

## Điều khiển thời gian thực robot hai bánh tự cân bằng sử dụng bộ điều khiển PID mờ tự chỉnh

### Real-time control of a two-wheeled self-balancing robot using self-tuning PID fuzzy controller

Nguyễn Văn Khanh  
Trường ĐH Cần Thơ  
e-Mail: vankhanh@ctu.edu.vn

Trần Thanh Hùng  
Trường ĐH Cần Thơ  
e-Mail: tthung@ctu.edu.vn

#### Tóm tắt

Bài báo trình bày một phương pháp thiết kế và điều khiển robot hai bánh tự cân bằng. Việc thiết kế tập trung chính vào việc mô tả phần cứng, xử lý tín hiệu, thuật toán lọc bù và thiết kế bộ điều khiển PID mờ tự hiệu chỉnh thông số. Mục tiêu của bài báo là cân bằng robot đồng thời điều khiển vị trí và góc nghiêng bám theo tín hiệu tham chiếu. Bộ điều khiển được đề nghị gồm hai vòng điều khiển. Vòng điều khiển thứ nhất gồm hai bộ điều khiển PD mờ điều khiển cân bằng robot tại vị trí tham chiếu. Vòng điều khiển thứ hai sử dụng một bộ điều khiển PID tự hiệu chỉnh thông số điều khiển góc xoay của robot. Bộ điều khiển được chạy thời gian thực trên Kit STM32F4 Discovery sử dụng hệ điều hành thời gian thực freeRTOS. Kết quả thực nghiệm cho thấy bộ điều khiển đề nghị đã hoạt động tốt trên hệ thống nhúng và cho đáp ứng nhanh, cân bằng tốt, ổn định.

**Từ khóa:** Robot hai bánh tự cân bằng, bộ điều khiển mờ, bộ điều khiển PID, hệ thống nhúng, hệ điều hành thời gian thực.

#### Abstract:

This paper aims to represent a method to design and control a two-wheeled self-balancing robot. This work is to concentrate on hardware description, signal processing, complementary filter algorithm and self-tuning Fuzzy PID controller. The original objective of this controller is that stabilize the robot at the up-right equilibrium while keeping track the reference signals of position and rotation angle. The proposed controller is comprised of two control loops. The first loop is to use two fuzzy PD controllers which balance robot at a specific position reference. The second loop is to use a self-tuning PID controller to control the rotation angle of the robot. The whole controller is ported and run on STM32F4 Discovery kit based on the freeRTOS realtime operating system. The experimental results indicate that the proposed controller system runs exactly on embedded system and has desired performance in term of fast response, good balance and stabilize with disturbance.

**Keywords:** Two-wheeled self-balancing robot, fuzzy control, PID control, embedded system, realtime operating system.

#### Ký hiệu

Ký hiệu	Đơn vị	Ý nghĩa
$q, q_{accel}, q_{gyro}$	Radian	Góc nghiêng robot, góc nghiêng tính từ cảm biến gia tốc, góc nghiêng tính từ con quay hồi chuyển
$x_{robot}, x_{Left}, x_{Right}$	Meter	Vị trí của robot, bánh xe trái, bánh xe phải
$j_{robot}$	Radian	Góc xoay của robot
$u_R, u_L$		Tín hiệu điều khiển hai động cơ phải và trái của robot
D	Meter	Khoảng cách hai bánh xe của robot

#### Chữ viết tắt

LQR	Linear-Quadratic Regulator
PID	Proportional Integral Derivative
ARM	Advanced RISC Machines
RTOS	Real-time Operating System
DSP	Digital Signal Processing
PWM	Pulse Width Modulation
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
DMA	Dynamic Memory Access
FPU	Floating Point Unit
GUI	Graphical User Interface
IMU	Inertial Measurement Unit

### 1. Phần mở đầu

Robot hai bánh tự cân bằng là một thiết bị tự động di chuyển bằng hai bánh xe. Nó hoạt động dựa trên nguyên lý cân bằng con lắc ngược. Đây là một hệ thống “under-actuated”, đa biến, phi tuyến và không ổn định [1, 2, 7]. Robot này đã được các công ty phát triển thành một phương tiện di chuyển thương mại có tên gọi là Segway với một số kiểu dáng phổ biến như H. 1. Tuy nhiên, hệ thống này vẫn đang được các nhà khoa học quan tâm nghiên cứu và thử nghiệm nhiều thuật toán điều khiển khác nhau từ tuyến tính đến phi tuyến lẫn các thuật toán điều khiển thông minh. Các thuật toán này chủ yếu tập trung vào việc điều khiển cân bằng, điều khiển vị trí và hướng của robot.



**H. 1** Một số kiểu dáng của Segway

Hai thuật toán điều khiển tuyến tính kinh điển được áp dụng trên robot hai bánh là LQR và PID. Hai thuật toán này đều cho kết quả tốt nhưng việc hiệu chỉnh các thông số rất khó khăn, nhất là bộ điều khiển PID [11]. Để khắc phục nhược điểm này, đã có nhiều nghiên cứu về bộ tự chỉnh thông số bộ điều khiển tuyến tính và cho kết quả rất khả quan [5,6,10].

Mặc dù cho kết quả như mong muốn, nhưng việc thể kế bộ điều khiển tuyến tính còn gặp một khó khăn đó là mô hình toán và thông số của robot này rất phức tạp, khó xác định. Bộ điều khiển mờ là bộ điều khiển thông minh được thiết kế chủ yếu dựa vào kinh nghiệm về đối tượng nên nó tỏ ra khá phù hợp khi áp dụng vào robot hai bánh. Samer Miasa và các cộng sự [12] đã thiết kế bộ điều khiển mờ và chạy thời gian thực trên chip vi điều khiển dsPIC30F2010 của hãng Microchip. Bộ điều khiển được thiết kế bằng công cụ Matlab/Simulink. Kết quả mô phỏng và thực nghiệm cho thấy robot đã cân bằng tốt. Tuy nhiên, nhóm tác giả chỉ thiết kế bộ điều khiển cân bằng góc nghiêng của robot. Tương tự, Cheng-Hao Huang và các cộng sự [3] đã thiết kế ba bộ điều khiển mờ để điều khiển cân bằng, vị trí và hướng của robot. Bộ điều khiển được thiết kế hoàn toàn bằng ngôn ngữ C và chạy thời gian thực trên chip FPGA. Kết quả mô phỏng và thực nghiệm cho thấy bộ điều khiển đã hoạt động ổn định và điều khiển tốt các thông số của robot. Tuy nhiên, việc thiết kế và thực thi bộ điều khiển khá phức tạp.

