Global Journal of Researches in Engineering: J
General Engineering
Volume 19 Issue 1 Version 1.0 Year 2019
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

Control Unit for a Two-Wheel Self-Balancing Robot

By D.P.V.J. Jayakody & K.P.G.C. Sucharitharathna

Abstract- A self-balancing personal transporter which is based on the inverted pendulum concept has sufficient potential to provide solutions for the upcoming global issues in the transportation industry. However due to the expensive price range which the self-balancing scooters are introduced at and few safety issues, this concept has failed in reaching the hands and becoming popular among the majority of the society. Therefore this research paper consists of a comprehensive literature review on the existing models of the self-balancing transporter scooters, possible ways to reduce the initial cost of implementing a control unit for self-balancing transporter vehicles and methods to address the issues which generate along with the proposed cost-reduction methods. Real-time comparison of Kalman and Complementary filtering processes are performed to sort out the optimum algorithm to estimate the true angle of the inclination of the self-balancing prototype. Similarly several forms of control system implementation are compared through simulations and real-time experiments to obtain the ideal motor response for variations in the position of the prototype.

Keywords: inverted-pendulum, PID control, self-balancing robot, complementary filter, kalman filter, MPU6050.

GJRE-J Classification: FOR Code: 091599

Control Unit for a Two-Wheel Self-Balancing Robot

D.P.V.J. Jayakody $^{\alpha}$ & K.P.G.C. Sucharitharathna $^{\sigma}$

Abstract- A self-balancing personal transporter which is based on the inverted pendulum concept has sufficient potential to provide solutions for the upcoming global issues in the transportation industry. However due to the expensive price range which the self-balancing scooters are introduced at and few safety issues, this concept has failed in reaching the hands and becoming popular among the majority of the society. Therefore this research paper consists of a comprehensive literature review on the existing models of the self-balancing transporter scooters, possible ways to reduce the initial cost of implementing a control unit for self-balancing transporter vehicles and methods to address the issues which generate along with the proposed cost-reduction methods. Real-time comparison of Kalman and Complementary filtering processes are performed to sort out the optimum algorithm to estimate the true angle of the inclination of the self-balancing prototype. Similarly several forms of control system implementation are compared through simulations and real-time experiments to obtain the ideal motor response for variations in the position of the prototype.

Keywords: inverted-pendulum, PID control, self-balancing robot, complementary filter, kalman filter, MPU6050.

I. Introduction

In today's society transportation is undoubtedly a fast-growing industry. Due to the rapid growth in the demand for personal transporter vehicles, self-balancing personal transporter scooters were introduced by the Segway Company. For the intention of increasing the efficiency of humans and to reduce the cost, the self-balancing personal transporter which is also a great representation of the personal mobility device concept is now widely used in many industries and institutions such as police departments, tourism industry, factories, and airports. The benefits which are offered by this personal transporter vehicle such as higher accessibility and zero fuel consumption can be considered as the ultimate solutions for the upcoming global issues caused by the growth of traffic and the environmental pollution happening all around the world. Even though the self-balancing transporter represents a better version of the personal transporter type vehicles that are being used nowadays, it simply failed in reaching the hands of the majority of society due to the expensive price range and the safety issues pointed out by the existing users of these self-balancing transporter models. The self-balancing personal transporter models (mainly Segway models) are comprised of multiple gyroscope and accelerometer sensors (few as additional) to obtain the angular rate and acceleration readings along different axes.[1] The drawback which comes along using multiple sensors is the additional cost and the extra computational power required by the control unit. In addition to being expensive, the fact of having none of the common safety system features available in the modern vehicles to increase the passenger's safety can also be considered as a cause of the failure of self-balancing personal transporter concept.

The working principle of a self-balancing personal transporter is involved in continuously obtaining the feedback of the tilt (angle of inclination) of the platform, compensating the error with respect to the reference angle and maintaining the entire platform in an upright position. Further the ability of responding to any unexpected external force being applied in order to recover back to the stable position has been included in the control unit of the self-balancing transporter platform as it improves the overall safety assurance of the passenger

For the self-balancing transporter prototype presented by this research paper, an IMU unit (MPU6050) which is comprised of built-in accelerometer and a gyroscope is used to measure the acceleration and angular velocity readings along multiple axes and the angle of inclination of the platform can be simply estimated from both of these measurements separately.[7] However a single IMU unit which performs the task of multiple gyroscope and accelerometer sensors typically offer output signals combined with serious noise and therefore these signals are required to pass through a noise filtering process to achieve true angle of inclination estimation values. The main considerable noise components generated by the IMU unit can be listed as the gyroscopic drift and the horizontal acceleration dependency. Therefore a nose filtering process such as Complementary filtering or Kalman filtering can be applied to the IMU unit's output to obtain a better estimation of the angle of inclination of the self-balancing platform. The filtering process to be implemented highly depends on the performance of the microcontroller unit of the self-balancing transporter and it could also end up in indirectly affecting to the total

Author α σ ρ: Department of Electrical and Computer Engineering, Sri Lanka Institute of Information Technology, New Kandy Road, Malabe, Sri Lanka. e-mails: wish.pubudu1995@gmail.com, charith.s@sliit.lk

implementation cost. Finally a control system is required to control the motors of the self-balancing transporter with respect to the estimated angle of inclination and therefore the speed of the motors has to vary in order to maintain the platform in the upright position. A PID system is implemented as the control system of the self-balancing prototype and further designing phases with circuitry work are carried out to add a more professional touch to the implementation of the control unit of the self-balancing platform.

