# A PID Backstepping Controller For Two-Wheeled Self-Balancing Robot

Nguyen Gia Minh Thao (1), Duong Hoai Nghia (2), Nguyen Huu Phuc (3)
Faculty of Electrical and Electronics Engineering
HoChiMinh city University of Technology (HCMUT)
HoChiMinh city, VietNam
(1) ngmthao@hcmut.edu.vn, (2) dhnghia@hcmut.edu.vn, (3) nhphuc@hcmut.edu.vn

***Abstract* – This paper presents a method to design and control a two-wheeled self-balancing robot and it focus on hardware description, signal processing, discrete Kalman filter algorithm, system modelling and PID backstepping controller design. In the system, signals from angle sensors are filtered by a discrete Kalman filter before being fed to the PID backstepping controller. The objectives of the proposed controller are to stabilize the robot while try to keep the motion of robot to track a reference signal. The proposed PID backstepping controller has three control loops, in which the first loop uses a backstepping controller to maintain the robot at equilibrium, the second loop uses a PD controller to control the position of robot and the last uses a PI controller to control the motion direction. Simulations and experimental results show that the proposed control system has good performances in terms of quick response, good balance, stability .**

***Keywords** – Two wheeled self-balancing robot, Discrete Kalman filter, Backstepping control, PID control, Embedded system.*

## I. INTRODUCTION

Two-wheeled self-balancing robot is a multi-variable and uncertain nonlinear system [1],[2],[7],[8], so that the performance of the robot depends heavily on the signal processing and the control method in use. In recent years, the number of researches on backstepping control have increased [4],[5],[6]. The backstepping approach provides a powerful design tool for nonlinear system in the pure feedback and strict feedback forms. So it is of great interest and feasible to use backstepping approach to design a compatible controller for two-wheeled self-balancing robot when the mathematical model of robot is identified.

The designed two-wheeled self-balancing robot is given in *Figure 1*. Signals from angle sensors are filtered by a discrete Kalman filter before being fed to the PID Backstepping controller. The purposes of the controller are to stabilize the robot and keep the motion of the robot to track a reference signal. The proposed PID Backstepping controller has three control loops (*Figure 7*). The first loop is a nonlinear controller based on backstepping approach, it maintains the tracking error of the pitch angle at zero. The second loop uses a PD controller to control the position of the robot, and the last loop uses a PI controller to control the direction of motion.

The remainder of this paper is organized as follows. Section II describes hardware and sensor signals processing of the designed robot. Section III presents the discrete Kalman filter algorithm. Section IV shows the mathematical model of the robot. Section V presents the method to design the proposed PID Backstepping controller. Section VI shows the simulation results and experimental results of the proposed control system. Section VII includes conclusions and direction of future development of this project. Section VIII is the list of reference documents used for this paper.

## II. HARDWARE DESCRIPTION

### A. *Hardware of the Designed Robot* [10], [11], [12]

*Figure 1* and *Figure 2* show the prototype design of the robot. The designed robot has an aluminum chassis, two 24V-30W DC-Servo motors for actuation, an accelerator and a gyroscope for measuring pitch angle and angular velocity of the body of robot, two incremental encoders built in motors for measuring the position of the wheels. Two serial 12V-3.5Ah batteries make a 24V-3.5Ah power which supplies energy for two DC-Servo motors. Another 12V-1.5Ah battery supplies energy for the central control module. Some operations of the robot (upright and balance, moving forward, moving backward, turning left, turning right,…) which can be controlled from buttons on the central control module or the RF remote control.

Signals processing and control algorithm are embedded in a two-core micro-controller (MCU) MC9S12XDP512 which is a 16-bit MCU of FreeScale.

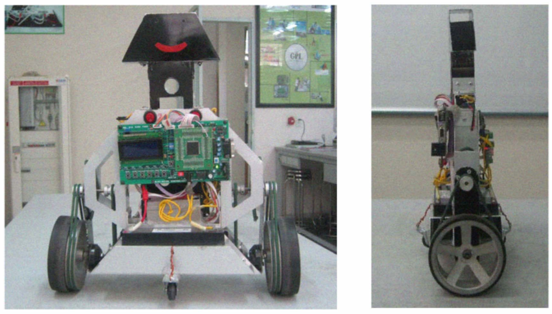

Figure 1. The prototype two-wheeled self-balancing robot

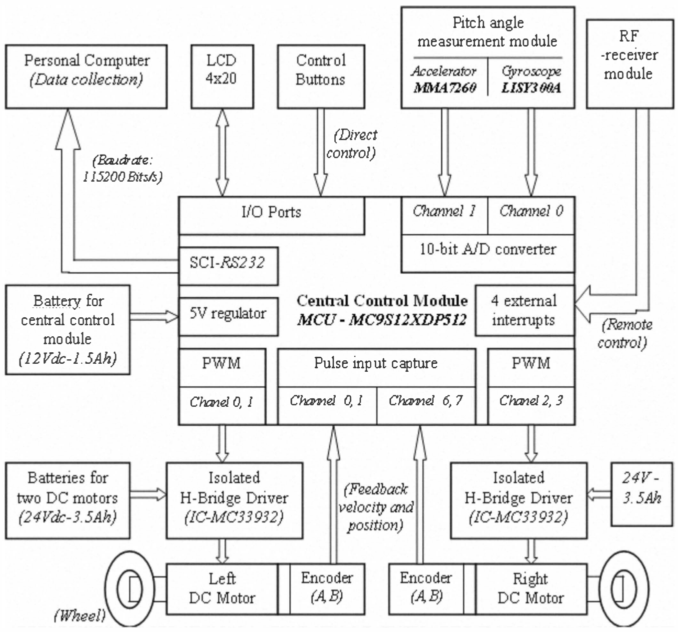


Figure 2. Hardware description

### B. *Sensor Signals Processing* [1]

The angle sensor module includes an accelerator and a gyroscope, two incremental encoders which provide full state