Việc thiết kế bộ điều khiển mờ tuy đã cho thấy nhiều ưu điểm so với thiết kế bộ điều khiển tuyến tính nhưng về chất lượng của đáp ứng chưa thể khẳng định được ưu điểm. Vì thế, việc so sánh đáp ứng của các bộ điều khiển này đã được các nhà nghiên cứu quan tâm. Ahmad Nor Kasruddin Nasir và các cộng sự [2] đưa ra sự so sánh về đáp ứng của hai bộ điều khiển mờ và PID. Dựa vào kết quả mô phỏng, nhóm tác giả đã khẳng định bộ điều khiển mờ cho kết quả tốt hơn bộ điều khiển PID. Bộ điều khiển mờ cho ra luật điều khiển hai động cơ không quá giới hạn, giảm vọt lố, rút ngắn thời gian tăng so với bộ điều khiển PID. Một sự so sánh cũng đã được Amir A. Bature và các cộng sự [7] công bố tại hội nghị quốc tế về cơ khí và cơ điện tử. Trong đó, nhóm tác giả tiến hành thực nghiệm và so sánh đáp ứng của bộ điều khiển mờ với bộ điều khiển PID và LQR thay vì mô phỏng. Kết quả so sánh cũng cho thấy bộ điều khiển mờ cho đáp ứng nhanh, độ vọt lố thấp hơn nhưng tiêu thụ nhiều năng lượng hơn hai bộ điều khiển còn lại.

Trong bài báo này, nhóm tác giả đưa ra đề nghị sử dụng bộ điều khiển tự chỉnh PID mờ để điều khiển robot để tận dụng được thế mạnh của bộ điều khiển

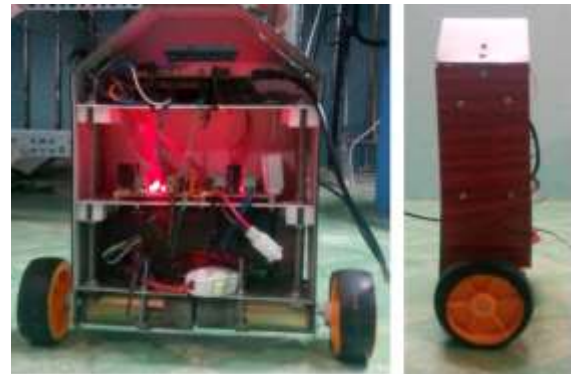
tuyến tính và bộ điều khiển thông minh. Bộ điều khiển đề nghị gồm hai vòng điều khiển. Vòng thứ nhất có cấu hình gồm hai nhánh đều sử dụng bộ điều khiển PD mờ. Nhánh thứ nhất sẽ tính toán góc tham chiếu cho bộ điều khiển cân bằng dựa vào vị trí tham chiếu để giúp robot bám được vị trí. Nhánh thứ hai sẽ cân bằng robot tại góc tham chiếu. Vòng điều khiển thứ hai là một bộ điều khiển PID tự chỉnh thông số điều khiển hướng của robot. Góc nghiêng của robot được đo bằng cảm biến tích hợp MPU6050 kết hợp với bộ lọc bù. Bộ điều khiển được chạy thời gian thực trên vi điều khiển 32-bit nhân ARM Cortex-M4 STM32F407VG của hãng STMicroelectronic. Để thiết kế và thực thi bộ điều khiển nhúng, nhóm tác giả phối hợp sử dụng Matlab/Simulink và hệ điều hành nhúng thời gian thực FreeRTOS.

Các phần còn lại của bài báo này có cấu trúc như sau. Phần 2 trình bày việc thiết kế mạch điều khiển, các bộ điều khiển và cách thực hiện bộ điều khiển nhúng. Phần 3 trình bày các kết quả thực nghiệm của bộ điều khiển đề nghị. Cuối cùng là phần 4 trình bày kết luận và hướng phát triển.

## 2. Nội dung chính

### 2.1 Thiết kế mạch điều khiển

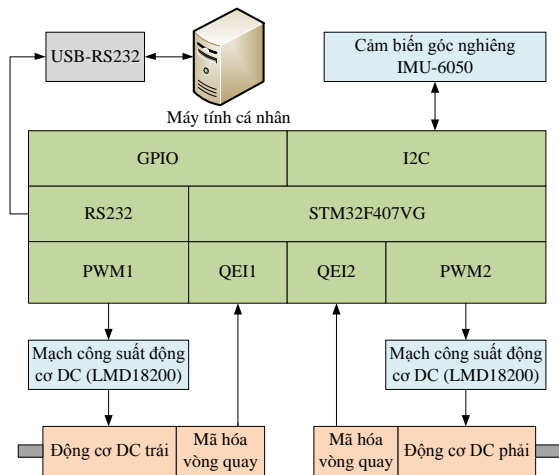
Mô hình robot hai bánh tự cân bằng được thiết kế như H. 2. Mô hình do nhóm tác giả thiết kế có khung rbot sử dụng chất liệu nhôm, hai động cơ sử dụng loại 24VDC tích hợp bộ mã hóa vòng quay tương đối 100 xung. Hai bộ mã hóa vòng quay này được sử dụng để đo vị trí và góc xoay của robot. Góc nghiêng được đo bằng cảm biến gia tốc tích hợp trên chip MPU-6050. Robot sử dụng hai nguồn pin Lipo 24VDC-3.5Ah cấp cho hai động cơ và mạch công suất, pin dự phòng điện thoại thông minh 5VDC-2.4Ah cấp nguồn cho mạch điều khiển chính STM32F4 Discovery và các mạch phụ trợ.



**H. 2** Mô hình robot hai bánh tự cân bằng

Bộ xử lý trung tâm sử dụng vi điều khiển STM32F407VG của hãng STMicroelectronic. Đây là vi điều khiển 32-bit, tích hợp bộ tính toán số thực dấu chấm động FPU, nhân tính toán DSP, tần số xung nhịp 168MHz, bộ nhớ chương trình 1Mb, bộ nhớ dữ liệu 192Kb. Ngoài ra, STM32F407VG còn có khả năng đọc được 6 bộ mã hóa vòng quay, các ngoại vi hỗ trợ DMA và nhiều ngoại vi hữu ích khác. Vì vậy, vi điều khiển này rất phù hợp để chạy thời gian thực

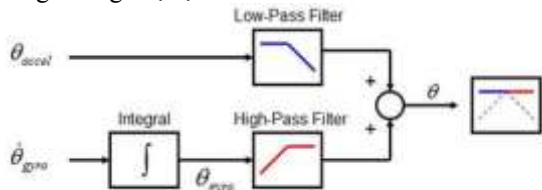
bộ điều khiển robot. Cấu trúc phần cứng điều khiển của robot như H. 3.



