_4 \dot{\Theta}^2}{2} + \frac{m_5 \dot{x_5}^2}{2} + \frac{I_5 \dot{\Theta}^2}{2} \tag{3}$$

where m_i, I_i are masses and moments of inertia of elements, $i \in \{1, 2, 3, 4, 5\}$, and r_0, r_1, d_2, d_3, d_4 and d_5 are parameters presented in Fig. 4. In (3) we have included the kinetic energy of elements due to their translation and rotation. Using tables for calculation of the moment of inertia and the parallel axis theorem [21], we can approximate moments of inertia for each element shown in Fig. 4. These approximations are expressed as follows:

$$I_0 = \frac{1}{2} m_o r_0^2, I_1 = \frac{1}{2} m_1 r_1^2, I_2 = \frac{1}{12} m_2 a_2^2 + m_2 d_2^2,$$
$$I_3 = \frac{1}{12} m_3 a_3^2 + m_3 d_3^2, I_4 = \frac{1}{12} m_4 a_4^2 + m_4 d_4^2, I_5 = \frac{1}{3} m_5 d_5^2$$

Based on the model shown in Fig. 4 the following equations are obtained:

$$x_2 = x - d_2 \sin\Theta, x_3 = x - d_3 \sin\Theta, x_4 = x - d_4 \sin\Theta$$
$$x_5 = x - d_5 \sin\Theta$$

as well as their time derivatives:

$$\dot{x_2} = \dot{x} - d_2 \cos\Theta\dot{\Theta}, \dot{x_3} = \dot{x} - d_3 \cos\Theta\dot{\Theta},$$
$$\dot{x_4} = \dot{x} - d_4 \cos\Theta\dot{\Theta}, \dot{x_5} = \dot{x} - d_5 \cos\Theta\dot{\Theta}$$

Substituting previous equations in (3) we can rewrite the equations (2) and (3) as

$$V = m_2 g d_2 \cos\Theta + m_3 g d_3 \cos\Theta + m_4 g d_4 \cos\Theta + m_5 g d_5 \cos\Theta \tag{4}$$

$$T = \frac{m_0 \dot{x}^2}{2} + \frac{I_0 \dot{\Theta}^2}{2} + \frac{m_1 \dot{x}^2}{2} + \frac{I_1 \dot{\Theta}^2}{2} + \frac{m_2 (\dot{x} - d_2 \cos\Theta\dot{\Theta})^2}{2} + \frac{I_2 \dot{\Theta}^2}{2} + \frac{m_3 (\dot{x} - d_3 \cos\Theta\dot{\Theta})^2}{2} + \frac{I_3 \dot{\Theta}^2}{2} + \frac{I_4 \dot{\Theta}^2}{2} + \frac{I_5 \dot{\Theta}^2}{2} + \frac{m_4 (\dot{x} - d_4 \cos\Theta\dot{\Theta})^2}{2} + \frac{m_5 (\dot{x} - d_5 \cos\Theta\dot{\Theta})^2}{2} \tag{5}$$

Now we can substitute equations (4) and (5) into (1) to calculate Lagrangian of the system. According to chosen generalized coordinates x and Θ, the following equations of the robot system are obtained:

$$\frac{\mathrm{d}}{\mathrm{dt}} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = F(t) - F_{fr}(t) \tag{6}$$

$$\frac{\mathrm{d}}{\mathrm{dt}} \frac{\partial L}{\partial \dot{\Theta}} - \frac{\partial L}{\partial \Theta} = 0 \tag{7}$$

where $F_{fr}(t)$ is a friction force, which is modelled as:

$$F_{fr}(t) = f_{fr} \dot{x} \tag{8}$$

where f_{fr} is a friction coefficient and $F(t)$ is a contact force between wheels and surface produced by motors. Finding all necessary partial derivatives, the equations of motion for the robot model illustrated in Fig. 4 can be expressed as

$$M\ddot{x} + K_3(\sin\Theta\dot{\Theta}^2 - \cos\Theta\ddot{\Theta}) = -F(t) + f_{fr}\dot{x} \tag{9}$$

$$K_3\dot{\Theta}\sin\Theta - K_4\dot{\Theta}\cos\Theta + K\sin\Theta - K_5\ddot{\Theta} + K_3\ddot{x}\cos\Theta - K_3\dot{x}\sin\Theta\dot{\Theta} + K_4\sin 2\Theta\dot{\Theta}^2 - K_4\cos^2\Theta\ddot{\Theta} = 0 \tag{10}$$

where $M = m_0 + m_1 + m_2 + m_3 + m_4 + m_5$, $K = g(m_2 d_2 + m_3 d_3 + m_4 d_4 + m_5 d_5)$, $K_3 = m_2 d_2 + m_3 d_3 + m_4 d_4 + m_5 d_5$, $K_4 = m_2 d_2^2 + m_3 d_3^2 + m_4 d_4^2 + m_5 d_5^2$, $K_5 = I_0 + I_1 + I_2 + I_3 + I_4 + I_5$.