 978-1-4244-9037-0/10/$26.00 

of the robot (*Figure 3*). Signals from accelerator sensor and gyroscope sensor are being fed to the Kalman filter (presented in Section III).

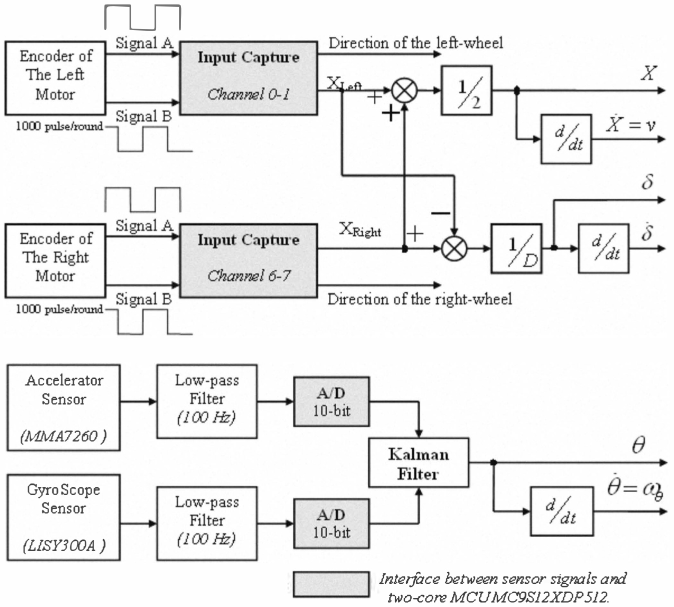


Figure 3. Sensor signals processing

## III. DISCRETE KALMAN FILTER

The Kalman filter estimates a process by using a form of feedback control [3]. The filter estimates the process state at some times and then obtains feedback in the form of (noise) measurements. The equations of discrete Kalman filter can be decomposed into two groups: *Time Update* equations and *Measurement Update* equations, as shown in *Figure 4.*

The *time update* equations use the current state and error covariance estimates to obtain the estimates for the next time step (*Table I*). The *measurement update* equations use the current measurements to improve the estimates which obtained from the *time update* equations (*Table II*).

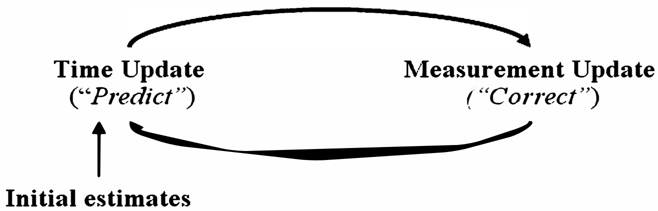


Figure 4. The discrete Kalman filter algorithm

TABLE I. TIME UPDATE EQUATIONS

| | |
|---|---|
| Project the state ahead<br>$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$ | (1) |
| Project the error covariance ahead<br>$P_k^- = AP_{k-1}A^T + Q$ | (2) |

TABLE II. MEASUREMENT UPDATE EQUATIONS

| | |
|---|---|
| Compute the Kalman gain<br>$K_k = P_k^- H^T \left(HP_k^- H^T + R\right)^{-1}$ | (3) |
| Update estimate with measurement $z_k$<br>$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - H\hat{x}_k^-\right)$ | (4) |
| Update the error covariance<br>$P_k = \left(I - K_k H\right)P_k^-$ | (5) |

The parameters of discrete Kalman filter are as follows

$$A = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix};\ B = \begin{bmatrix} 0 \\ 0 \end{bmatrix};\ X = \begin{bmatrix} angle \\ gyro \end{bmatrix};\ R = 0.15$$

$$P_{initial} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};\ Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.004 \end{bmatrix};\ H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

*Figure 5* shows that the angle measurement filtered by Kalman Filter is better than the unfiltered angle measurement.

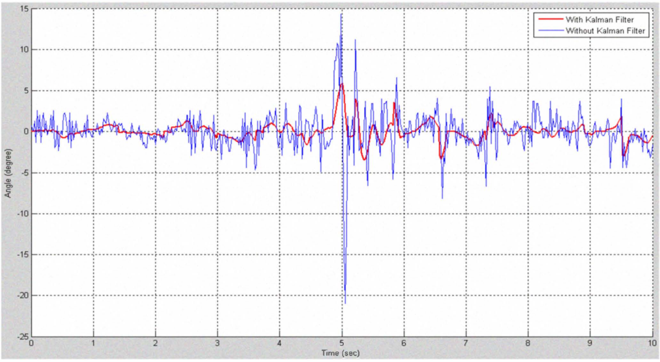


Figure 5. Comparision between filtered and unfiltered angle measurement

## IV. MATHEMATICAL MODEL OF THE ROBOT

The coordinate system of robot is shown in *Figure 6*, we use the Newton's Second Law of motion to identify the mathematical model of robot [1], [7], [8], [9].