H. 3 Sơ đồ khối phần cứng điều khiển robot hai bánh tự cân bằng

### 2.2 Xử lý tín hiệu vào ra

Góc nghiêng robot (ký hiệu  $q$ ) được tính dựa vào giá trị trả về từ IMU. Bộ lọc bù [13] được sử dụng để ước lượng chính xác góc nghiêng của robot. Bộ lọc này được chọn vì cho kết quả ước lượng tốt, tính toán đơn giản phù hợp với việc chạy thời gian thực trên hệ thống nhúng. Bộ lọc có cấu trúc như H. 4.

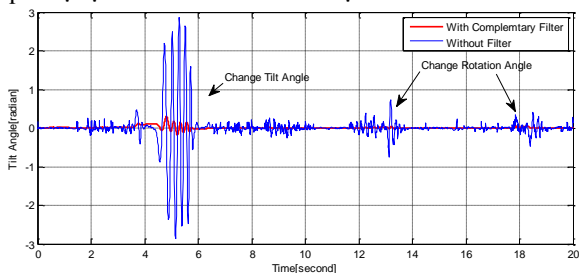


H. 4 Cấu trúc bộ lọc bù ước lượng góc nghiêng

Với tần số lấy mẫu 100Hz, thời hằng bộ lọc 0.5s, công thức ước lượng góc nghiêng của bộ lọc như sau:

$$q_{n+1} = 0.98(q_n + \dot{q}_{gyro}) + 0.02q_{accel} \quad (1)$$

Kết quả thí nghiệm H. 5 cho thấy góc nghiêng sao khi qua bộ lọc bù tốt hơn khi chưa lọc.



H. 5 So sánh giữa góc nghiêng đã lọc và chưa lọc. Biểu thức (2), (3) được sử dụng để tính vị trí và góc xoay của robot được tính dựa vào quãng đường di chuyển của hai bánh xe

[1]. Quãng đường này được tính toán dựa vào số xung trả về từ hai bộ mã hóa vòng quay.

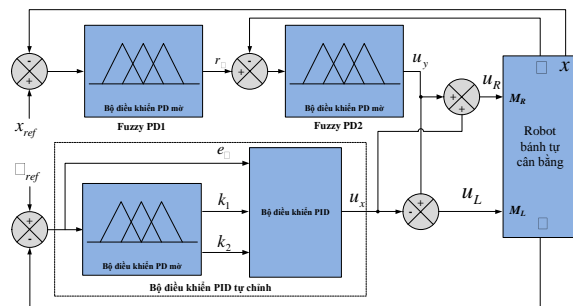
$$x_{robot} = \frac{x_{Left} + x_{Right}}{2} \quad (2)$$

$$j_{robot} = \frac{x_{Right} - x_{Left}}{D} \quad (3)$$

### 2.3 Thiết kế bộ điều khiển

#### 2.3.1 Cấu trúc bộ điều khiển

Bộ điều khiển robot được thiết kế như H. 6.

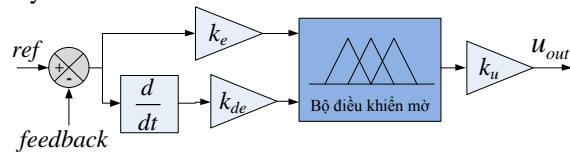


H. 6 Cấu trúc bộ điều khiển robot

#### 2.3.2 Bộ điều khiển cân bằng và bám vị trí

Bộ điều khiển này cân bằng robot đồng thời điều khiển bám nó theo vị trí đặt trước. Để thực hiện điều này, bộ điều khiển được chia thành hai bộ điều khiển PD mờ riêng biệt. Bộ điều khiển thứ nhất (Fuzzy PD1) nhận vào sai số vị trí để tính toán ra một góc tham chiếu hợp lý để xe có thể bám được vị trí tham chiếu. Bộ điều khiển thứ hai (Fuzzy PD2) nhận vào sai số góc nghiêng để tính toán ra điện áp điều khiển hai động cơ giúp xe bám góc nghiêng đặt trước. Giá trị tham chiếu của Fuzzy PD2 chính là ngõ ra của Fuzzy PD1.

Để đơn giản cho việc thiết kế, các bộ điều khiển mờ trong bài báo này đều có cấu trúc giống nhau như H. 7. Tuy nhiên, các hệ số vào/ra của các bộ điều khiển này khác nhau.



H. 7 Cấu trúc bộ điều khiển PD mờ

Bộ điều khiển mờ này có hai ngõ vào là sai số  $e(t)$  và vi phân sai số  $de(t)$  giữa giá trị đặt và giá trị thực và một ngõ ra  $u_{out}(t)$ . Hai hệ số  $k_e$  và  $k_{de}$  được sử dụng để điều chỉnh độ lớn của ngõ vào, hệ số  $k_u$  hiệu chỉnh tín hiệu điều khiển ngõ ra. Các hệ số này giúp hiệu chỉnh miền giá trị vào ra của bộ điều khiển cho phù hợp với đối tượng điều khiển. Việc thiết kế bộ điều khiển mờ được thực hiện qua bốn bước: xử lý vào ra, mờ hóa, suy diễn mờ và giải mờ [9]. Sau đây việc thiết kế Fuzzy PD1 và Fuzzy PD2 sẽ được trình bày chi tiết.

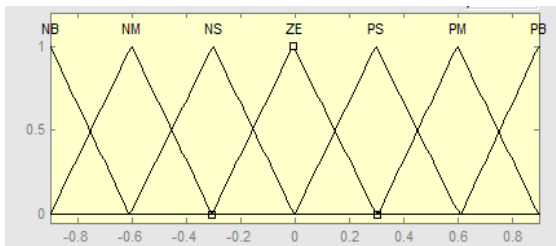
*Fuzzy PD1 - Bộ điều khiển mờ tính toán góc tham chiếu của robot:*

Như đã trình bày, bộ điều khiển này sẽ dựa vào sai số vị trí để tính toán ra một góc tham chiếu hợp lý để robot có thể bám được vị trí đặt. Bộ điều khiển mờ này có hai ngõ vào là sai số vị trí  $e_x(t)$  và tốc độ thay đổi sai số vị trí  $de_x(t)$ , một ngõ ra  $r_f(t)$  là góc nghiêng tham chiếu của robot. Ba biến ngôn ngữ được định nghĩa và đặt tên giống với các ngõ vào ra cho bộ điều

