# Nonlinear Controllers for Two-wheeled Self-Balancing Robot

**Conference Paper** · November 2011



4 authors:

Gia Minh Thao NGUYEN
Toyota Technological Institute


Mai Tuan Dat
KU Leuven


Nghia Duong
Ho Chi Minh City University of Technology (HCMUT)


Nguyen Huu Phuc
Ho Chi Minh City University of Technology (HCMUT)

*Proceedings*

# THE ASEAN SYMPOSIUM ON AUTOMATIC CONTROL

# ASAC 2011

*November 8-9, 2011*

*Ho Chi Minh City, Vietnam*

*Organized by*

**Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology**

**Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University**

# Nonlinear Controllers For Two-Wheeled Self-Balancing Robot

Nguyen Gia Minh Thao (1) , Mai Tuan Dat (2) , Duong Hoai Nghia (3) , Nguyen Huu Phuc (4)
Faculty of Electrical and Electronics Engineering
HoChiMinh City University of Technology (HCMUT)
HoChiMinh City, Vietnam
(1) ngmthao@hcmut.edu.vn , (2) tuandat.mai@hcmut.edu.vn, (3) dhnghia@hcmut.edu.vn, (4) nhphuc@hcmut.edu.vn

***Abstract* – This paper presents methods to design a compatible controller for the two-wheeled self-balancing robot. It focuses on hardware description and nonlinear controllers design. The signals from angle sensors are filtered by a discrete Kalman filter before being fed to the embedded controller. Objectives of the proposed controllers are twofold: regulation of the pitch angle and tracking the desired position. Each proposed controllers based nonlinear control method has two loops. The inner loop is a PD position tracking controller. The outer loop is a pitch angle regulator based Backstepping approach and Sliding mode control. Simulations and experimental results show that two proposed nonlinear controllers have better performances than other controllers shown in this paper (PID multi-loop controller and Pole-Placement controller).**

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

## I. INTRODUCTION

Two-wheeled self-balancing robot is a multi-variable and uncertain nonlinear system [1],[2],[4],[10], so that the performance of the robot depends heavily on the signal processing and the control method in use. The backstepping approach and sliding mode control provide powerful design tools for nonlinear system in pure and strict feedback forms. So it is feasible to use nonlinear control approaches ([5],[6],[7],[8],[9]) to design a compatible controller for two-wheeled self-balancing robot when the mathematical model of the robot is identified.

The designed two-wheeled self-balancing robot is given in *Figure 1*. The purposes of the controller are to stabilize the robot and keep the motion of the robot to track a reference signal. Each proposed controllers has two control loops (from *Figure 6* to *Figure 8*). The inner loop uses a PD controller to control the position of the robot. The outer loop is a pitch angle regulator based on Backstepping approach (see *Figure 7*) and Sliding mode (see *Figure 8*), respectively.

The remainder of this paper is organized as follows. Section II describes hardware and sensor signals processing of the designed robot. Section III shows the mathematical model of the robot. Section IV presents the proposed controller schemes. Section V illustrates simulations and experimental results of the control system. Section VI includes conclusions and direction of future development of this project. Section VII is the list of reference documents used for this paper.

## II. HARDWARE DESCRIPTION

### A. Hardware of the Designed Robot

*Figure 1* shows 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 12V-3Ah batteries are connected serially to make a 24V-3Ah power supplying energy for two DC-Servo motors. Another 12V-1.2Ah battery supplies energy for the central control module. Signals processing and control algorithm are embedded in a 16-bit two-core micro-controller (MCU) MC9S12XDP512 of FreeScale company. Operations of the robot (upright and balance, moving forward, moving backward, turning left, turning right,…) can be controlled from buttons on the control module or the RF remote.

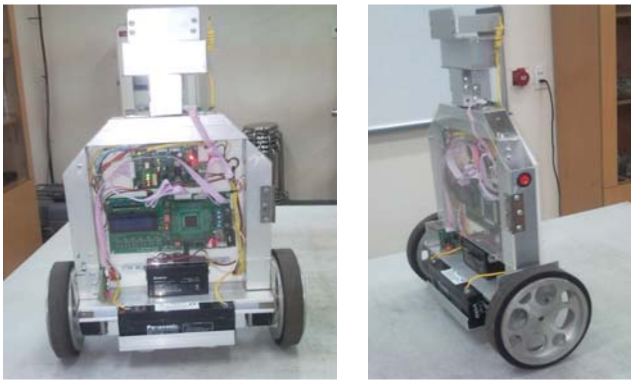

Figure 1. The prototype two-wheeled self-balancing robot

### B. Sensor Signals Processing

The angle sensor module includes an accelerator and a gyroscope, two incremental encoders built in motors, which provide full state of the robot (see *Figure 2*). Signals from accelerator and gyroscope sensors are filtered by the Kalman filter [3] before being fed to the proposed controller. *Figure 3* shows that the angle measurement filtered by Kalman Filter is better than the unfiltered angle measurement.