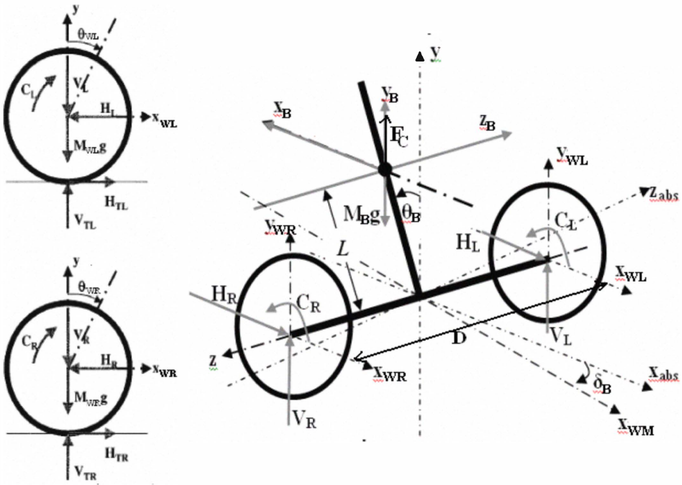

Figure 6. Free body diagram of the chassis and the wheels

- For the left wheel of robot (same as the right wheel).

$$M_W \ddot{x}_{WL} = H_{TL} - H_L \tag{6}$$

$$M_W \ddot{y}_{WL} = V_{TL} - V_L - M_W g \tag{7}$$

$$J_{WL}\ddot{\theta}_{WL} = C_L - H_{TL}R \tag{8}$$

$$x_{WL} = \theta_{WL} R \tag{9}$$

$$J_{WL} = \frac{1}{2} M_{WL} R^2 \tag{10}$$

- For the body of robot.

$$x_B = L\sin\theta_B + \left(\frac{x_{WL} + x_{WR}}{2}\right) \tag{11}$$

$$y_B = -L\left(1 - \cos\theta_B\right) \tag{12}$$

$$M_B \ddot{x}_B = H_L + H_R \tag{13}$$

$$M_B \ddot{y}_B = V_L + V_R - M_B g + F_C$$
$$= V_L + V_R - M_B g + \frac{(C_L + C_R)}{L} \sin \theta_B \quad (14)$$

$$J_B \ddot{\theta}_B = (V_L + V_R) L \sin \theta_B - $$
$$-(H_L + H_R) L \cos \theta_B - (C_L + C_R) \quad (15)$$

$$J_B = \frac{1}{3} M_B L^2 \quad (16)$$

$$\theta = \theta_B \quad (17)$$

Where $V_{TL}, V_{TR}$, $H_{TL}, H_{TR}$ are reaction forces between the wheels and ground. $H_L, H_R, V_L, V_R$ are reaction forces between the chassis and the wheels. The parameters of robot are defined in *Table III*.

TABLE III. PARAMETERS OF ROBOT

| Symbols | Parameter | Value *[unit]* |
|---|---|---|
| $\theta$, $\delta$ | Pitch and yaw angle | *[rad]* |
| $M_W$ | Mass of wheel | 0.5 *[kg]* |
| $M_B$ | Mass of body | 7 *[kg]* |
| $R$ | Radius of wheel | 0.07 *[m]* |
| L | Distance between the centres of the wheels and the robot's centre of gravity | 0.3 *[m]* |
| D | Distance between the contact patches of the wheels | 0.41 *[m]* |
| g | Gravity constant | 9.8 *[$ms^{-2}$]* |
| $C_L, C_R$ | Input torque for left and right wheels | *[N.m]* |

From equations (6) to (17), the state equations of two-wheeld self-balancing robot are written as:

$$\begin{cases} \circ \left( \dfrac{0,75(M_W R + M_B L(\cos\theta))(\cos\theta)}{(2M_W + M_B)L} - 1 \right)\ddot{\theta} = \dfrac{-0,75g(\sin\theta)}{L} + \dfrac{0,75 M_B L(\sin\theta)(\cos\theta)}{(2M_W + M_B)L}(\dot{\theta})^2 + \\ \qquad + \left( \dfrac{0,75(1+(\sin\theta)^2)}{M_B L^2} + \dfrac{0,75(\cos\theta)}{(2M_W + M_B)RL} \right) C_\theta \\ \circ \left( 2M_W + M_B - \dfrac{0,75(M_W R + M_B L(\cos\theta))(\cos\theta)}{L} \right)\ddot{x} = \dfrac{-0,75g(M_W R + M_B L(\cos\theta))(\sin\theta)}{L} + \\ \qquad + M_B L(\sin\theta)(\dot{\theta})^2 + \left( \dfrac{0,75(M_W R + M_B L(\cos\theta))(1+(\sin\theta)^2)}{M_B L^2} + \dfrac{1}{R} \right) C_\theta \end{cases} \quad (18)$$

Define some state variables: $x_1 = \theta,\ x_2 = \dot{\theta},\ x_3 = x,\ x_4 = \dot{x}$.