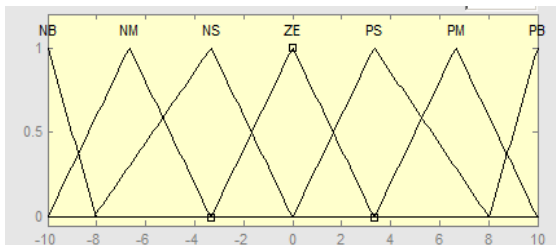
kiểm mờ. Các biến ngôn ngữ này có thể có các giá trị như sau:

- NB (Negative Big): âm lớn
- NM (Negative Medium): âm vừa
- NS (Negative Small): âm nhỏ
- ZE (Zero): bằng 0
- PS (Positive Small) : dương nhỏ
- PM (Positive Medium): dương vừa
- PB (Positive Big): dương lớn

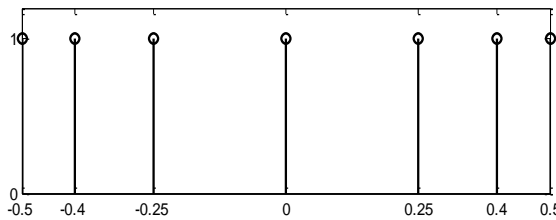
Hàm liên thuộc của của các biến ngôn ngữ này được định nghĩa như H. 8, 9, 10.



H. 8 Hàm liên thuộc lỗi vị trí



H. 9 Hàm liên thuộc tốc độ lỗi vị trí



H. 10 Hàm liên thuộc ngõ ra góc robot

Luật mờ xác định góc tham chiếu của robot dựa vào vị trí đặt như H. 11.

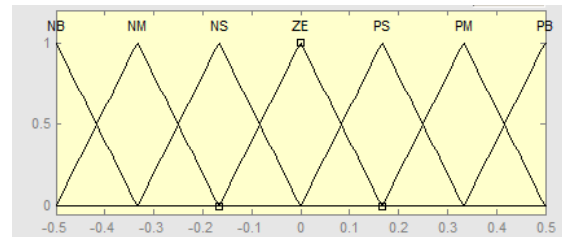
		Lỗi vị trí [ $\epsilon(t)$ ]						
		NB	NM	NS	ZE	PS	PM	PB
Tốc độ lỗi [ $\dot{\epsilon}(t)$ ]	NB	PB	PB	PB	PB	PM	PS	ZE
	NM	PB	PB	PB	PM	PS	ZE	NS
	NS	PB	PB	PM	PS	ZE	NS	NM
	ZE	PB	PM	PS	ZE	NS	NM	NB
	PS	PM	PS	ZE	NS	NM	NB	NB
	PM	PS	ZE	NS	NM	NB	NB	NB
	PB	ZE	NS	NM	NB	NB	NB	NB

H. 11 Luật mờ bộ điều khiển tính góc tham chiếu

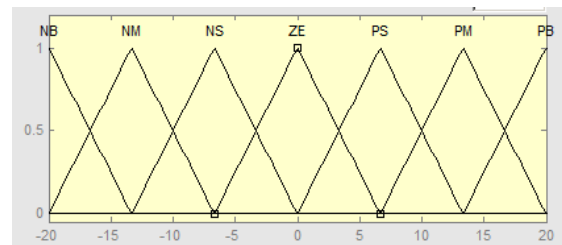
Bước cuối cùng trong việc thiết kế bộ điều khiển mờ là chọn luật hợp thành và phương pháp giải mờ. Bộ giải mờ đóng vai trò kết hợp ảnh hưởng của các luật điều khiển thành một giá trị rõ ở đầu ra. Trong bài báo này, luật hợp thành min-Prod và phương pháp giải mờ trọng tâm được sử dụng vì có công thức đơn giản, dễ thực hiện trên hệ thống nhúng.

Fuzzy PD2 - Bộ điều khiển cân bằng robot:

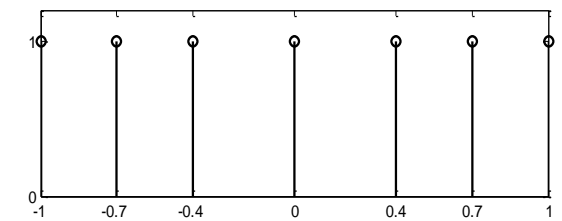
Bộ điều khiển này giữ vai trò ổn định robot tại góc tham chiếu được tính toán từ Fuzzy PD1, có cấu trúc như H. 7. Tương tự, bộ điều khiển này cũng có hai ngõ vào là sai số góc  $e_q(t)$  và tốc độ thay đổi sai số góc  $de_q(t)$ , một ngõ ra  $u_y(t)$  để tính phần trăm điện áp điều khiển hai động cơ. Các giá trị ngôn ngữ, cơ luật mờ, luật hợp thành và phương pháp giải mờ của bộ điều khiển này tương tự như bộ điều khiển Fuzzy PD1. Tuy nhiên, hàm liên thuộc của các biến ngôn ngữ có dạng khác nhau và như H. 12, 13, 14.



H. 12 Hàm liên thuộc lỗi góc



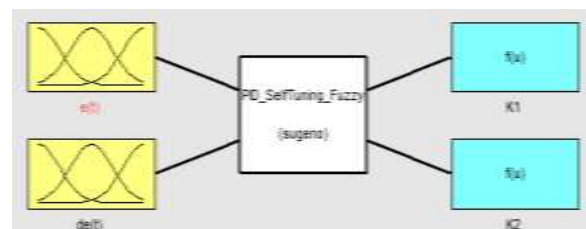
H. 13 Hàm liên thuộc tốc độ lỗi góc



H. 14 Hàm liên thuộc phần trăm điện áp điều khiển

### 2.3.3 Bộ điều khiển góc xoay của robot

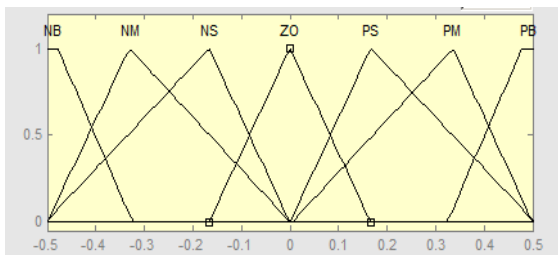
Đây là bộ điều khiển PID tự chỉnh có nhiệm vụ điều khiển robot bám theo góc xoay đặt trước. Để đơn giản trong việc thiết kế, các hệ số của bộ điều khiển PID cũng được tự chỉnh bằng một bộ điều khiển PD mờ có cấu trúc như H. 15. Bộ điều khiển này có hai ngõ vào là sai số góc xoay và tốc độ sai số góc xoay, hai ngõ ra  $k_1$  và  $k_2$  để tự chỉnh ba hệ số của bộ điều khiển PID. Các giá trị ngôn ngữ của hai ngõ ra của bộ điều khiển mờ này là: VVS-rất rất nhỏ, VS-rất nhỏ, S-nhỏ, M-vừa, B-lớn, VB-rất lớn, VVB-rất rất lớn.