Linearising the equations (9) and (10) around an equilibrium point at $\Theta \approx 0^\circ$ and taking into account the following approximations: $\sin\Theta \approx \Theta$, $\cos\Theta \approx 1$, $\dot{\Theta}^2 \approx 0$ and $\Theta\dot{\Theta} \approx 0$, the following equations of motion are obtained:

$$M\ddot{x} - K_3\ddot{\Theta} = -F(t) + f_{fr}\dot{x} \tag{11}$$

$$-K_4\dot{\Theta} - (K_5 + K_4)\ddot{\Theta} + K_3\ddot{x} + K\Theta = 0 \tag{12}$$

These approximations are made to provide the robot to be capable of reaching a stable vertical position with angle zero tilt. Applying the Laplace transform to (11) and (12), with initial conditions set to zero, the transfer function of the system becomes:

$$\frac{\Theta(s)}{F(s)} = \frac{-sK_3}{s^3 H_1 + s^2 H_2 + sH_3 - f_{fr}K} \tag{13}$$

where $H_1 = M(K_4 + K_5) - K_3^2$, $H_2 = MK_4 - f_{tr}(K_4 + K_5)$, $H_3 = MK_4 - f_{tr}K_4$, $X(s) = \mathcal{L}(x(t))$, $\Theta(s) = \mathcal{L}(\Theta(t))$ and $F(s) = \mathcal{L}(F(t))$.

We consider $X(s) = \mathcal{L}(x(t))$ as a free variable. The relationship between the voltage applied to a motor and speed of the stepper motor can be approximated with $\omega(t) = kV_{in}$. Also, the wheel torque can be approximated by

$$\tau = F(t)r_1 = I_1\ddot{\Theta}_m = I_1\dot{\omega}_m \tag{14}$$

Applying Laplace transform with initial conditions set to zero, the equation (14) is transformed into the following transfer function:

$$\frac{F(s)}{\Omega_m(s)} = \frac{sI_1}{r_1} \tag{15}$$

where $\Omega_m(s) = \mathcal{L}(\omega_m(t))$.

Combining (13) and (15) gives the transfer function of the whole system:

$$G(s) = \frac{\Theta(s)}{\Omega_m(s)} = \frac{-s^2 K_3 I_1}{s^3 H_1 r_1 + s^2 H_2 r_1 + sH_3 r_1 - f_{fr} K r_1} \tag{16}$$

Based on the measurements and approximations of unknown parameters of the self-balancing robot system, we have obtained the following parameters values: $k = 2.7636 \frac{rad}{sV}$ (obtained knowing that for input of 11 V we get angular speed of wheels of approximately 290 rpm), $m_0 = 0.3\ kg, m_1 = 0.23\ kg, m_2 = 0.45\ kg, m_3 = 0.048876\ kg, m_4 =$

$0.045876\ kg, m_5 = 0.165\ kg$, $d_2 = 0.032\ m$, $d_3 = 0.118\ m$, $d_4 = 0.203\ m$, $d_5 = 0.108\ m$, $r_1 = 0.064\ m$, $r_0 = 0.015\ m$, $a_2 = a_3 = a_4 = 0.07\ m$, $I_0 = 0.000033750\ kgm^2$, $I_1 = 0.00047104\ kgm^2$, $I_2 = 0.00064455\ kgm^2$, $I_3 = 0.00059005\ kgm^2$, $I_4 = 0.0019\ kgm^2$, $I_5 = 0.00064152\ kgm^2$.

Substituting these parameters in (16), the poles of the system are located at $p_1 = 6.2957 + 0.0000i$, $p_2 = -1.9766 + 4.4774i$ and $p_3 = -1.9766 - 4.4774i$. Notice that $Re\{p_1\} > 0$, which implies that this system is naturally unstable and requires control in order to be stabilized.