The state equations of two-wheeled self-balancing robot are rewritten as:

$$\begin{cases} \dot{x}_1 = x_2 & (19.1) \\ \dot{x}_2 = f_1(x_1) + f_2(x_1, x_2) + g_1(x_1) C_\theta & (19.2) \\ \dot{x}_3 = x_4 & (19.3) \\ \dot{x}_4 = f_3(x_1) + f_4(x_1, x_2) + g_2(x_1) C_\theta & (19.4) \end{cases} \quad (19)$$

Where:

$$\circ\ C_\theta = C_L + C_R$$

$$\circ\ f_1(x_1) = \frac{\left( \dfrac{-0.75g(\sin x_1)}{L} \right)}{\left( \dfrac{0.75(M_W R + M_B L(\cos x_1))(\cos x_1)}{(2M_W + M_B)L} - 1 \right)}$$

$$\circ\ f_2(x_1, x_2) = \frac{\left( \dfrac{0.75 M_B L(\sin x_1)(\cos x_1)}{(2M_W + M_B)L} (x_2)^2 \right)}{\left( \dfrac{0.75(M_W R + M_B L(\cos x_1))(\cos x_1)}{(2M_W + M_B)L} - 1 \right)}$$

$$\circ\ g_1(x_1) = \frac{\left( \dfrac{0.75(1+(\sin x_1)^2)}{M_B L^2} + \dfrac{0.75(\cos x_1)}{(2M_W + M_B)RL} \right)}{\left( \dfrac{0.75(M_W R + M_B L(\cos x_1))(\cos x_1)}{(2M_W + M_B)L} - 1 \right)}$$

$$\circ\ f_3(x_1) = \frac{\left( \dfrac{-0,75g(M_W R + M_B L(\cos x_1))(\sin x_1)}{L} \right)}{\left( 2M_W + M_B - \dfrac{0,75(M_W R + M_B L(\cos x_1))(\cos x_1)}{L} \right)}$$

$$\circ\ f_4(x_1, x_2) = \frac{\left( M_B L(\sin x_1)(x_2)^2 \right)}{\left( 2M_W + M_B - \dfrac{0,75(M_W R + M_B L(\cos x_1))(\cos x_1)}{L} \right)}$$

$$\circ\ g_2(x_1) = \frac{\left( \dfrac{0,75(M_W R + M_B L(\cos x_1))(1+(\sin x_1)^2)}{M_B L^2} + \dfrac{1}{R} \right)}{\left( 2M_W + M_B - \dfrac{0,75(M_W R + M_B L(\cos x_1))(\cos x_1)}{L} \right)}$$

From equations (19.1) and (19.2), we can use Backstepping approach to control the pitch angle $\theta$ of robot.

## V. PID BACKSTEPPING CONTROLLER DESIGN

### A. *Structure of Control System*

The structure of the control system has three control loops as shown in *Figure 7*.

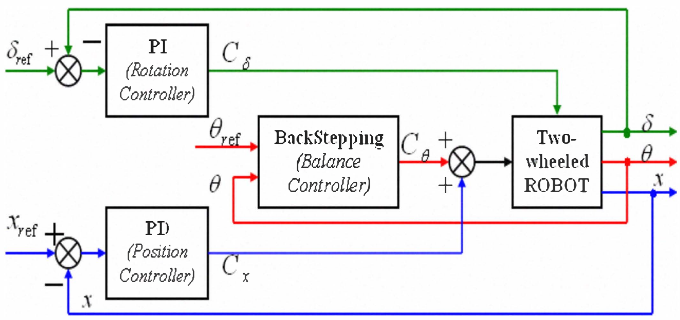


Figure 7. PID Backstepping controller scheme

### B. *Backstepping Controller Design* [4], [5], [6]

**Step 1a**: Define a tracking error as $e_1 = x_{1ref} - x_1$

where $x_{1ref} = \theta_{ref}$ is the reference value for pitch angle $\theta$


978-1-4244-9037-0/10/$26.00 

Define the virtual control $\alpha$ such that $\lim_{t\to\infty} e_1(t) = 0$, as

$$\alpha = k_1 e_1 + c_1 z_1 + \dot{x}_{1ref} \tag{20}$$

where $k_1, c_1$ are positive contants and $z_1 = \int e_1(\tau)d\tau$.

**Step 1b**: The first Lyapunov function is defined as

$$V_1 = \frac{c_1}{2} z_1^2 + \frac{1}{2} e_1^2 \tag{21}$$

The derivation of $V_1$:

$$\dot{V}_1 = c_1 z_1 \dot{z}_1 + e_1 \dot{e}_1 = c_1 z_1 e_1 + e_1 \dot{e}_1 = e_1 \left(c_1 z_1 + \dot{e}_1\right) \tag{22}$$

**Step 2a**: Define the error between the virtual control $\alpha$ and input $x_2$ of (19.2) as $e_2 = \alpha - x_2$

(19.1) is rewritten as $\dot{x}_1 = \alpha - e_2 = k_1 e_1 + c_1 z_1 + \dot{x}_{1ref} - e_2$ (23)

From (23), we have $\dot{e}_1 = \dot{x}_{1ref} - \dot{x}_1 = -k_1 e_1 - c_1 z_1 + e_2$ (24)