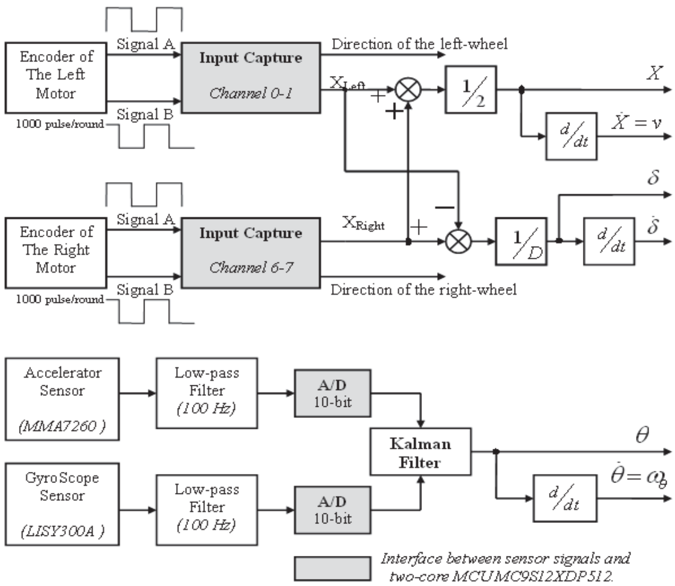


Figure 2. Sensor signals processing

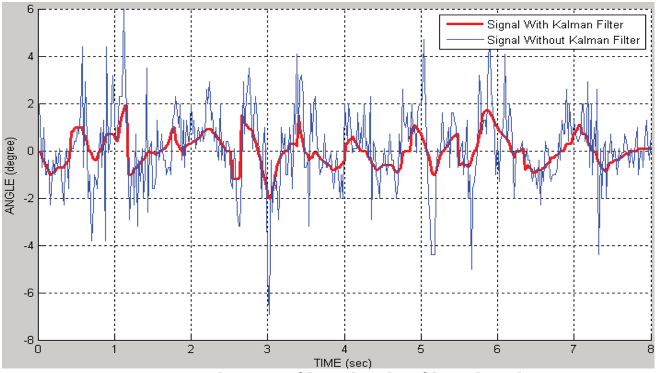


Figure 3. Comparision between filtered and unfiltered angle measurement

## III. MATHEMATICAL MODEL OF THE ROBOT

The coordinate system of robot is shown in *Figure 4*, we use the Newton's Second Law of motion to identify the mathematical model of robot [1], [4], [8], [10].

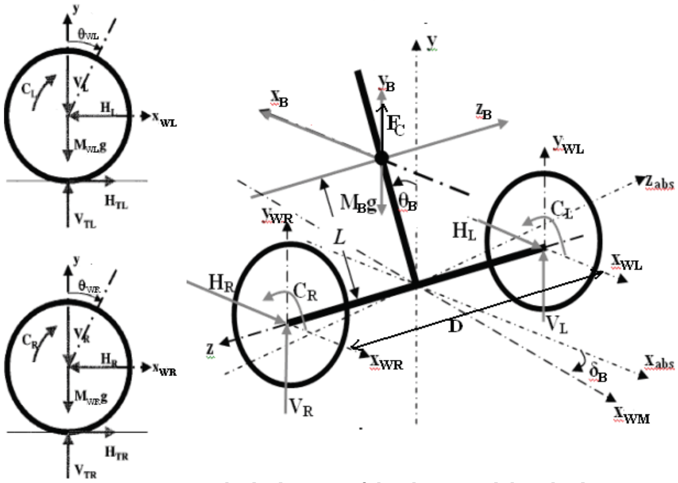

Figure 4. Free body diagram of the chassis and the wheels

Where, $\theta$ : Pitch angle [rad] ; $\delta$ : 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 centre 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].

$g$ : Acceleration of gravity, 9.8 [ms$^{-2}$].

$C_L, C_R$ : Input torque for left and right wheels [Nm].

$V_{TL}, V_{TR}, H_{TL}, H_{TR}, H_L, H_R, V_L, V_R$ : Reaction forces [N].

