ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)
Vol 4, Issue 2, February 2018

A new Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller

[1] S.Manikandan, [2] Reshma, [3] G Vani prasanna, [4] Ch Swetha, [5] V S Hariharan
[1] Professor, [2][3][4] Student, [5] Principal
Dept of EEE, Balaji Institute of Technology and Science, Warangal, Telangana

***Abstract*: -- In the last decade, the open source community has expanded to make it possible for people to build complex products at home. [1] In this thesis a two wheeled self-balancing robot has been designed. These types of robots can be based on the physical problem of an inverted pendulum [2]. In this paper, we can see the design, construction and control of a two-wheel self-balancing robot. This system consists of a pair of DC motor and an Arduino UNO R3 microcontroller board, make a robot which can balance itself on two wheels the platform will not remain stable itself. Our job will be to balance the platform using distance sensors and to maintain it horizontally. At first, we have decided to just balance the robot on its two wheels.**

***Keywords:* — Robot, Arduino, AT mega 328, Control Systems, PID controller, Linear Quadratic Regulator.**

I. INTRODUCTION

To make a robot which can balance itself on two wheels. There will be only one axle connecting the two wheels and a platform will be mounted on that .There will be a another platform above it. The platform will not remain stable itself. Our job will be to balance the platform using distance sensors and to maintain it horizontal. At first we have decided to just balance the robot on its two wheels. Basically, a two-wheel self-balancing robot is very similar to the inverted pendulum, and which is an important test part in control system and research education purpose; let us see, for an example [5], [6]. A Two-wheeled self-balancing vehicle commercially known as "Segway".And also the Segway can never stay upright.Besides the development of Segway, studies of two-wheel self-balancing robots have been widely reported. For example, JOE [5] and nBot [6] are both early versions complete with inertia sensors and motor encoders and also along with on-vehicle microcontrollers. Arduino is an open prototyping platform based up on Atmega processors .And It will be a fast becoming popular platform for both education [7] and product development, with applications ranging from robotics [8], [9] to process control [10], [11] and networked control [12].In this paper, we report a student project on the basis of design, construction and control of a two-wheel self-balancing robot with Ardunio software. The robot is driven by two DC motors, and is equipped with an Arduino Uno board which is based on the ATmega328 microprocessor, and also including with a single-axis gyroscope and a 2-axis accelerometer for attitude determination. In order to compensate for gyro drifts common in COTS sensors, a complementary filter is implemented [13]. For two wheel control designs are based up on the linearized equations of motion, such as a proportional-integral-differential (PID) control.

This paper is arranged as follows: The Section 2 describes the hardware and system architecture of the robot. The designs of filters, inner control loop to equilibrate the two motors,and balancing control in Section 3. Finally Section 4 represents the experimental results, followed by some conclusion in Section 5.

II. STRUCTURE OF THE TWO-WHEEL BALANCING ROBOT

In the structure of a self-balancing robot can be classified into three parts such as sensors, motor and motor control, and develop board [4], [5]. In Section II-A introduces the application and advantage of the sensors on the proposed balancing robot, andalso how these sensors are employed to obtain measurements of acceleration, and distance traveled,androbot tilt angle. The Section II-B describes the motor selection and control for the balancing robot. And Section II-C discusses the reason behind choosing the Arduino develop board, and how it is deployed.

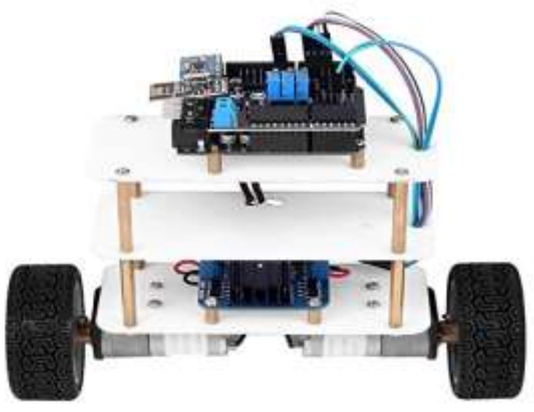


Fig 1: Self Balancing Robot

A. Selection and Application of Sensors

In order to make balance of the robot, there are two things must needed one is robot's tilt angle and angular velocity. It is possible by using MEMS sensors. And it may also including a gyroscope, wheel-angle encoders, and alsoan accelerometer and we can measure all the data needed for balancing control.