The derivation of $e_2$:

$$\begin{aligned} \dot{e}_2 &= \dot{\alpha} - \dot{x}_2 \\ &= \left(k_1 \dot{e}_1 + c_1 e_1 + \ddot{x}_{1ref}\right) - \left(f_1(x_1) + f_2(x_1, x_2) + g_1(x_1) C_\theta\right) \\ &= k_1\left(-k_1 e_1 - c_1 z_1 + e_2\right) + c_1 e_1 + \ddot{x}_{1ref} - f_1(x_1) - f_2(x_1, x_2) - g_1(x_1) C_\theta \\ &= \left(c_1 - k_1^2\right) e_1 - k_1 c_1 z_1 + k_1 e_2 + \ddot{x}_{1ref} - f_1(x_1) - f_2(x_1, x_2) - g_1(x_1) C_\theta \end{aligned} \tag{25}$$

Substituting (24) into (22), we obtain

$$\dot{V}_1 = e_1\left(c_1 z_1 - \left(k_1 e_1 + c_1 z_1\right) + e_2\right) = -k_1 e_1^2 + e_1 e_2 \tag{26}$$

**Step 2b**: The second Lyapunov function is defined as

$$V_2 = V_1 + \frac{1}{2} e_2^2 \tag{27}$$

Compute the derivation of $V_2$ as

$$\dot{V}_2 = \dot{V}_1 + e_2 \dot{e}_2 = -k_1 e_1^2 + e_1 e_2 + e_2 \dot{e}_2 \tag{28}$$

For making $\dot{V}_2$ definite negative, we choose $\dot{e}_2$ as

$$\dot{e}_2 = -k_2 e_2 - e_1 \quad ; \text{with} \quad k_2 \geq 0 \tag{29}$$

Substituting (29) into (28), we obtain

$$\dot{V}_2 = -k_1 e_1^2 - k_2 e_2^2 < 0 \tag{30}$$

From (25), (29) we have the equation of control signal as

$$C_\theta = \frac{\left(1 + c_1 - k_1^2\right) e_1 + \left(k_1 + k_2\right) e_2 - k_1 c_1 z_1 + \ddot{x}_{1ref} - f_1(x_1) - f_2(x_1, x_2)}{g_1(x_1)} \tag{31}$$

### C. *Position Controller Design.* [11], [12]

In this control loop, a PD controller is designed to control the position of the robot (*Figure 8*).

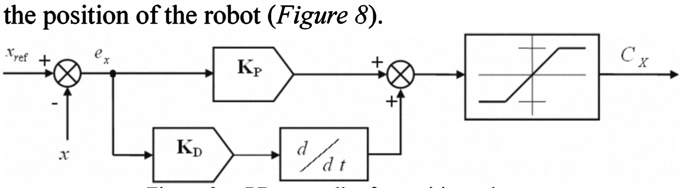


Figure 8. PD-controller for position scheme

When the robot uprights and doesn't move ($\theta_{ref} = 0$), the reference position equals zero ($x_{ref} = 0$).

If the reference pitch angle doesn't equal zero, the robot will move forward ($\theta_{ref} > 0$) or move backward ($\theta_{ref} < 0$). In this operation, the position PD controller isn't used.

### D. *Rotation Controller Design.* [11]

In this control loop, a PI controller is designed to control the direction of motion (*Figure 9*).

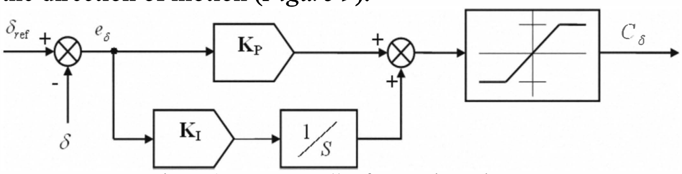


Figure 9. PI-controller for rotation scheme

When the robot doesn't move, or moves forward, or moves backward (doesn't turn left-right), the reference yaw angle equals zero ($\delta_{ref} = 0$).

The control signal from the *Rotation controller* combines with the control signals from the *Balance controller* and *Position Controller* to make two separately control signals for left and right wheels (*Figure 10*).

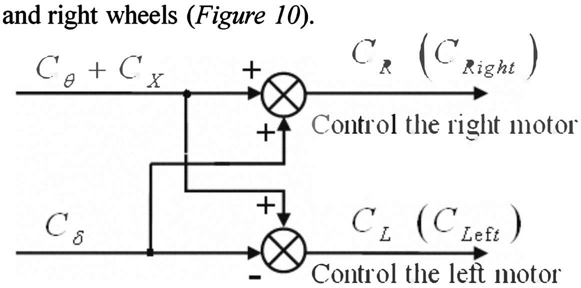


Figure 10. The scheme of control signals for two DC-Servo motor

## VI. SIMULATIONS, EXPERIMENTAL RESULTS

### A. *Simulations*

The parameters of the robot are defined in *Table III*. Both the pitch angle and the position of the robot are shown simultaneously in *Figure 11*, *Figure 12*, *Figure 13*, *Figure 14* . In these figures, the graph of pitch angle [deg] is the top graph, the graph of position [m] is the bottom graph and the horizontal axis is time [sec].