The state equations of two-wheeld self-balancing robot are:

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

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

(1) is rewritten as:

$$
\begin{cases}
\dot{x}_1 = x_2 & (2.1) \\
\dot{x}_2 = f_1(x_1)+f_2(x_1,x_2)+g_1(x_1)\,C & (2.2) \\
\dot{x}_3 = x_4 & (2.3) \\
\dot{x}_4 = f_3(x_1)+f_4(x_1,x_2)+g_2(x_1)\,C & (2.4)
\end{cases}
\tag{2}
$$

where: $C = C_L + C_R$

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

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

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

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

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

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

## IV. CONTROLLERS DESIGN

### A. *Structure of the Control System*

This paper proposes various controllers for the two-wheeled self-balancing robot, such as Pole-Placement controller (see *Figure 5*), PID multi-loop controller (see *Figure 6*), PD- Backstepping controller (see *Figure 7*) and PD Sliding mode controller (see *Figure 8*). Each proposed control systems (described in *Figure 6, Figure 7, Figure 8*) has two loops. The inner loop is a PD position tracking controller with slow dynamic (see *Figure 9*), and its output is clamped between $\pm 25\%$ of the rated torque of the DC motor. The outer loop is a pitch angle regulator.

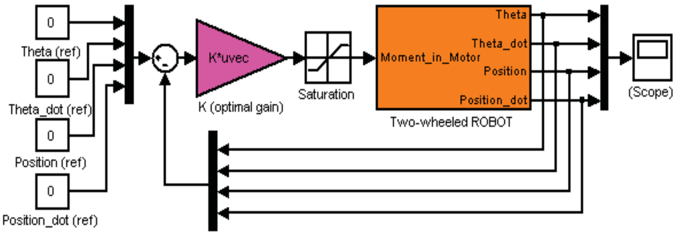


Figure 5. Pole placement controller scheme

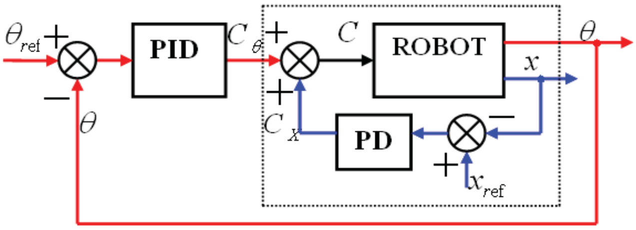


Figure 6. PID multi-loop controller scheme

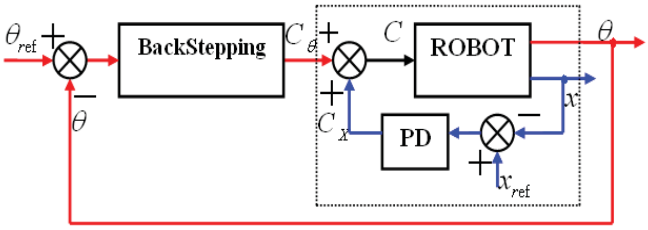


Figure 7. PD Backstepping controller scheme

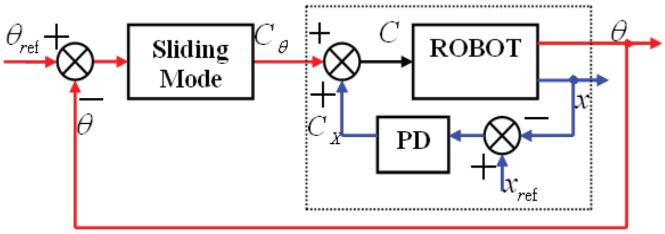


Figure 8. PD Sliding mode controller scheme

### B. *Pole Placement Controller Design .* *(Figure 5)*

Linearation at equilibrium point ($\theta << 1$ [radian]), we have:

$$\sin\theta \approx \theta;\ (\sin\theta)^2 \approx 0;\ \cos\theta \approx 1 \ .$$

Thus, (1) is rewritten as:

$$\begin{cases} \ddot{\theta} = \dfrac{gM_B}{X}\theta - \dfrac{Y}{X}C \\ \ddot{x} = \left(-\dfrac{4}{3}L\dfrac{gM_B}{X} + g\right)\theta + \left(\dfrac{4LY}{3X} - \dfrac{1}{M_B L}\right)C \end{cases} \tag{3}$$

Where: $X = \dfrac{4}{3}M_B L - \dfrac{M_B}{(2M_W + M_B)}(M_B L + M_W R)$

$$Y = \frac{M_B}{(2M_W + M_B)R} + \frac{1}{L}$$

From (1) and (3) , we have the state-space equations of the robot

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \dfrac{gM_B}{X} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\dfrac{4}{3}L\dfrac{gM_B}{X} + g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -\dfrac{Y}{X} \\ 0 \\ \dfrac{4LY}{3X} - \dfrac{1}{M_B L} \end{bmatrix} C \tag{4}$$

To ensure the transient responses (settling time, overshoot) and stability of the designed two-wheeled robot [11],[12],[13], we can choose poles of the control system as **V=[-4+2*i ; -4-2*I ; -1+i ; -1-i].** Then the optimal gain ($\mathbf{K_{opt}}$) of the pole-placement controller is calculated by the Ackerman formula.