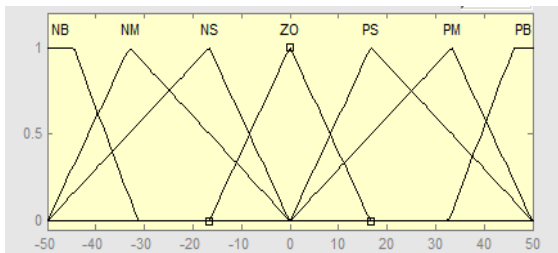
H. 15 Cấu trúc bộ điều khiển mờ tự chỉnh thông số cho bộ điều khiển PID



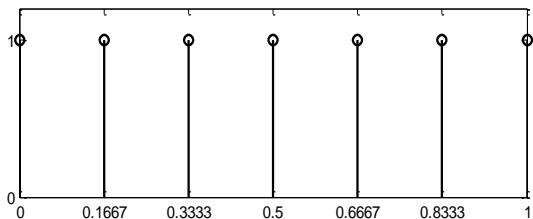
Hàm liên thuộc của các biến ngôn ngữ vào ra như H. 16, 17, 18.



H. 16 Hàm liên thuộc lỗi góc xoay



H. 17 Hàm liên thuộc tốc độ lỗi góc xoay



H. 18 Hàm liên thuộc ngõ ra  $k_1$  và  $k_2$

Luật suy diễn mờ xác định giá trị của hai hệ số  $k_1$  và  $k_2$  như H. 19, 20.

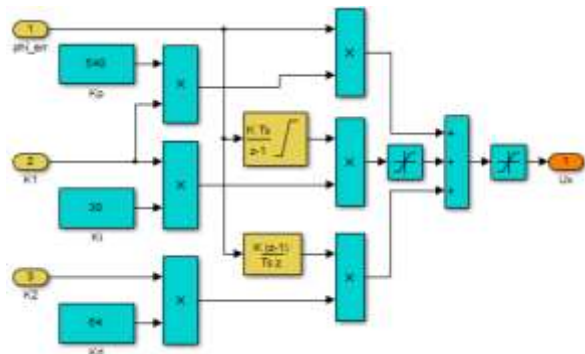
$K_1$	Lỗi góc xoay $[e(t)]$						
	NB	NM	NS	ZE	PS	PM	PB
NB	M	B	VB	VVB	VB	B	M
NM	S	M	B	VB	B	M	S
NS	VS	S	M	B	M	S	VS
ZE	VVS	VS	S	M	S	VS	VVS
PS	VS	S	M	B	M	S	VS
PM	S	M	B	VB	B	M	S
PB	M	B	VB	VVB	VB	B	M

H. 19 Luật mờ với ngõ ra  $k_1$

$K_2$	Lỗi góc xoay $[e(t)]$						
	NB	NM	NS	ZE	PS	PM	PB
NB	M	S	VS	VVS	VS	S	M
NM	B	M	S	VS	B	M	B
NS	VB	B	M	S	M	B	VB
ZE	VVB	VB	B	M	B	VB	VVB
PS	VB	B	M	S	M	B	VB
PM	B	M	S	VS	S	M	B
PB	M	S	VS	VVS	VS	S	M

H. 20 Luật mờ với ngõ ra  $k_2$

Hai hệ số  $k_1$ ,  $k_2$  được dùng để hiệu chỉnh các hệ số  $K_p$ ,  $K_I$  và  $K_D$  của bộ điều khiển PID điều khiển góc xoay robot như H. 21. Hệ số  $K_p$  và  $K_I$  cùng được hiệu chỉnh bởi giá trị  $k_1$ , hệ số  $k_2$  hiệu chỉnh  $K_D$ . Giá trị khởi động của các hệ số PID lần lượt là:  $K_D = 540$ ,  $K_I = 30$ ,  $K_D = 64$  đây chính là các hệ số của bộ điều khiển PID kinh điển giúp robot có thể bám tốt góc xoay khi đứng yên tại vị trí  $[0, 0, 0]$ .



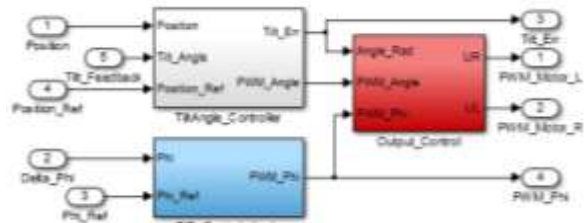
H. 21 Bộ điều khiển góc xoay của robot

Ngõ ra bộ điều khiển PID là  $u_x$  sẽ kết hợp với  $u_y$  để tạo ra phần trăm điện áp điều khiển hai động cơ trái và phải để robot có thể vừa cân bằng vừa bám vị trí và góc xoay đặt trước. Theo H. 6 ta có phần trăm điện áp điều khiển hai động cơ như biểu thức (4).

$$u_R = u_y + u_x \tag{4}$$

$$u_L = u_y - u_x$$

Bộ điều khiển hoàn chỉnh được thiết kế bằng Matlab/Simulink như H. 22.



H. 22 Bộ điều khiển robot hai bánh tự cân bằng.

### 2.3.4 Thực hiện bộ điều khiển nhúng

Để có thể chạy thời gian thực, bộ điều khiển đã thiết kế cần được chuyển thành bộ điều khiển dạng ngôn ngữ C. Bộ điều khiển này sẽ nhận tín hiệu hồi tiếp từ các mô-đun chương trình đọc góc nghiêng, vị trí và góc xoay để tính toán ra hai tín hiệu điều khiển  $u_x$  và  $u_y$ . Hai tín hiệu điều khiển này dưới dạng độ rộng của xung PWM để điều khiển hai động cơ qua mạch công suất. Các dữ liệu góc nghiêng, vị trí, góc xoay, các hệ số  $K_p$ ,  $K_I$ ,  $K_D$ , ngõ ra  $u_x$ ,  $u_y$  đồng thời cũng được gửi về máy tính PC để vẽ lên giao diện GUI để quan sát, hỗ trợ cho việc đánh giá đáp ứng và hiệu chỉnh các hệ số điều khiển.