1) Gyroscope: Gyroscope measures the angular rate around an axes. Tilt angle can be obtained by integrating angular rate over sampled time. An estimate of angular displacement is obtained by integrating velocity signal over time.

2)Accelerometer: This meter is used to measures the total external acceleration of the balancing robot, and also which includes the gravitational and motion accelerations. Accelerometer can measure the force of gravity and with that information, the angle of robot can be obtained. Kalman filter is used for the fusion of outputs of two sensors.

3) Encoders: In generally the encoders return the rotation angles of individual motor shafts as digital signals, which are sent to the processor. Next, after conversion based on gear ratio and wheel radius, the distance traveled can be calculated. The encoder chosen for this project is the 64 cycle-per-revolution, which provides a resolution of 64 counts per revolution of the motor shaft.

B. Motor Control Board

One of the most important part in balancing robot is Dc Motor. It emphasizes torque output instead of velocity, because it has to oppose the rotational moment that gravity applies on the robot.Hence, the motors need to provide enough torque to correct the robot's body pose back to a balanced position.

Generally, Each motor provides a torque of 200 oz-in, sufficient for our purpose. The maximum operating current and voltage ratings are givenas 14 Ampere and 16 Volts.

C. Arduino Mega Development Board

Selection of the development board is based on the following features:

1) Performance: Normally, the self-balancing robot needs almost real-time response to estimate and correct its tilt angle.Hence, the development of board must provide a processing speed that is sufficiently fast to perform the processing tasks, including data acquisition, control computation and signal output, within the sampling time.The Arduino Mega development board is equipped with the ATmega328 processor, which features a maximum clock rate of 16 MHz.

2) Input / Output Pins:On the robot, sensors are deployed to obtain measurements of its motion: a gyroscope and an accelerometer are used to estimate the tilt angle,encoders are used to obtained the measurements of the robot. The Arduino UNO features 54 general-purpose digital I/O pins of which 15 provide PWM output. And alsosome of the digital I/O pins also support serial communication such as I2C and SPI, as well as interrupt handles. The board also features sixteen 10-bit analog inputs for 0-5V input, giving a quantization limit of 4.9mV.

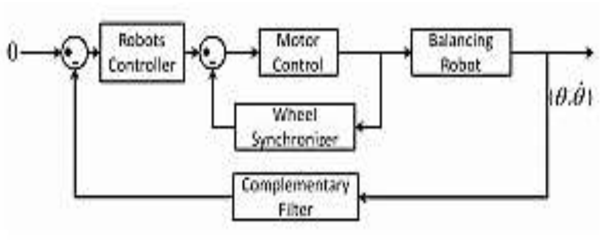


Fig. 2. Architecture of the controlled system

3) Price and Expansions: As we all know Arduino boards are low-cost and expandable, where optional peripherals called shields can be purchased as and when needed.

D. Power Supply : In generally power supply, the motors a voltage between 12V to 16V, and for the development voltage between 5V to 15V.These analog voltages lie in therange from 0V to 5V.

III. ANGLE ESTIMATION AND BALANCING CONTROL

The fig 2 represents the system architecture of the self-balancing robot. A balancing controller which delivers a motor-control signal, and a inner-loop controller for wheel synchronization. And also a Feedback is provided through a complementary filter. The filter function is to provide an estimate of the robot tilt angle from gyroscope and accelerometer measurements.

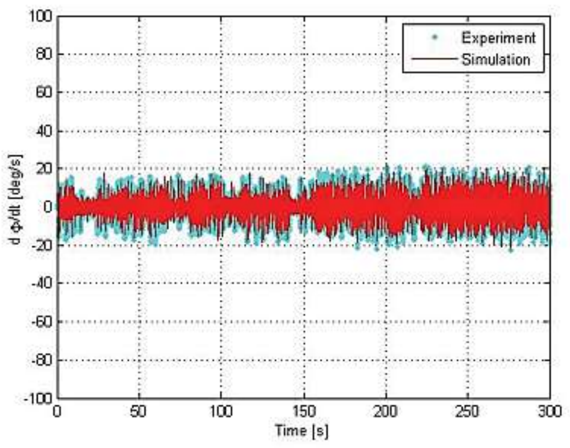


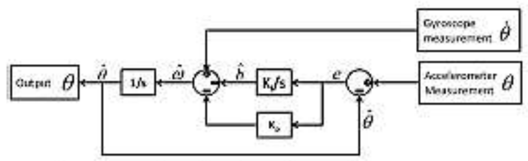
Fig 3: Tilt velocity of the robot body

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)
Vol 4, Issue 2, February 2018