$\mathbf{K_{opt}}$ = [$K_1$; $K_2$; $K_3$; $K_4$] = **[-8.4209; -1.1898; -0.4371; -0.6119]**

### C. *Backstepping Controller Design.* *(see Figure 7 )*

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

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

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

$$\alpha = k_1 e_1 + c_1 z_1 + \dot{x}_{1ref} \tag{5}$$

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

The first Lyapunov function is defined as

$$V_1 = \frac{c_1}{2}z_1^2 + \frac{1}{2}e_1^2 \tag{6}$$

The derivation of $V_1$:

$$\dot{V}_1 = c_1 z_1 \dot{z}_1 + e_1\dot{e}_1 = c_1 z_1 e_1 + e_1\dot{e}_1 = e_1(c_1 z_1 + \dot{e}_1) \tag{7}$$

Define the error between the virtual control $\alpha$ and input $x_2$ of (2.2) as $e_2 = \alpha - x_2$

(2.1) is rewritten as $\dot{x}_1 = \alpha - e_2 = k_1 e_1 + c_1 z_1 + \dot{x}_{1ref} - e_2$ (8)

From (8), we have $\dot{e}_1 = \dot{x}_{1ref} - \dot{x}_1 = -k_1 e_1 - c_1 z_1 + e_2$ (9)

The derivation of $e_2$:

$$\begin{aligned} \dot{e}_2 &= \dot{\alpha} - \dot{x}_2 = \left(k_1\dot{e}_1 + c_1 e_1 + \ddot{x}_{1ref}\right) - \left(f_1(x_1) + f_2(x_1,x_2) + g_1(x_1)C\right) \\ &= k_1(-k_1 e_1 - c_1 z_1 + e_2) + c_1 e_1 + \ddot{x}_{1ref} - f_1(x_1) - f_2(x_1,x_2) - g_1(x_1)C \\ &= (c_1 - k_1^2)e_1 - k_1 c_1 z_1 + k_1 e_2 + \ddot{x}_{1ref} - f_1(x_1) - f_2(x_1,x_2) - g_1(x_1)C \end{aligned} \tag{10}$$

where $C = C_X + C_\theta$ (see *Figure 7*)

Substituting (9) into (7), we obtain

$$\dot{V}_1 = e_1\left(c_1 z_1 - (k_1 e_1 + c_1 z_1) + e_2\right) = -k_1 e_1^2 + e_1 e_2 \tag{11}$$

The second Lyapunov function is defined as

$$V_2 = V_1 + \frac{1}{2}e_2^2 \tag{12}$$

Compute the derivation of $V_2$ as

$$\dot{V}_2 = \dot{V}_1 + e_2\dot{e}_2 = -k_1 e_1^2 + e_1 e_2 + e_2\dot{e}_2 \tag{13}$$

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

$$\dot{e}_2 = -k_2 e_2 - e_1 \ \ ; \text{with} \ \ k_2 \geq 0 \tag{14}$$

Substituting (14) into (13), we obtain

$$\dot{V}_2 = -k_1 e_1^2 - k_2 e_2^2 < 0 \tag{15}$$

From (10), (14) we have the equation of control signal as

$$C_\theta = \frac{(1 + c_1 - k_1^2)e_1 + (k_1 + k_2)e_2 - k_1 c_1 z_1 + \ddot{x}_{1ref} - f_1(x_1) - f_2(x_1,x_2)}{g_1(x_1)} - C_X \tag{16}$$

### D. *Sliding Mode Controller Design.* *(see Figure 8 )*

Define the tracking error for the pitch angle as

$$e = x_{1ref} - x_1 \tag{17}$$

Where $x_{1ref} = \theta_{ref}$ is the reference value of the pitch angle of the robot ($\theta$). We have

$$\dot{e} = \dot{x}_{1ref} - \dot{x}_1 = \dot{x}_{1ref} - x_2 \tag{18}$$

$$\ddot{e} = \ddot{x}_{1ref} - \ddot{x}_1 = \ddot{x}_{1ref} - \dot{x}_2 \tag{19}$$

The sliding function is defined as

$$S = \tau\dot{e} + e \tag{20}$$

Where $\tau > 0$ is the desired time constant of the sliding phase. We have

$$\dot{S} = \tau\ddot{e} + \dot{e} \tag{21}$$

Substituting (18),(19) into (21), we obtain

$$\begin{aligned}\dot{S} &= \tau\left(\ddot{x}_{1ref} - \dot{x}_2\right) + \left(\dot{x}_{1ref} - x_2\right) \\ &= \tau\left(\ddot{x}_{1ref} - f_1(x_1) - f_2(x_1,x_2) - g_1(x_1)C\right) + \dot{x}_{1ref} - x_2 \end{aligned} \tag{22}$$

where $C = C_X + C_\theta$ (see *Figure 8*)

The control signal $C$ is determined such that

$$\dot{S} = -K\left[sat(S)\right] \tag{23}$$

Where

- $sat(.)$ is the saturation function

$$sat(S) = \begin{cases} -0.4\ , & \text{if } S < -0.4 \\ S\ , & \text{if } -0.4 \le S \le 0.4 \\ 0.4\ , & \text{if } S > 0.4 \end{cases} \tag{24}$$

  The saturation value of *sat(.)* is chosen by a trial and error method.
- $K$ is a positive constant. A large value of $K$ results in a robust controller but it also causes chattering phenomenon.