IV. Design of Controller Based on Dynamic Model of Robot

As mentioned earlier, the two-wheeled and self-balancing robot must be actively balanced during the motion using an appropriate controller. The proposed control system depicted in Fig. 3 is implemented in Simulink and then used for simulating system behavior and the tuning of PD controller. The value of the friction coefficient $f_{fr} = 3$ was used in simulation. The transfer function of PD controller is expressed as

$$G_{PD}(s) = P + D\frac{N}{1 + N\frac{1}{s}} \tag{17}$$

where P is a proportional controller component, D is a derivative or damping component and N is a filter coefficient for derivative component of the controller. It is important to note that D component of the controller is very sensitive to a noise. Using PID Tuner in Simulink, the components of the proposed PD controller are adjusted in order to stabilize the considered self-balancing robot. Obtained controller parameters are: $K_p = -16920651.8804378$, $K_d = -6808.76134085549$ and $N = 148348.840694991$.

V. Simulation and Experimental Results

In this section, both simulation and experimental results, achieved using PD controller based on dynamic model of the self-balancing robot will be presented as well as implementation of an empirical PD controller. The designed controllers use only one control variable - tilt angle, which is measured using IMU. The main tasks of them are to stabilize the robot and keep it in vertical position at desired tilt angle ($\Theta_r \approx 0$), regardless of the disturbances.

A. Simulation Results

The simulation results obtained using the PD controller designed in Section IV are presented in Fig. 5. In this scenario, the self-balanced robot is moving autonomously and external disturbance is applied to the robot (push or hit) at $t = 1\ s$ after starting from the initial position. From the obtained results can be concluded that the designed controller is capable to balance the robot around the equilibrium point $\Theta \approx 0°$ in the presence of external disturbance.

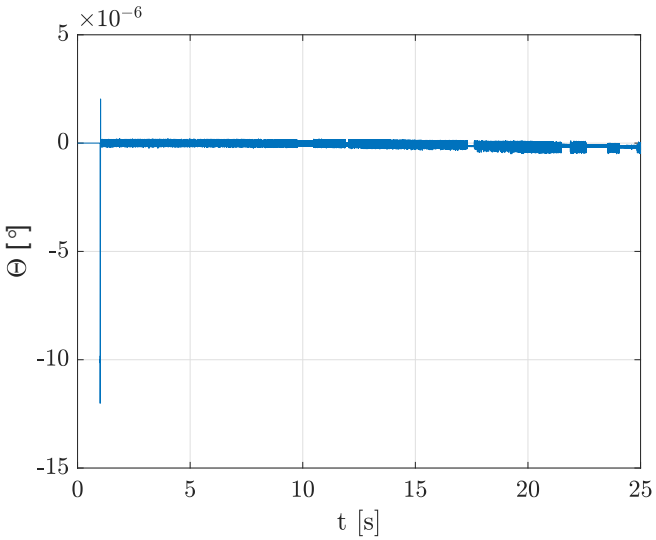

Figure 5. Simulation results for PD controller based on dynamic model.

B. Experimental Results

The implementation of designed controller as well as design of an empirical controller will be explained in this section. Using zero-order-hold method it is possible to discretize and implement the controller (17) on the Arduino Uno control board. In order to make the PD controller causal and implementable, we have added one pole to the transfer function that does not affect the system dynamics. For this purpose we multiplied transfer function of the designed PD controller with $\frac{1000}{s+1000}$ and transfer function of the empirical controller with $\frac{1}{s+1000}$. When applying discretization of (17), we get discrete transfer function of the PD controller given by

$$G_{PD}(z) = \frac{Y(z)}{X(z)} = \frac{b_0 z + b_1}{z^2 + a_1 z + a_2} \tag{18}$$

where a_1, a_2, b_0 and b_1 are coefficients of the transfer function (18). Finally, it is possible to develop a difference equation that will be implemented in our controller. This difference equation can be expressed as

$$y[k] = -a_1 y[k-1] - a_2 y[k-2] + b_0 x[k-1] + b_1 x[k-2] \tag{19}$$

The controller is implemented using an interrupt routine in the microcontroller to ensure that cycle repeats itself every T_s seconds.

1) Implementation of the dynamic model based PD controller: In order to obtain discrete transfer function (18), the sample time of $T_s = 2[ms]$ was chosen. The voltage value $V_{out} = \frac{y[k]}{y_{max}}$ is delivered to motors, where y_{max} is a coefficient of reduction which is obtained experimentally for the robot. In our case $y_{max} = 2500000$. This coefficient is chosen so that we have maximum output voltage when the robot is excited by a step input of amplitude $\frac{\pi}{6}$. This controller was tested in a way that we randomly pushed the robot several times during its motion and observed its balance abilities. Results of this experiment are shown in Fig. 6, where the robot kept his balance and did not fall.