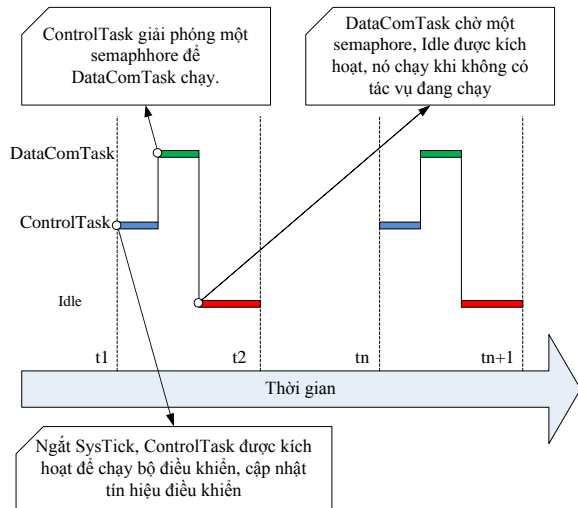
Để thực thi bộ điều khiển trên hệ thống nhúng, nhóm tác giả sử dụng hệ điều hành nhúng thời gian thực freeRTOS [8]. Đây là một hệ điều hành thời gian thực (RTOS) được thiết kế để hoạt động trên các hệ thống nhúng có tài nguyên bộ nhớ hạn chế. Hệ điều hành này có nhân viết bằng ngôn ngữ C có cấu trúc đơn giản. FreeRTOS được chọn để phát triển ứng dụng vì một số lý do như: mã nguồn mở, phát triển ứng dụng hướng tác vụ, được sử dụng rộng rãi và có thể hoạt động với nhiều nền tảng kiến trúc [4]. Với bộ điều khiển đã thiết kế, chương trình freeRTOS sẽ gồm hai tác vụ chính:

**ControlTask:** cập nhật các ngõ vào, chạy bộ điều khiển, cập nhật ngõ ra điều khiển động cơ, gửi dữ liệu sang tác vụ DataComTask. Tác vụ này được chạy

theo chu kỳ 10ms (đây cũng chính là thời gian lấy mẫu của bộ điều khiển).

**DataComTask:** Tác vụ này chờ nhận dữ liệu từ tác vụ ControlTask, gửi đến bộ đệm DMA của mô-đun USART sau đó kích hoạt để DMA điều khiển USART gửi dữ liệu về máy tính. Do đó, thời gian thực thi của tác vụ này rất nhỏ không ảnh hưởng đến tác vụ ControlTask.

Giản đồ thời gian thực thi các tác vụ của hệ thống như H. 23.



H. 23 Giản đồ thời gian thực thi các tác vụ của hệ thống

### 3. Kết quả thực nghiệm

Các thông số của bộ điều khiển như Bảng 1. Các thông số này được chọn bằng phương pháp thử - sửa sai lúc thực nghiệm.

Bảng 1.

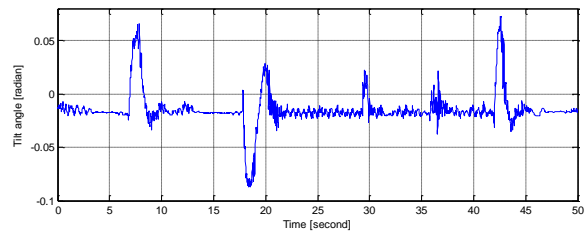
Bộ điều khiển	Thông số
Fuzzy PD1	$k_{ex} = 0.41, \quad k_{dex} = 7.3,$ $k_{ux} = 0.4$
Fuzzy PD2	$k_{eq} = 2.5, \quad k_{deq} = 1.8,$ $k_{uq} = 100$
PID tự chỉnh (giá trị mặc định)	$k_p = 540, \quad k_i = 30,$ $k_D = 64$

- Đáp ứng trường hợp robot cân bằng và không di chuyển ( $x_{ref} = 0[m], j_{ref} = 0[rad]$ )

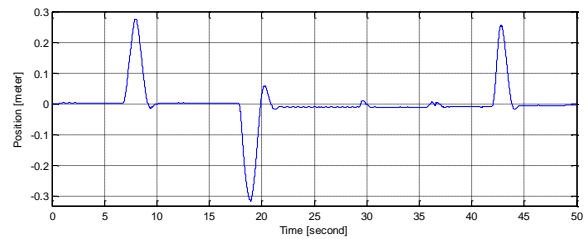
H. 24 là đáp ứng của robot đã thiết kế tại điểm cân bằng đồng thời chịu tác động của nhiễu từ bên ngoài làm thay đổi vị trí và góc xoay. Khi tác động nhiễu làm thay đổi vị trí (tại các thời điểm: 6.8[s], 17.8[s], 41.8[s]), bộ điều khiển đã tính toán góc tham chiếu hợp lý để đưa robot về vị trí tham chiếu, góc xoay thay đổi không quá 1.1[độ] khi nhiễu tác động. Khi tác động nhiễu làm góc xoay thay đổi (tại các thời điểm: 29.2[s] và 35.8[s]), bộ điều khiển đã đáp ứng lại với nhiễu và nhanh chóng đưa góc xoay về giá trị đặt. Khi tác động nhiễu, góc nghiêng lệch không quá 2.3[độ], vị trí gần như không đổi. Khi ổn định, góc nghiêng của robot thay đổi không quá 1[độ], sai số xác lập 1[độ]; vị trí của robot có sai số xác lập lớn

nhất 0.01[m]; góc xoay của robot có sai số xác lập 1[độ]. Từ đáp ứng cho thấy robot đã cân bằng tốt ngay cả khi có nhiễu tác động vào làm thay đổi vị trí và góc xoay.