It follows from (22), (23) that

$$C_\theta = \frac{\ddot{x}_{1ref} - f_1(x_1) - f_2(x_1,x_2) + \dfrac{K}{\tau}\left[sat(S)\right] - \dfrac{x_2}{\tau} + \dfrac{\dot{x}_{1ref}}{\tau}}{g_1(x_1)} - C_X \tag{25}$$

### E. *The PD Position Controller Design.*

In this control loop, a PD controller (shown in PID multi-loop, PD Backstepping and PD Sliding mode controllers) is designed to control the position of the robot (*Figure 9*). Parameters of the PD position controller are chosen from the optimal gain ($\mathbf{K_{opt}}$ = [$K_1$; $K_2$; $K_3$; $K_4$]) in the section of pole placement controller design, particularly $K_P = K_3 = -0.4317$ ; $K_D = K_4 = -0.6119$.

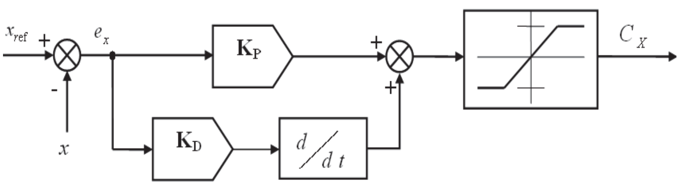


Figure 9. PD-controller for position scheme

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 ($\theta_{ref} < 0$). Furthermore, the position PD controller isn't used in this operation of the robot.

## V. SIMULATIONS, EXPERIMENTAL RESULTS

### A. *Simulations*

The parameters of four proposed controllers are defined in *Table I*. Both the pitch angle and the position of the robot are shown simultaneously from *Figure 10* to *Figure 12* .

TABLE I. PARAMETERS OF THE PROPOSED CONTROLLERS

| *Controller* | *Parameters* |
|---|---|
| Pole Placement | $K_{opt} = [K_1;\ K_2;\ K_3;\ K_4] =$ [-8.4209; -1.1898; -0.4371; -0.6119] |
| PID Multi-loop | $K_P = -9.1;\ K_I = -0.8;\ K_D = -1.4$ |
| | $K_P = -0.4371$ ; $K_D = -0.6119$ |
| PD – Backstepping | $K_1 = 17$ ; $K_2 = 3.8$ ; $C_1 = 10.1$ |
| | $K_P = -0.4371$ ; $K_D = -0.6119$ |
| PD – Sliding mode | $\tau = 0.21$ ; $K = 9.7$ |
| | $K_P = -0.4371$ ; $K_D = -0.6119$ |

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

  *Figure 10* shows the response of the robot with small initial pitch angle ($\theta_0 = 5$ [degree]), and *Figure 11* illustrates the response of the robot with large initial angle ($\theta_0 = 20$ [degree]). Both the pitch angle and the position of the robot converge to zero in two conditions. These results demonstrate that the robot can keep balance well.

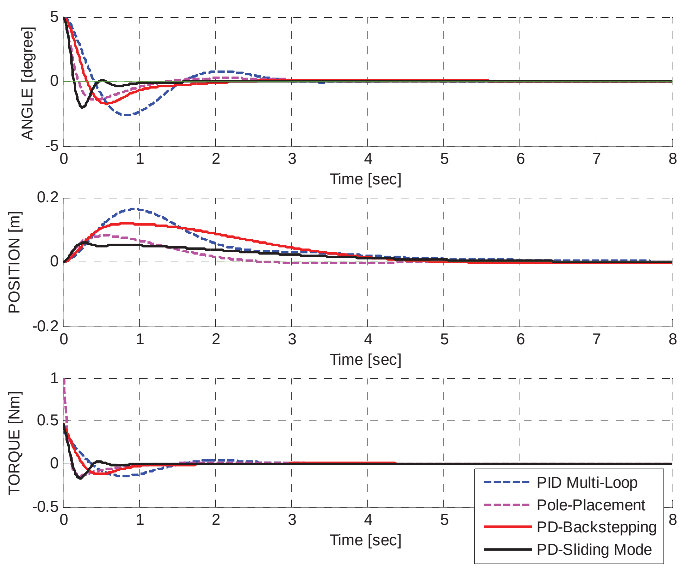


Figure 10. The robot maintains at equilibrium with small initial pitch angle $\left(\theta_0 = 5[\text{degree}]\right)$