2) Implementation of the empirical controller: Designing controller empirically relays on the trial-and-error tuning method. First, we chose the value of sample time for discretization of $T_s = 10.88$. Then we change values of proportional P component of the controller until the robot reaches a balance. However, the robot twitching is observed even when it slightly pushed. After that we have added a derivative D component of the controller which dampens robots push response. Following these steps we have obtained the following controller parameters values: $K_p = 1600$, $K_d = 47$, $N = 10000$. The voltage value $V_{out} = \frac{y[k]}{y_{max}}$ was also delivered to motors with $y_{max} = 0.28$. The results obtained in this case are depicted in Fig. 7. It is obvious from obtained results that the robot kept his balance and did not fall. The empirical controller exhibited better results because steady state is reached faster with a lower oscillation rate, since it eliminated disturbances more effectively than previous PD controller.

VI. Conclusion

It is noticeable that the empirical PD controller exhibits better control performance in comparison to the PD controller based on dynamic model of the robot. The empirical PD controller compensates a noise faster and with less oscillations as well as external disturbances that occur when we push or hit the robot. This is a result of the fact that the robot system model is not ideal and its parameters are approximated. Also, when adjusting the controller parameters based on the dynamic robot model, it is worthy to note that the model was obtained by linearization of a more complete model around the operating point $\Theta \approx 0°$. In both cases, the robot is controlled to maintain balance, i.e. it remains upright and does not fall even when we push or hit the robot either forward or backward. Improvements are possible by including additional control variables, such as the speed and relative position of the motor, which are measured by incremental encoders. Based on them it is possible to design the controller that ensures the robot stands in one place. This can be accomplished by using a cascade control. Finally, we conclude that the designed controllers meet the desired control performance, i.e. they maintain the balance and equilibrium of the robot, even in presence of external disturbances.

References

[1] J.H. Park and B.K. Cho, "Development of a self-balancing robot with a control moment gyroscope," *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1–11, 2018.

[2] F. Dai, X. Gao and S. Jiang, "A two-wheeled inverted pendulum robot with friction compensation," *Mechatronics*, vol. 30, pp. 116-125, 2015.

[3] G.H. Nguyen, J. Morrell, K.D. Mullens, et al., "Segway robotic mobility platform," In Proc. Society of photo-optical instrumentation engineers (SPIE), 2004, pp. 207–220.

[4] K.M. Tsui, M. Desai, H.A. Yanco, et al., "Exploring use cases for telepresence robots," In Proc. 6th ACM/IEEE international conference on human-robot interaction (HRI), Lausanne, 2011, pp. 11–18.

[5] M. Shino, N. Tomokuni, G. Murata, and M. Segawa, "Wheeled inverted pendulum type robotic wheelchair with integrated control of seat slider and rotary link between wheels for climbing stairs," In Proc. IEEE International Workshop on Advanced Robotics and its Social Impacts, Evanston, USA, 2014, pp. 121-126.

[6] T. Takei, R. Imamura, and S. Yuta, "Baggage transportation and navigation by a wheeled inverted pendulum mobile robot," *IEEE Transactions on Industrial Electronic*, vol. 56, pp. 3985-3994, 2009.

[7] A. Johnson and A. Nasar, "Design and development of a two wheeled self balancing robot for an application of object carrying," *Int. Journal of Engineering Research and Technology*, vol. 6, pp. 580-587, 2017.

[8] M. Galicki, "Robust task space trajectory tracking control of robotic manipulators," *International Journal of Applied Mechanics and Engineering*, vol. 21, pp.547-568, 2016.

[9] R. Ping, M. Chan, K. A. Stol and C. R. Halkyard, "Review of modelling and control of two-wheeled robots," *Annual Review Control*, vol. 37, pp. 89–103, 2013.

[10] S. Wenxia and C. Wei, "Simulation and Debugging of LQR control for two-wheeled self-balanced robot," In Proc. of the Chinese Automation Congress (CAC), Jinan, China, 2017, pp. 2391–2395.

[11] J. Fang, "The LQR controller design of two-wheeled self-balancing robot based on the particle swarm optimization algorithm," *Mathematical Problems in Engineering*, vol. 1, pp. 1-6, 2014.