A. Angle Estimation via Complementary Filter

In Section II, a gyroscope is used to measure the angular velocity of the robot. And accelerometer measures the Y - and Z-components of gravitationalacceleration, encoders measure the distancetravelled by the wheels.For thebalacning control, the tilt angle is corrected to the upright position. Balancing control is possible by using wheel motion of the robot. In order to solve the problem of angle measurement, we can design a complementary filter [13], a block diagram of which is as shown in Fig. 3.



θ = measured angle of accelerometer
$\dot{\theta}$ = measured angular velocity of gyroscope
$\hat{\theta}$ = estimated angle of balancing robot
e = estimation error between $\hat{\theta}$ and θ measurement
$\hat{b}$ = estimated gyro bias
$\hat{\omega}$ = estimated angular velocity
k_p = proportional gain
k_i = integral gain

Fig. 4. Block Diagram of the Complementary Filter

Next tuning the filter, its performance under noise isfirst tested in simulation as shown in Fig. 4, whichcompares the three ways to estimate the tilt angle. As we know, the best estimate of the angle is provided by the complementary filter. Here, we can see the same result is also observed in experiment, as shown in Fig.5

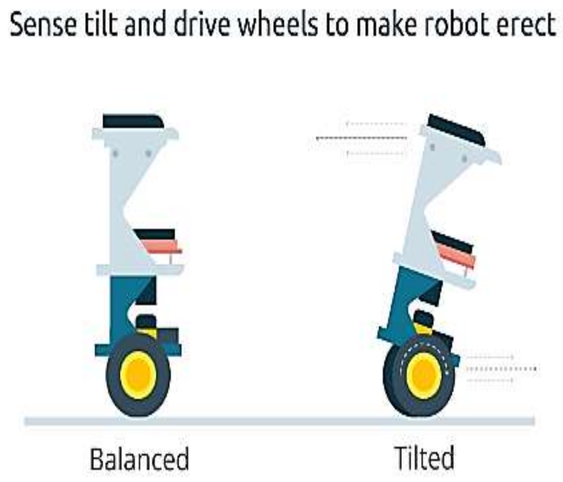
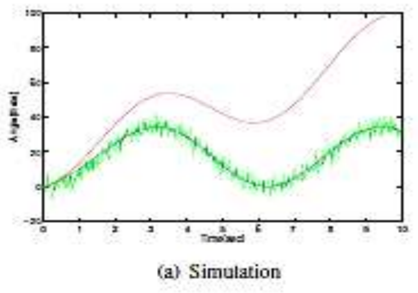


Fig 5: Sense tilt and drive wheels to make robot erect



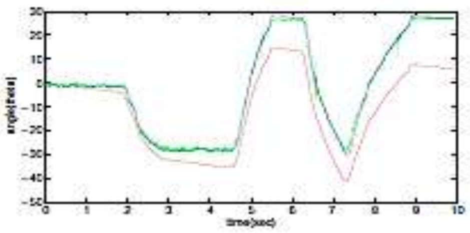


Fig. 6. Tilt angle estimation.

Here Blue line represents: output value of complementaryfilter; red line represents: direct integration of gyroscope angular rate; green line represents: angle deduced from accelerometer measurement using the direction cosinemethod.

B. Synchronization of a Inner-loop Control for Wheel

If we can assume ideally, if the same motor control signal is sent to the motors control board, both motors should rotate at the same speed. A method is used to synchronize both motors is needed.

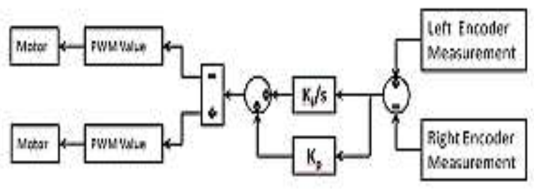


Fig. 7. Block diagram of wheel synchronizer.