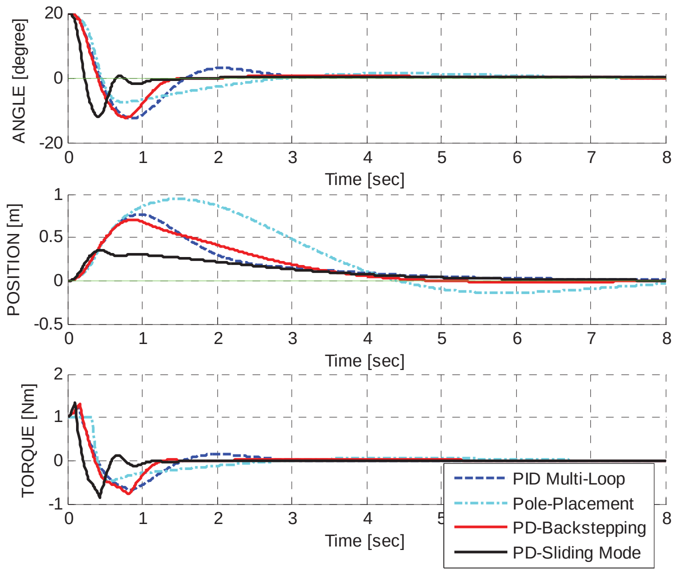


Figure 11. The robot maitains at equilibrium with large initial pitch angle $\left(\theta_0 = 20\left[\deg ree\right]\right)$

- *The robot moves to the set positive position* ($\theta_{ref} = 0$ [degree], $x_{ref} = 3$ [m]; see *Figure 12*)

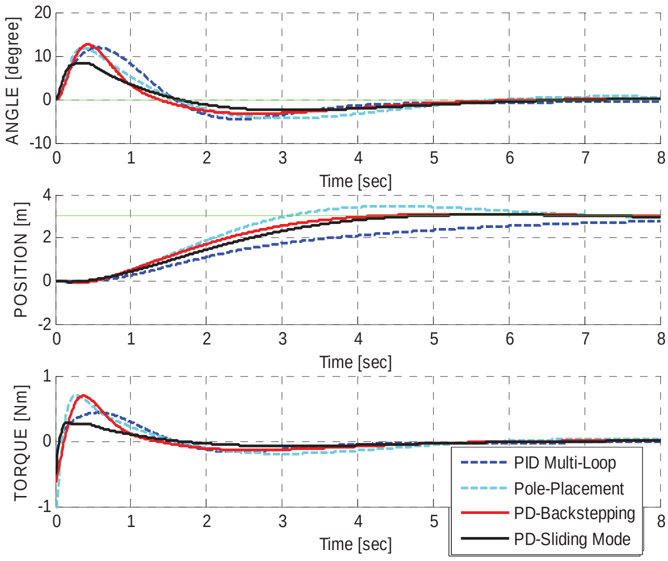


Figure 12. The robot moves to the new position which is at the distance of two mets.

From simulation results, we observe that two proposed controllers based nonlinear control approaches (Backstepping, Sliding mode) have better performances in terms of settling time and overshoot than others presented in this paper (PID multi-loop, Pole placement).

## B. *Experimental Results*

In the current research time, we chose the PID multi-loop controller and the PD-Backstepping controller to implement into the designed robot. Responses of the control system are shown from *Figure 13* to *Figure 18* for comparison.

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

*Figure 13* and *Figure 14* show the experimental results that the designed robot is in equilibrium and don't move. The pitch angle of the robot ($\theta$) is regulated under 2 [degree] (used PID multi-loop controller) and under 1.1 [degree] (used PD-Backstepping controller).

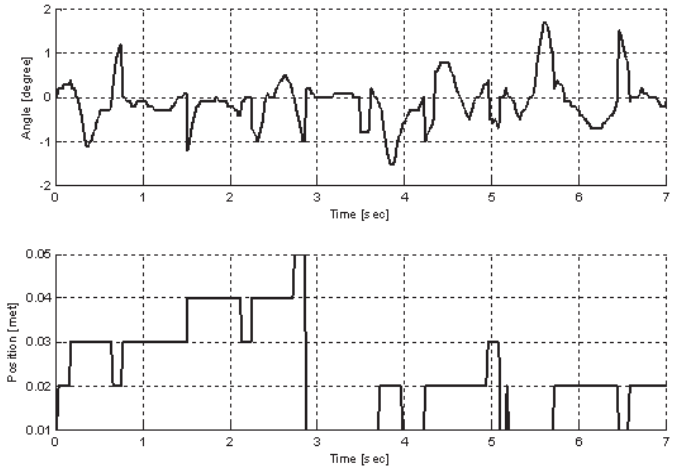


Figure 13. A experimental result that the designed robot is in equilibrium – (used PID multi-loop controller)