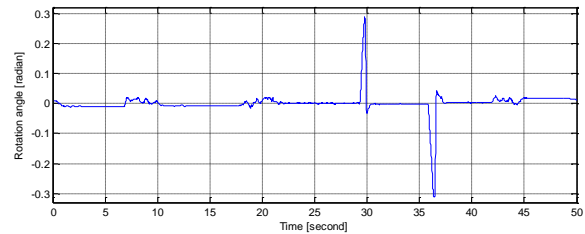
a)



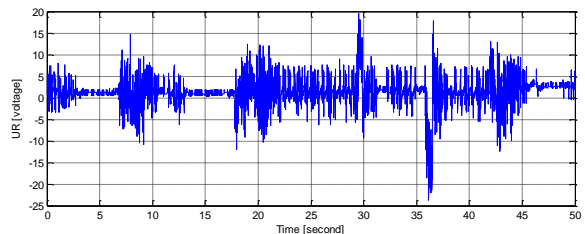
b)



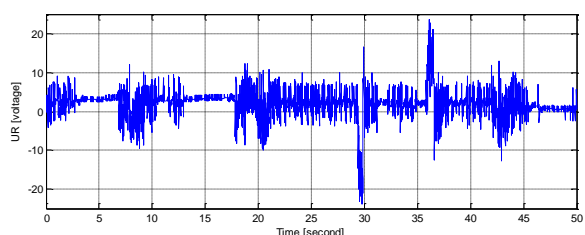
c)



d)



e)

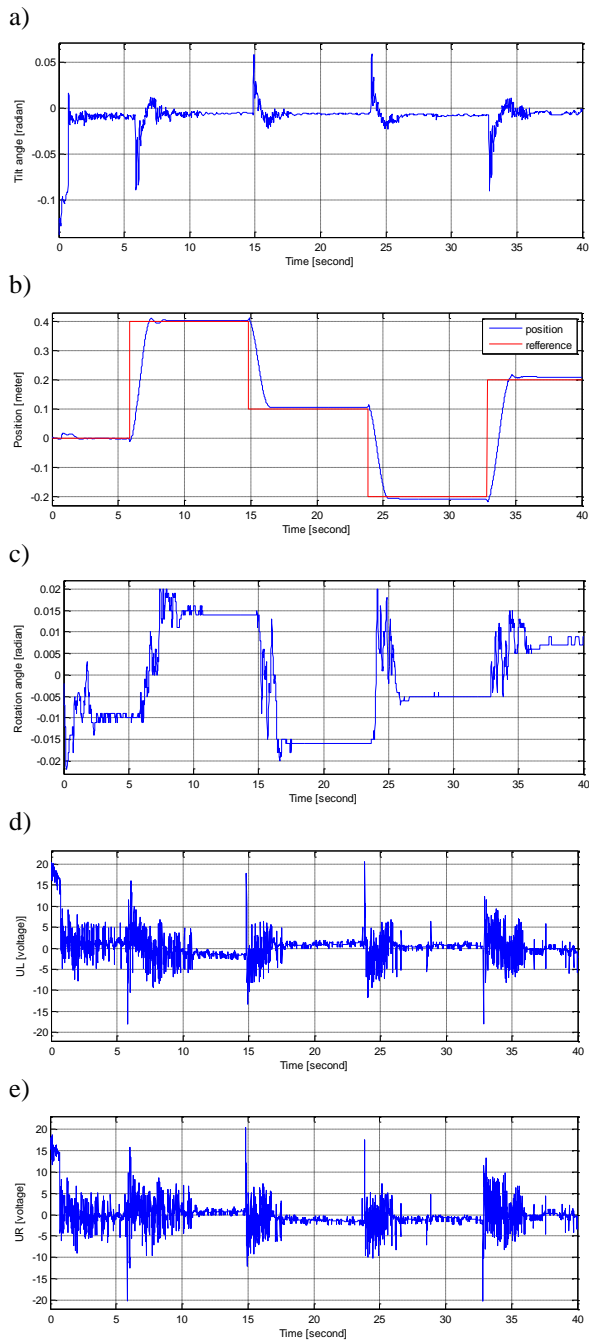


H. 24 Đáp ứng khi robot đứng yên và bị nhiễu tác động vào, a) góc nghiêng, b) vị trí, c) góc xoay, d) tín hiệu điều khiển động cơ phải, e) tín hiệu điều khiển động cơ trái

- Đáp ứng trường hợp robot di chuyển bám vị trí và hướng cố định ( $x_{ref}$  có giá trị lần lượt là 0, 0.4, 0.1, - 0.2 và 0.2 mét,  $j_{ref} = 0[rad]$ )

H. 25 là đáp ứng của robot khi thay đổi vị trí tham chiếu. Đáp ứng cho thấy robot đã bám tốt vị trí với sai số xác lập vị trí lớn nhất 0.01 [m], thời gian tăng khi vị trí thay đổi lớn nhất (0.4 [m]) là 1.5[s], độ vọt lố vị

trị tối đa 0.015[m]. Góc nghiêng của robot khi ổn định tại vị trí đặt có sai số xác lập 1[độ]. Góc xoay của robot thay đổi (khi vị trí đặt thay đổi) nhưng không vượt quá 1.15[độ]. Từ đáp ứng cho thấy robot hoạt động ổn định khi vị trí thay đổi liên tục.

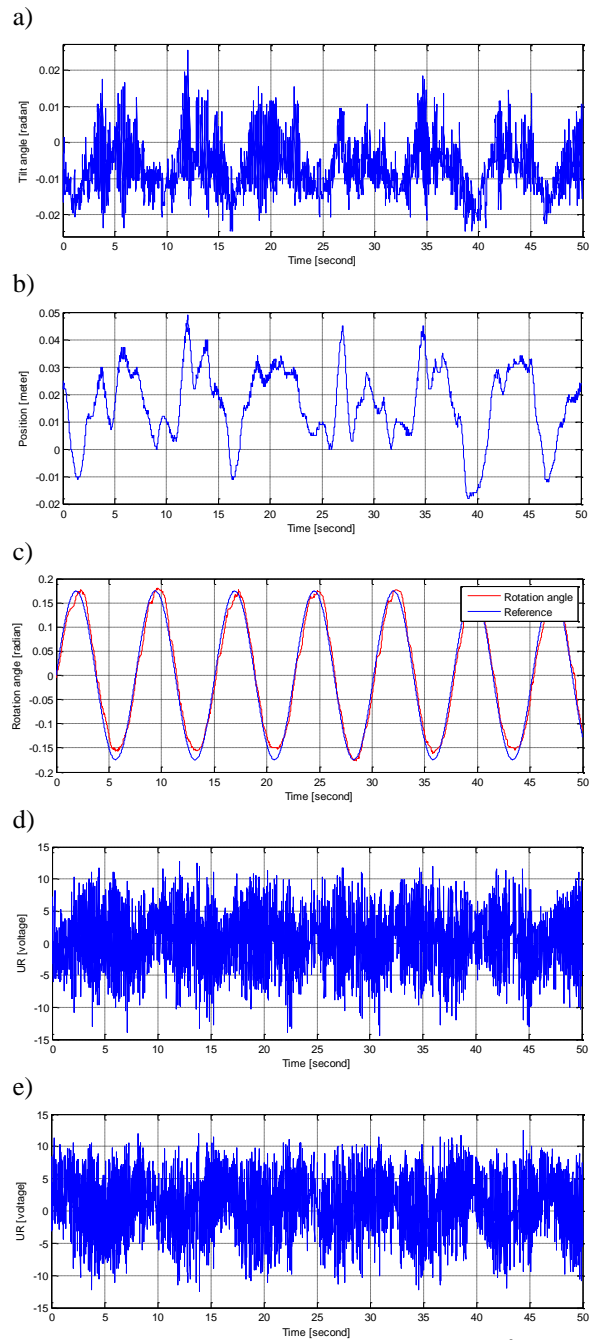


**H. 25** Đáp ứng khi robot bám vị trí đặt trước, a) góc nghiêng, b) vị trí, c) góc xoay, d) tín hiệu điều khiển động cơ phải, e) tín hiệu điều khiển động cơ trái