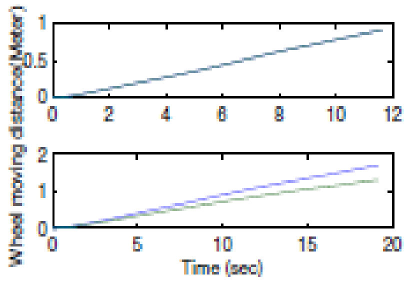


Fig. 8. Test results of wheel travel distances.

Top: The wheels'travel distances coincide when wheel synchronization control is employed.
Bottom: At equal motor input, the wheels travel at different speed withwithout wheel synchronization.

C. Balancing Control Designs
Here In this project, two control design strategies are studied one is proportional-integral-differential (PID) control, and proportional-integral proportional-differential (PI-PD) control [6] based on linear-quadratic-regulator (LQR) design.

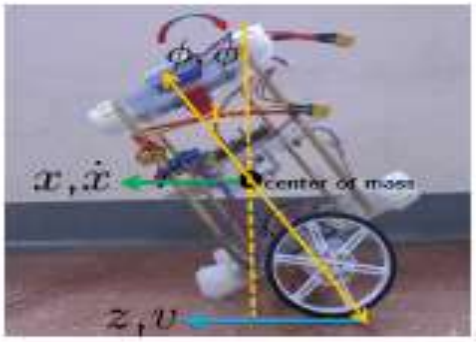


Fig. 9. Definition of motion variables

1) PID Control:

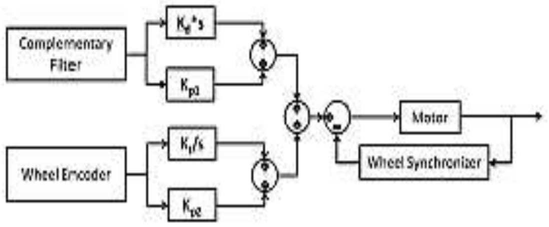


Fig. 10. Block diagram of PI-PD controller
The block diagram of the PID controller is as shown in Fig. 10, where the complementary filter generates estimates of the angular velocity and angle.
Proportional gain:
This gain directly relates to the current error to the current motor output. This forms the bulk of the gain in the controller and reacts instantly to any error
Integral gain:
This gain compensates for any steady state error caused by drift or the offsets in the center of mass of the robot. It works by integrating the values of error over a period of time and hence reacts to steady state error after it accumulates.
Derivative gain
This gain uses the derivative of the error to reduce the overshoot of the output erivative gain This gain uses the derivative of the error to reduce the overshoot of the output. erivative gain This gain uses the derivative of the error to reduce the overshoot of the output.

Derivative gain:
The gain is uses the derivative of the error to reduce the overshoot of the output. The three gains, kP, kI and kD are tunable parameters and must be adjusted according to the specific hardware used. There are very many ways to tune these parameters to an optimal value. We used the manual method to do this. First, we set all the gains to 0. Then kP is increased until the response oscillates. Then kP is set to half its value. kI is now increased until the steady state error is corrected in sufficient time. Finally, kD is increased to minimize overshoot.
2) LQR, Linear Quadratic Regulator

To control a linear system ,LQR-control can be applied. The states of the dynamical system are described by a state space model including the matrices A, B and C as seen in Figure 11:

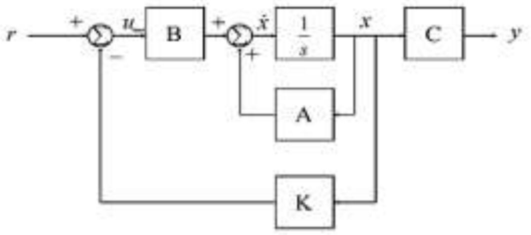


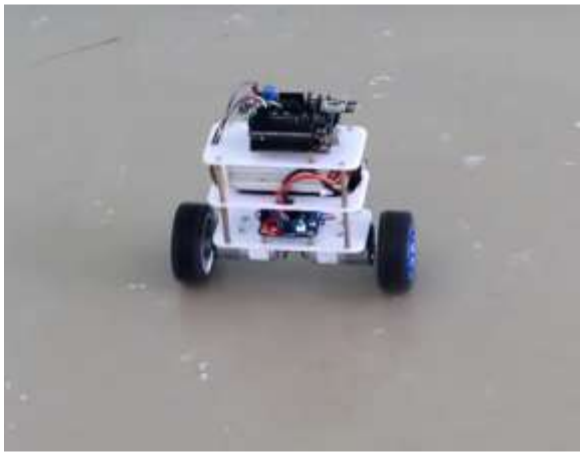
Fig 11: Block representation of state space system showing the system matrices A, B and C as well as the gain matrix K.