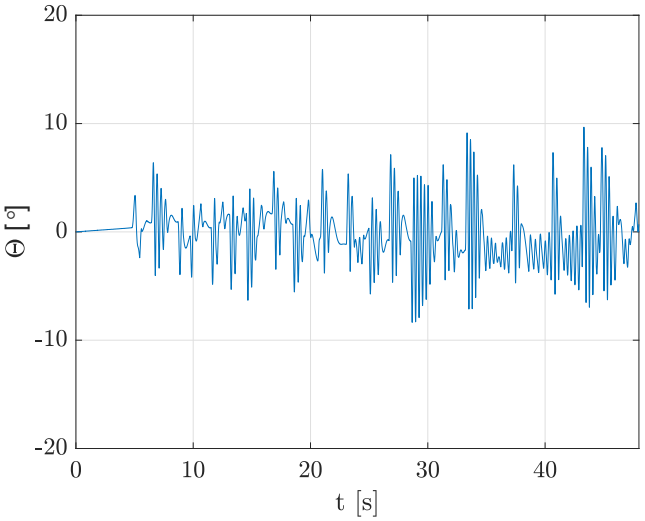

Figure 6. Experimental results for PD controller based on dynamic model.

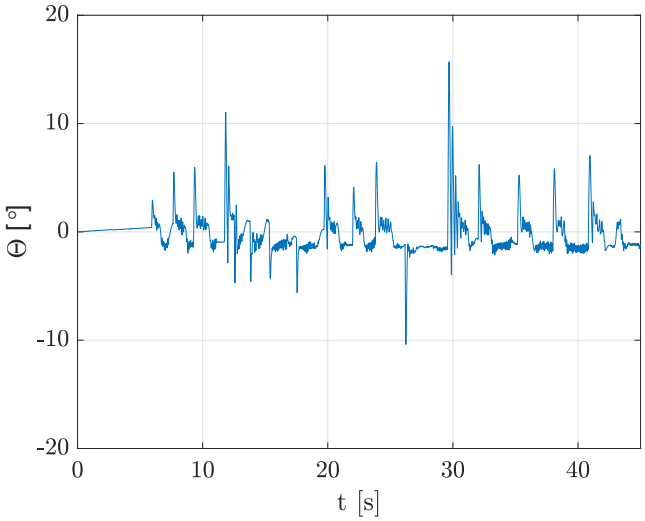

Figure 7. Experimental results for empirical PD controller.

[12] J. Dabbagh and I.H. Altas, "Nonlinear two-wheeled self-balancing robot control using LQR and LQG controllers," In Proc. of 11th International Conference on Electrical and Electronics Engineering, Bursa, Turkey, 2019, pp. 855-859.
[13] I. Jmel, H. Dimassi, S. Hadj-Said and F. M'Sahli, "Adaptive observer-based sliding mode control for a two-wheeled self-balancing robot under terrain inclination and disturbances," *Mathematical Problems in Engineering*, pp. 1-15, 2021.
[14] N. Esmaeili, A. Alfi, and H. Khosravi, "Balancing and trajectory tracking of two-wheeled mobile robot using backstepping sliding mode control: design and experiments," *Journal of Intelligent Robot Systems*, Vol. 87, pp. 601–613, 2017.
[15] J. Huang, S. Member, M. Ri and D. Wu, "Interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum," *IEEE Transactions on Fuzzy Systems*, vol. 26, pp. 2030-2038, 2017.
[16] J.C. Wang, C.P. Huang and J.C. Hung, "Intelligent algorithm design by using fuzzy inference on two-wheeled self-balancing vehicle," In Proc. International Conference on Applied System Innovation, Sapporo, Japan, 2017, pp. 1825–1828
[17] S. Jung and S.S. Kim, "Control experiment of a wheel-driven mobile inverted pendulum using neural network," *IEEE Transactions on Control Systems Technology*, vol. 16, pp. 297–303, 2008.
[18] Z. Yu, T. Tong and S.F. Wong, "Experiment and controller design for two-wheeled robot with nonlinear damping and road disturbance," In Proc. Chinese Control and Decision Conference, Shenyang, China, 2018, pp. 1983–1987.
[19] S. Kim and S. Kwon, "Nonlinear optimal control design for underactuated two-wheeled inverted pendulum mobile platform," *IEEE/ASME Transactions on Mechatronics*, vol. 22, pp. 2803–2808, 2017.
[20] L. Guo, S.A. Asad Rizvi and Z. Lin, "Optimal control of a two-wheeled self-balancing robot by reinforcement learning," *International Journal of Robust and Nonlinear Control*, 2021.
[21] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics - Modelling, Planning and Control*, Berlin:Springer Verlag, Germany, 2009.