- *The robot maintains at equilibrium and doesn't move.* ($\theta_{ref} = 0$ [deg], $x_{ref} = 0$ [m])

  *Figure 11* shows the response of the robot with small initial pitch angle ($\theta_0 = 5$ [deg]), and *Figure 12* shows the response of the robot with large initial angle ($\theta_0 = 25$ [deg]) . Both the angle ($\theta$) and the position of robot ( $x$ ) converge to zero for two initial conditions.

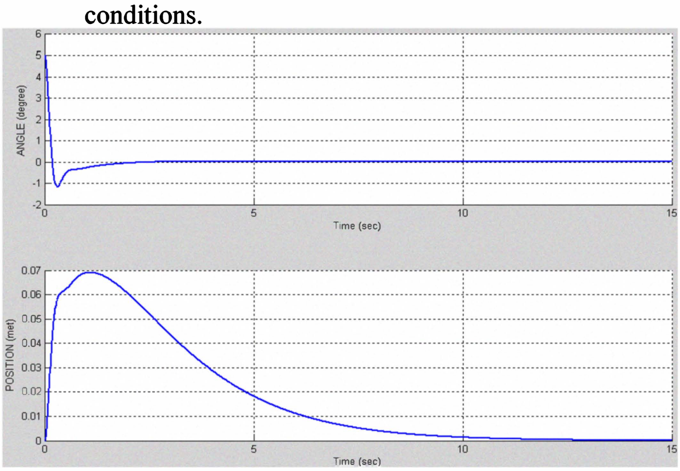


Figure 11. The robot maintains at equilibrium with small initial pitch angle ($\theta_0 = 5$ [deg])

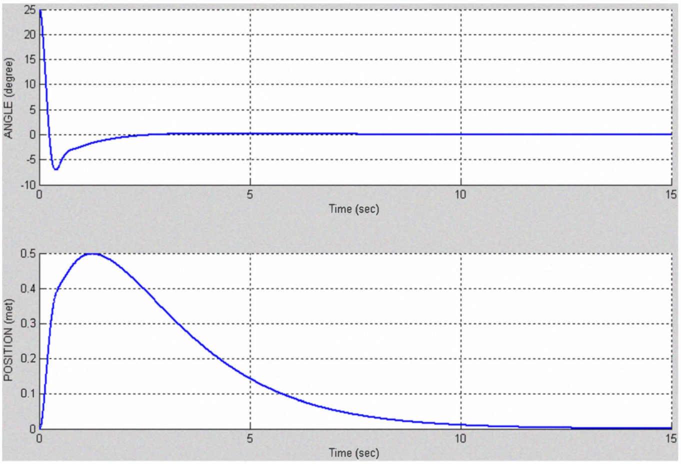


Figure 12. The robot maitains at equilibrium with large initial pitch angle ($\theta_0 = 25$ [deg])

- *The robot tracks a reference pitch angle,* $\theta_{ref} \neq 0$ [deg].

*Figure 13* shows the response of robot with positive reference pitch angle ( $\theta_{ref} = 5$ [deg]). In this condition, the robot moves forward, so that the PD-controller for position isn't used ($K_P$ =0 ; $K_D$ =0).

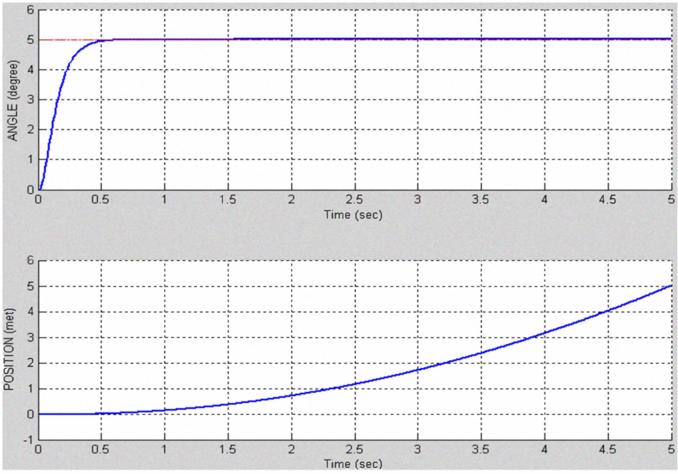


Figure 13. The robot tracks a reference pitch angle ($\theta_{ref} = 5$ [deg])

- *The robot moves to the set position* ($x_{ref} = 2$ [m])

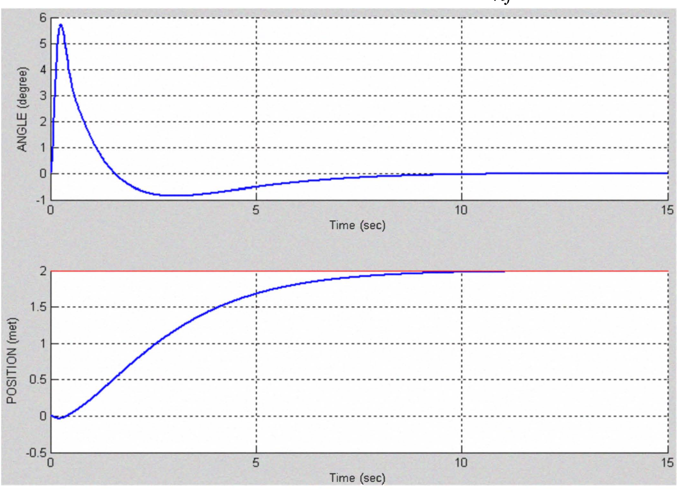


Figure 14. The robot moves to the new position which is at the distance of two mets.