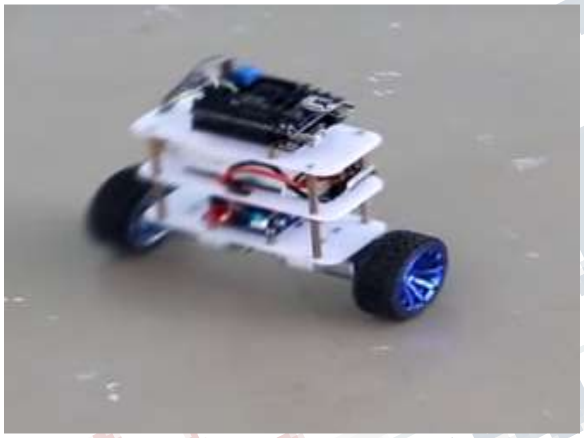
The Design of the Self-Balancing Robot LQR Controller. The LQR method is the most mature controller design method in the development of modern control theory [15]; LQR optimal control is to seek the control amount $u*(t)$ to make the system reach the steady state and guarantee the performance index J to take the minimum value:
$J = \int \infty 0 \, (XTQX + uTRu) \, dt$
The parameter of the robot is shown in below table:

Symbol	Actual value
k_m	0.0136 Nm/A
k_e	0.01375 V/(rad/s)
R	1.6 Ω
M_p	0.52 kg
M_w	0.02 kg
l	0.16 m
I_p	0.0038 kg·m^2
g	9.8 m/s^2
I_w	0.0032 kg·m^2
r	0.025 m

IV. EXPERIMENTAL RESULTS

The figure shown below shows the working of the Self Balancing Robot with Ardunio Microcontroller





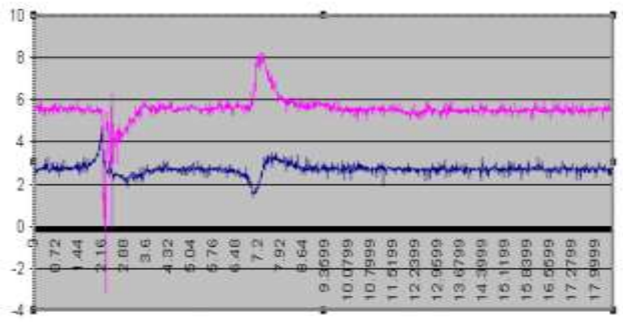


Fig 11: Individual outputs of accelerometer and gyro

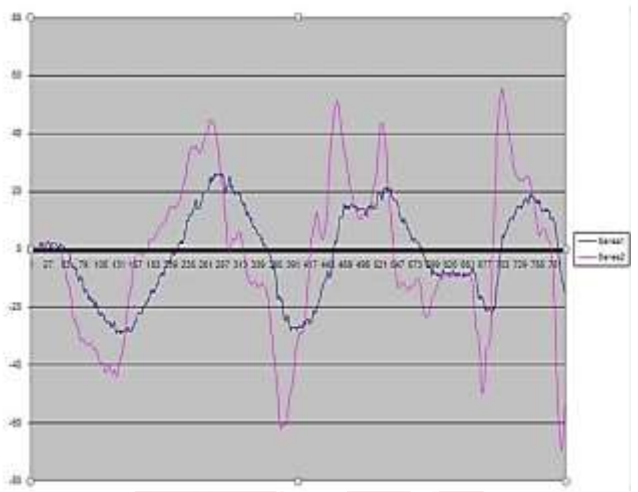


Fig 12: Combined outputs Series1-gravity angle(only accelerometer reading) Series2-stabilized angles(combined readings)

V. CONCLUSION

This paper presents an experimental, Arduino based, low cost self-balancing robot as an educational control system.Low-cost communicate hardware such Xbee may be added to the robot, so that the user can control its motion remotely and read the data via telemetry. Also, user-controlled motion such as forward, backward, turn right and left, may be implemented.A possible extension is to combine the Arduino system and other sensors such ultrasonic and IR senses, GPS, digital compass and Camera to address other advanced applications, e.g. obstacle avoidance and perimeter following. The main drawback that hampers the overall project result its unable to attain. Therefore appropriate conclusions are not able to achieve. The problem with the oscillation still remains with the system and future work has to be done to achieve a stable solution.