- Đáp ứng trường hợp robot cân bằng, không di chuyển ( $x_{ref} = 0[m]$ ), hướng bám theo giá trị đặt dạng sin biên độ 0.175[rad], tần số 0.13[Hz].

H. 26 là đáp ứng của robot khi góc xoay thay đổi theo dạng sin. Đáp ứng H. 26c cho thấy góc xoay robot bám tốt giá trị đặt. Góc nghiêng của robot thay đổi nhanh nhưng không quá 1.5[độ]. Vị trí cũng thay đổi

nhưng không quá 0.05[m]. Từ đáp ứng ta thấy robot vẫn hoạt động ổn định khi góc xoay thay đổi theo tham chiếu sin.



**H. 26** Đáp ứng khi robot bám góc xoay tham chiếu sin, a) góc nghiêng, b) vị trí, c) góc xoay, d) tín hiệu điều khiển động cơ phải, e) tín hiệu điều khiển động cơ trái

#### 4. Kết luận

Bài báo trình bày một phương pháp thiết kế và điều khiển robot hai bánh tự cân bằng. Bộ điều khiển đã thiết kế được chạy thời gian thực trên hệ thống nhúng. Kết quả thực nghiệm cho thấy bộ điều khiển thiết kế đáp ứng nhanh, robot cân bằng tốt với sai số xác lập trong các trường hợp nhỏ, bền vững với nhiễu tác động từ bên ngoài.

## Tài liệu tham khảo

- [1] Thao Ng.G.M, Nghia D.H, Phuc Ng.H, *A PID backstepping controller for two-wheeled self-balancing robot*, Proceeding of International Forum on Strategic Technology, Ulsan, Oct. 2010.
- [2] Nasir A.N.K., Ahmad M.A., Ghazali R., Pakheri N.S., Performance comparison between fuzzy logic controller (FLC) and PID controller for a highly nonlinear two-wheels balancing robot, The First International Conference on Informatics and Computational Intelligence, pp. 176-181, Dec. 2011.
- [3] Huang C.H, Wang W.J, *Design and implementation of fuzzy control on a two-wheel inverted pendulum*, IEEE Transactions on Industrial Electronics, vol. 58, no. 7, pp. 2988-3001, July 2011.
- [4] Pinto S., Pereira J., Oliveira D., Alves F., Qaralleh E., Ekpanyapong M., Cabral J., Tavares A., *Porting SLOTH system to freeRTOS running on ARM Cortex-M3*, International Symposium on Industrial Electronics, pp.1888-1893, Istanbul, Turkey, June 2014.
- [5] Qiu C., Huang Y., *The design of fuzzy adaptive PID controller of two-wheeled self-balancing robot*, International Journal of Information and Electronics Engineering, vol. 5, no. 3, pp. 193-197, May 2015.
- [6] Fang J., The research on the application of fuzzy immune PD algorithm in the two-wheeled and self-balancing robot system, International Journal of Control and Automation, vol. 7, no. 10, pp. 109-118, Oct. 2014.
- [7] Amir A.B, Salinda Buyamin, Mohamed N.A, Mustapha Muhammad, *A comparison of controller for balancing two wheeled inverted pendulum robot*, International Journal of Mechanical and Mechatronics Engineering, vol. 14, no. 03, pp. 62-68, June 2014.
- [8] FreeRTOS home page, [Online], Available: <http://freertos.org/> access on July 19, 2015.
- [9] Wu J., Zhang W., Wang S., *A two-wheeled self-balancing robot with the Fuzzy PD controller method*, Mathematical Problems in Engineering Research Article, Volume 2012, Article ID 469491, 13 pages.
- [10] Ren T.J., Chen T.C., Chen C.J., *Motion control for a two-wheeled vehicle using a self-tuning PID controller*, Control Engineering Practice, vol. 16, pp. 365-375, Mar. 2008.
- [11] Muhammad T. A., *Large Signal Analysis of the Mach-Zender Modulator with Variable BIAS*, Proceeding of Natl. Sci. Council. ROC(A), vol.25, no. 4, pp. 254-258, 2001.
- [12] Miasa S., Al-Mjali M., Al-Haj Ibrahim A., Tutunji T.A., *Fuzzy control of a two-wheel balancing robot using DSPIC*, The 7<sup>th</sup> International Multi-Conference on Systems

Signals and Devices, pp. 1-6, Amman, Jordan, June 2010.

- [13] Higgins W.T., *A comparison of Complementary and Kalman Filtering*, IEEE Transactions on Aerospace and Electronic Systems, vol. AES-11, no. 3, Feb. 2007.

[14]



**Nguyễn Văn Khanh** nhận bằng kỹ sư Điện tử tại trường Đại học Cần Thơ năm 2005, bằng thạc sỹ Tự động hóa tại trường Đại học Bách Khoa TP. HCM năm 2014. Thạc sỹ Nguyễn Văn Khanh tham gia giảng dạy tại trường Đại học Cần Thơ từ năm 2006 đến nay. Hiện anh đang là Giảng viên thuộc bộ môn Tự động hóa, khoa Công Nghệ.



**Trần Thanh Hùng** nhận bằng kỹ sư Điện tử tại trường Đại học Cần Thơ năm 1996, bằng thạc sỹ Kỹ thuật Điện tử tại trường Đại học Bách Khoa TP. HCM năm 2000, và nhận bằng tiến sỹ Kỹ thuật tại trường Đại học Kỹ thuật Sydney, Úc năm 2008.

Tiến sỹ Trần Thanh Hùng tham gia giảng dạy tại trường Đại học Cần Thơ từ năm 1996 đến nay. Hiện anh đang là Giảng viên chính thuộc bộ môn Tự động hóa, khoa Công Nghệ; giữ chức vụ Phó trưởng khoa Công Nghệ, trường Đại học Cần Thơ.