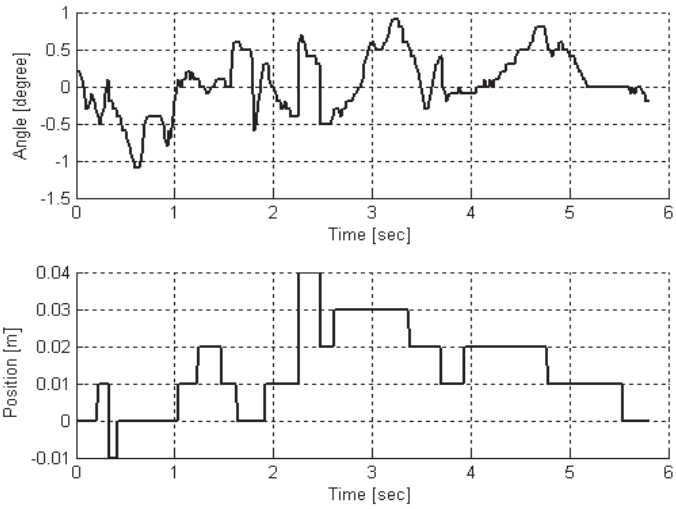


Figure 14. A experimental result that the designed robot is in equilibrium – (used PD Backstepping controller)

- *The robot maintains at equilibrium with an external disturbance.*($\theta_{ref} = 0$ [degree]; *Figure 15*, *Figure 16*)

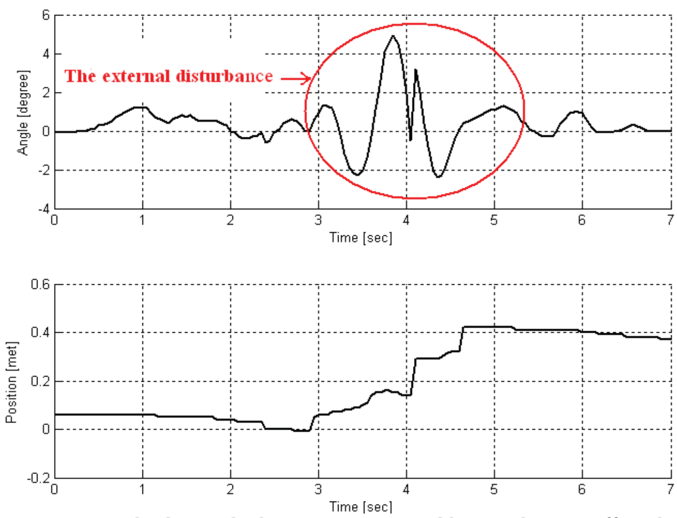


Figure 15. The designed robot maintains at equilibrium when it is affected by a external disturbance – (used the PID multi-loop controller)

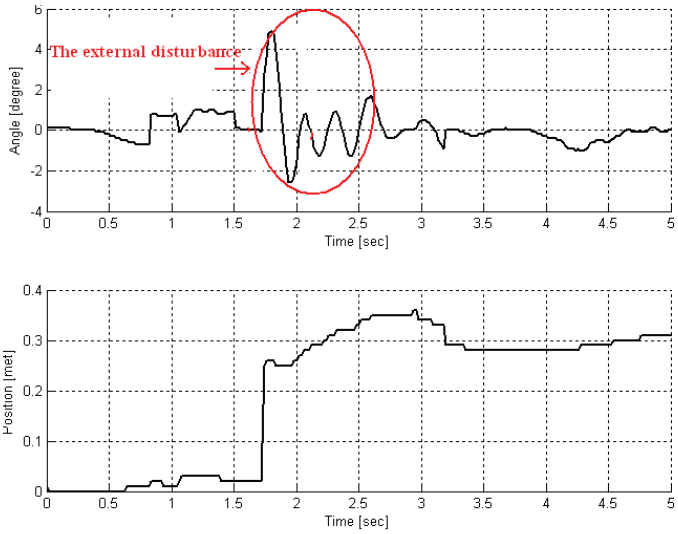


Figure 16. The designed robot maintains at equilibrium when it is affected by a external disturbance – (used the PD Backstepping controller)

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

*Figure 17* and *Figure 18* show that the designed robot moves forward to set position ( $x_{ref} = 1\,[\mathrm{m}]$), and it takes 4[sec] and 2.5[sec] with PID multi-loop controller, PD-Backstepping controller, respectively.