VI. FUTURE IMPROVEMENT

The stabilization provided by the reaction wheel is limited be the torque provided by the reaction wheel motor. Subsequent plan is to use a rotating disc and its gyroscopic precession for balancing. This would provide a more stable design capable of providing higher restoring torque .In such a case particular attention should be paid to any rotary axes, their alignment, and how they are fixed to the model, to the position and alignment of brackets, and to the mounting and fastening of any flexible couplings. In addition to this, fuzzy logic controller can also be implemented to provide flexibility and accuracy in control.

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)
Vol 4, Issue 2, February 2018

REFERENCES

[1] Arduino. [Online]. Available: http://arduino.cc

[2] D. Arndt, J. E. Bobrow, S. Peters, K. Iagnemma, and S. Bubowsky, "Two-wheel self-balancing of a four-wheeled vehicle," IEEE Control Systems Magazine, vol. 31, no. 2, pp. 29–37, April 2011.

[3] R. Fierro, F. Lewis, and A. Lowe, "Hybrid control for a class of underactuated mechanical systems," IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 29, no. 6, pp. 649–4, nov 1999.

[4] K. Xu and X.-D.Duan, "Comparative study of control methods of single-rotational inverted pendulum," in Proceedings of the First International Conference on Machine Learning and Cybernetics, vol. 2, 2002, pp. 776–8.

[5] F. Grasser, A. D'Arrrigo, S. Colombi, and A. C. Rufer, "JOE: A mobile, inverted pendulum," IEEE Transactions on Industrial Electronics, vol. 49, no. 1, pp. 107–14, 2002.
[6] D. P. Anderson. (2003, Aug.)nBot balancing robot. Online.[Online]. Available: http://www.geology.smu.edu/ dpa-www/robo/nbot

[7] J. Sarik and I. Kymissis, "Lab kits using the Arduino prototyping platform," in IEEE Frontiers in Education Conference, 2010, pp. T3C– 1–5.

[8] G. Guo and W. Yue, "Autonomous platoon control allowing rangelimited sensors," IEEE Trans. on Vehicular Technology, vol. 61, no. 7, pp. 2901–2912, 2012.

[9] P. A. Vignesh and G. Vignesh, "Relocating vehicles to avoid traffic collision through wireless sensor networks," in 4th International Conference on Computational Intelligence, Communication Systems and Networks. IEEE, 2012.

[10] M. J. A. Arizaga, J. de la Calleja, R. Hernandez, and A. Benitez, "Automatic control for laboratory sterilization process rsd on Arduino hardware," in 22nd International Conference on Electrical Communications and Computers, 130-3, Ed., 2012.

[11] S. Krivic, M. Hujdur, A. Mrzic, and S. Konjicija, "Design and implementation of fuzzy controller on embedded computer for water level control," in MIPRO, Opatija, Croatia, May 21–25 2012, pp. 1747–51.

[12] V. Georgitzikis and I. Akribopoulos, O.andChatzigiannakis, "Controlling physical objects via the internet using the Arduino platform over 802.15.4 networks," IEEE Latin America Transactions, vol. 10, no. 3, pp. 1686–9, 2012.

[13] A.-J. Baerveldt and R. Klang, "A low-cost and low-weight attitude estimation system for an autonomous helicopter," in IEEE International Conference on Intelligent Engineering Systems, sep 1997, pp. 391–5.

[14] R. C. Ooi, "Balancing a two-wheeled autonomous robot," Final Year Thesis, The University of Western Australia, School of Mechanical Engineering, 2003.

[15] L. Cheng, H. Dewen, P. Yaodong, Z. Xiaocai, and D. Guohua, "Fuzzy control of a quintuple inverted pendulum with the LQR method and 2-ary fuzzy piecewise interpolation function," in Proceedings of the 45th IEEE Conference on Decision and Control (CDC '06), pp. 6307–6312, December 2006.