### B. *Experimental Results*

- Parameters of the proposed PID Backstepping controller are shown in *Table IV*.

TABLE IV. PARAMETERS OF THE PID BACKSTEPPING CONTROLLER

| Controller | Parameters |
|---|---|
| **Backstepping** – *Balance* | $k_1 = 110.5$; $k_2 = 21.4$; $c_1 = 3$ |
| **PD** – *Position* | $K_P = 60$ ; $K_D = 7.5$ |
| **PI** – *Rotation* | $K_P = 47.5$ ; $K_I = 5$ |

- *The robot is in equilibrium and doesn't move.* ($\theta_{ref} = 0$ [deg], $x_{ref} = 0$ [m])

*Figure 15* shows the experimental result that the designed robot is in equilibrium. In this operation, the Backstepping controller for balance and the PD controller for position are active. The pitch angle of robot ( $\theta$ ) is regulated under 1.1 [deg], and the position of robot ( $x$ ) is regulated under 0.04 [m].

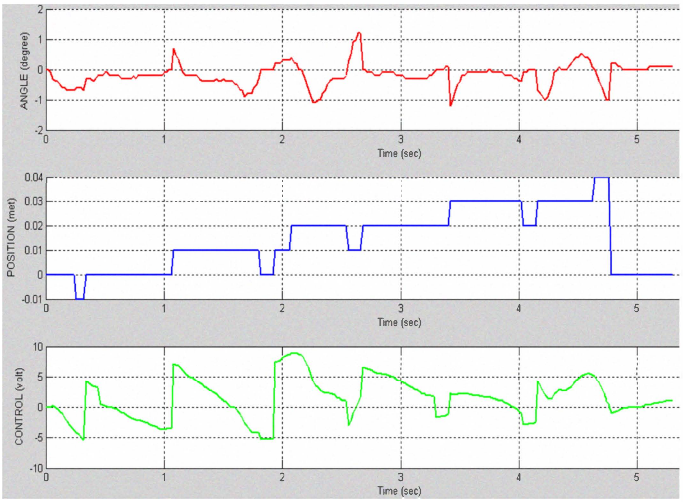


Figure 15. A experimental result that the designed robot is in equilibrium

- *The robot maintains at equilibrium with an external disturbance.* ($\theta_{ref} = 0$ [deg])

*Figure 16* shows the response of the designed robot when it is affected by a external disturbance. The robot still maintains at equilibrium and the pitch angle of robot is regulated after 1.5[sec]. The Backstepping controller for balance is active and the PD controller for position isn't active. So, when the robot is affected by an external disturbance, the robot moves forward to regulate the pitch angle. Finally, the robot is balanced in the new position.

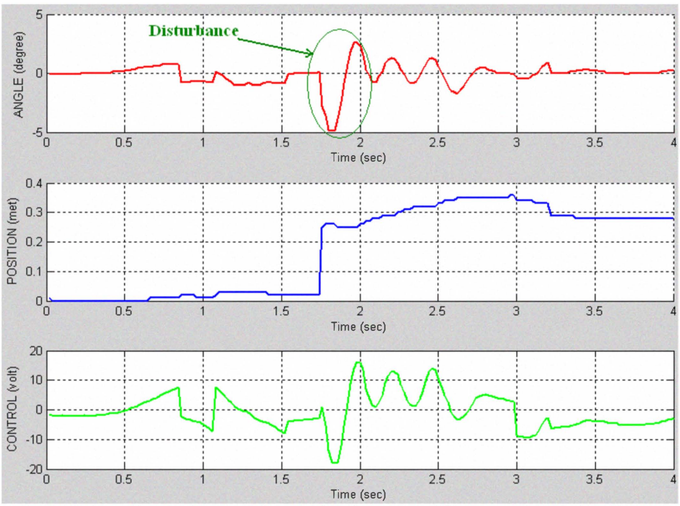


Figure 16. The designed robot maintains at equilibrium when it is affected by a external disturbance

- *The robot moves to the set position* ($x_{ref} \neq 0$ [m])

In this operation, both the Backstepping controller for balance and the PD controller for position are active. When the robot moves forward, the reference signals of the proposed controller are: $\theta_{ref} = 0$, $x_{ref} > 0$. When the robot moves backward, reference signals of the proposed controller are: $\theta_{ref} = 0$, $x_{ref} < 0$. *Figure 17* and *Figure 18* show the responses of the robot when the robot moves forward (with $x_{ref} = 0.8$ [m]) and moves backward (with $x_{ref} = -0.4$ [m]).