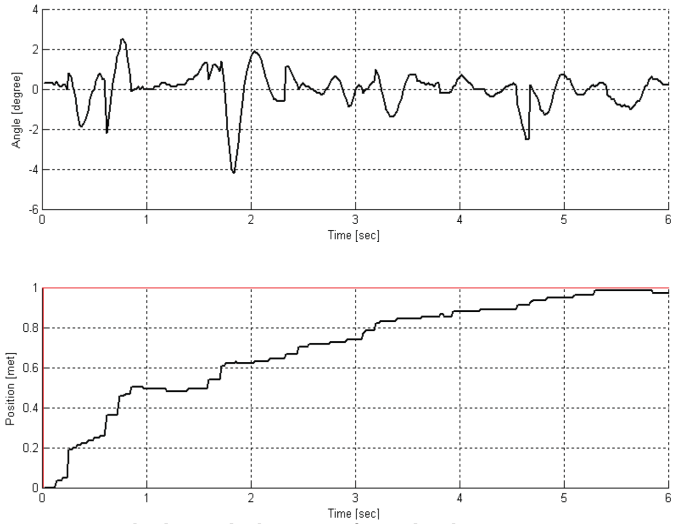


Figure 17. The designed robot moves forward to the positive set position – (used PID multi-loop controller)

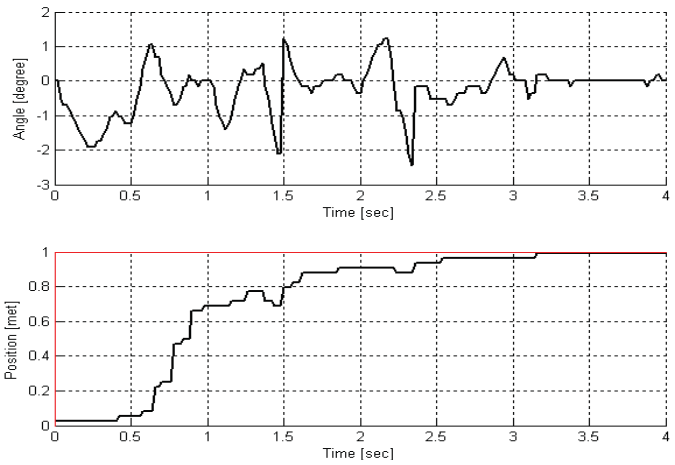


Figure 18. The robot moves backward to the negative set position – (used PD Backstepping controller)

### C. *Demo Operation Videos of the Designed 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=1CMCgXJOACM**
- The robot maintains at equilibrium with external disturbance, moves forward and moves backward.
  **http://www.youtube.com/watch?v=RGbuzXyYayg**

## VI. CONCLUSION

This paper presented approaches to design a compatible controller for a two-wheeled self-balancing robot. Simulations show that two proposed controllers based on nonlinear control methods (Backstepping approach and Sliding mode) have better performances than the PID multi-loop and pole placement controllers in terms of quick response, good balance and robust against disturbance. Furthermore, experimental results illustrate that the PD Backstepping controller has more improvements than the PID multi-loop controller.

## VII. REFERENCES


[1] Felix Grasser, Aldo D'Arrigo, Silvio Colombi, Alfred Ruffer , "JOE: A Mobile Inverted Pendulum", IEEE Trans. Electronics, vol. 49, 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, pp153-158

[3] Greg Welch, Gary Bishop; "An Introduction to the Kalman Filter"; Siggraph 2001 Conference, pp. 8-15.

[4] Tatsuya Nomura , Yuta Kitsuka, Hauro Suemistu, Takami Matsuo , "Adaptive Backstepping Control for a Two-Wheeled Autonomous Robot", Proceedings of 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", Proceedings of 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", Proceedings of the Thirty-Seventh Southeastern Symposium on, 2005, pp. 172-174.

[7] Shui-Chun Lin, Ching-Chih Tsai, Hsu-Chih Huang, "Nonlinear adaptive sliding mode control design for two-wheeled human transportation vehicle", Proceedings of Systems Man and Cybernetics 2009, IEEE International Conference on, pp 1965 – 1970.

[8] Dianwei Qian, Jianqiang Yi, Dongbin Zhao, "Hierarchical Sliding Mode Control to Swing up a Pendubot", Proceedings of American Control Conference, 2007, ACC '07, pp 5254 – 5259.

[9] Jian Huang, Hongwei Wang,T. Matsuno, T. Fukuda, K. Sekiyama, "Robust velocity sliding mode control of mobile wheeled inverted pendulum systems", Proceedings of Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, pp 2983 – 2988.

[10] 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.

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

[12] Lv Qiang, Wang Ke-ke, Wang Guo-sheng; "Research of LQR controller based on Two-wheeled self-balancing robot"; Proceedings of Control and Decision Conference 2009, CCDC '09, pp. 2343 – 2348.

[13] Ruan Xiaogang, Liu Jiang, Di Haijiang, Li Xinyuan; " Design and LQ Control of a two-wheeled self-balancing robot"; Proceedings of Control Conference, 2008. CCC 2008, pp. 275 – 279.

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

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

[16] Segway co., "The simply moving scooter", www.segway.com