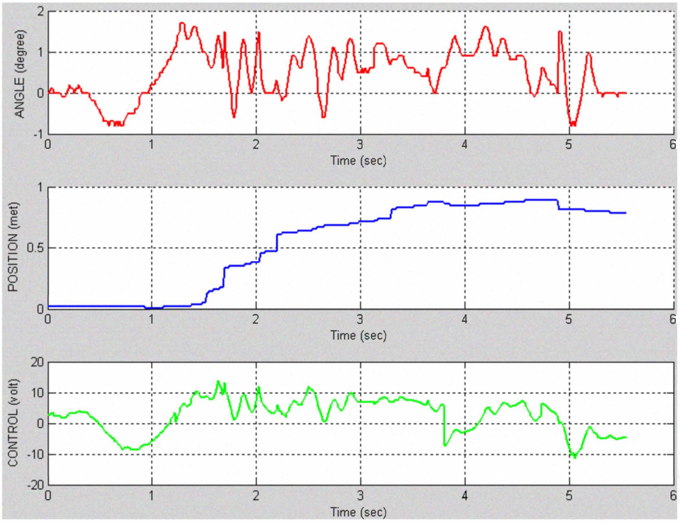


Figure 17. The designed robot moves forward to the positive set position.

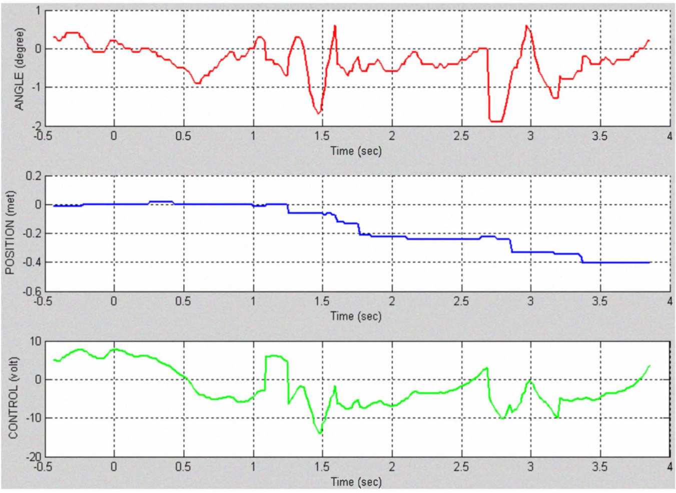


Figure 18. The robot moves backward to the negative set position.

### C. *Demo Operation Videos of Robot*

Some demo operation videos of the robot were uploaded to Youtube as

- The robot is in equilibrium and doesn't move. **http://www.youtube.com/watch?v=ork5lFdlT14**
- The robot is affected by an external disturbance, and it still maintains at equilibrium. **http://www.youtube.com/watch?v=efx41ltkxuM**
- The robot maintains at equilibrium, moves forward and moves backward. **http://www.youtube.com/watch?v=RNO9TUtpog4**

## VII. CONCLUSION

This paper presented a method to design and control a two-wheeled self-balancing robot. Simulations and experimental results show that the nonlinear controller based on Backstepping approach has good performances in terms of quick response, good balance and robust against disturbance. In the future, the proposed Backstepping controller will be combined with fuzzy logic to design a hybrid controller for improving the performance of the robot.

## VIII. REFERENCES

[1] Felix Grasser, Aldo D'Arrigo, Silvio Colombi, Alfred Ruffer , "JOE: A Mobile Inverted Pendulum", IEEE Trans. Electronics, vol. 49, no 1, Feb 2002, pp. 107-114.

[2] S.W.Nawawi , M.N.Ahmad , J.H.S. Osman , "Development of Two-Wheeled Inverted Pendulum Mobile Robot", SCOReD07,Malaysia, Dec 2007, pp 153-158

[3] Greg Welch, Gary Bishop , "An Introduction to the Kalman Filter" Siggraph 2001 Conference, Course 8.

[4] Tatsuya Nomura , Yuta Kitsuka, Hauro Suemistu, Takami Matsuo , "Adaptive Backstepping Control for a Two-Wheeled Autonomous Robot", ICROS-SICE International Joint Conference 2009, Aug. 2009, pp 4687 - 4692.

[5] Xiaogang Ruan , Jianxian Cai , "Fuzzy Backstepping Controller for Two-Wheeled Self-Balancing Robot", 2009 International Asia Conference on Informatics in Control, Automation and Robotics.

[6] Arbin Ebrahim, Gregory V.Murphy, "Adaptive Backstepping Controller Design of an Inverted Pendulum", System Theory, 2005, Proceedings of the Thirty-Seventh Southeastern Symposium on, pp. 172-174.

[7] D. Y. Lee, Y. H. Kim, B. S. Kim, Y. K. Kwak , "Dynamics and Control of Non-holonomic Two Wheeled Inverted Pendulum Robot", Proceedings of the eighth International Symposium on Artificial Life and Robotics, 2003.

[8] Jingtao Li, Xueshan Gao, Qiang Huang, Matsumoto, "Controller design of a two-wheeled inverted pendulum mobile robot", Mechatronics and Automation 2008, ICMA 2008, IEEE International Conference on.

[9] S.W.Nawawi , M.N.Ahmad, J.H.S. Osman , "Real-Time Control of Two-Wheeled Inverted Pendulum Mobile Robot", www.waset.org/journals/waset/v39/v39-39.pdf

[10] David P. Anderson , "nBot – a Balancing Robot", 2003 http://www.geology.smu.edu/~dpa-www/robo/nbot/

[11] Brian Kuschak, "bk-Bot, yet another 2-wheel balancing robot", 2007 www.bkinnovation.com/bkbot/

[12] TedLarson, Bob Allen, "Bender - a balancing robotic" , 2003 www.tedlarson.com/robots/balancingbot.htm