II. Literature Review

Comprehensive research was carried out to find out the information about the existing self-balancing transporter products and to reveal out design architectural information in order to implement a low-cost control unit for a self-balancing transporter vehicle.

a) Segway Self-Balancing Scooter Models

'Segway' company led by the inventor Dean Kamen was the very first to introduce a two wheeled self-balancing personal transporter type scooter in 2001. Even though the Segway was appeared to be a completely new form of transportation in the early stages, the concept completely failed in building a considerable customer base due to its' extremely high introductory price. Therefore in 2006, the company came up with a couple of new designed two-wheeled self-balancing personal transporters to suit different types of terrains. Segway I2 was introduced as the on-road general purpose personal transporter model while the Segway X2 model was designed with more advanced features for rough terrains and introduced as the off-road model. [2]

Figure 1: Segway I2 Model [2]

Figure 2: Segway X2 Model [2]

Both of these models consist of the working principle which requires the rider to lean forward to travel forward and do the opposite to move backward. Once the rider leans to the forward or reverse directions; the self-balancing scooter will start to move in the desired direction by maintaining the tile angle of the entire platform. The rider on the self-balancing scooter gets the opportunity to tilt the handlebar to drive the scooter in different directions. The tilt of the scooter platform is measured by a sensor unit consists of five gyroscope sensors and two accelerometer sensors.[2] Accelerometers and gyroscope sensors work separately to process the multiple accelerations and angular rate readings along multiple axes precisely in an extremely fast rate, the controller units of these personal transporter models are equipped with a highly powerful, expensive unit comprising of ten on-board microprocessors.[2] These facts can be considered as the major reasons for the Segway products to be tagged at an expensive price range. (Above $5000) However, these Segway models do not consist of any passenger safety features such as obstacle detection and braking system methods and as a result in most countries these models are banned from using in the public roads.[3]

b) Hover Boards

Hover boards can also be introduced as a representation of the self-balancing transporter concept. The steering operation is entirely different compared to self-balancing scooter models as the pressure sensor plates are placed on the pedal surface of hover boards to calculate the pressure difference and determine the turning direction. However the similar feature of both of the products can be highlighted as the self-balancing driving method which requires the rider to lean forward or reverse in order to move in the desired direction. The speed control unit of the hover board consists of two separate gyroscope sensors and two tilt sensors to obtain the angular rate and the accelerations along different axes to determine the tilt angle of the platform. (Figure 3) Even though there is a noticeable reduction in the number of accelerometer and gyroscope sensors compared to the control unit of the Segway models, the multiple gyroscope and accelerometer units in a hover board would still demand higher processing power.

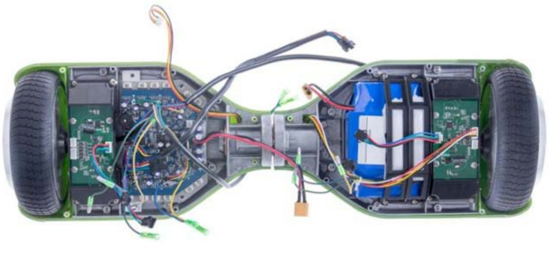


Figure 3: Inside of a hover board (Showing two separate speed control units)

III. Process of Obtaining the Angle of Inclination

The angle of inclination of the self-balancing prototype platform was obtained through the accelerometer readings of the IMU unit. Acceleration readings had to be converted into the degrees by considering the inverse tangent angle calculated from the acceleration readings alone y and z-axes. Changes in the angle of inclination concerning time had to be calculated by multiplying the angular velocity reading of the gyroscope of IMU with the time difference.

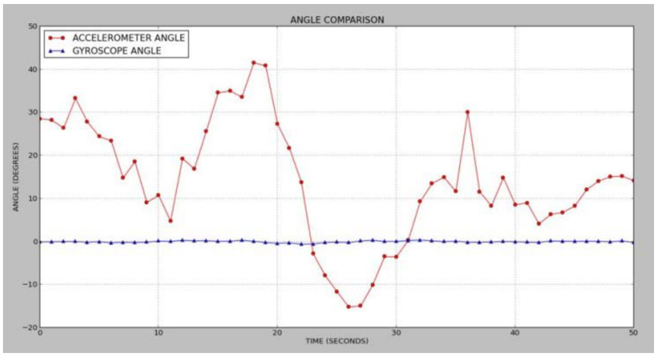


Figure 4: Real Time Python plot displaying the angle readings

IV. Implementation of the Noise Filtering Algorithms

a) *Estimation of the true angle of inclination*

The position and the stability of a self-balancing robot are simply affected by accelerations acting on it and the changing angular velocity of the robot platform. Therefore it was clear that both angle of inclination values and the angular change derived from accelerometer and gyroscope readings are required for a better estimation of the true angle of inclination of the self-balancing platform. Therefore the 'Sensor fusion' technique which is an input combination of multiple sensor readings to derive a single output was applied for the estimation process.

b) *Noise observations*

To obtain the true angle of inclination, it is obvious that the noises generated by the IMU unit must be cancelled out from a noise filtering process. Generally, the accelerometer is sensitive to the horizontal (x-axis) accelerations, and therefore it considers a horizontal acceleration as a change in the derived angle which causes huge noise in the derived angle output. On the other hand, the gyroscopic angle is sensitive to gyroscopic drift. Gyroscopic drift can be mainly introduced as the non-zero value that the gyroscope outputs when it is stationary even though it is supposed to output zero.

c) *Complementary Filter Algorithm Implementation*

Complementary filter algorithm which is a combination of high pass, low pass filtering stages and mathematical processes such as integration was selected as the first method to obtain true angle estimation of the platform. The true estimation of a sensor reading using the current and previously obtained sensor measurements can be considered as an intuitive approach for a sensor fusion application. The complementary filtering process inside the self-balancing platform can be represented as,

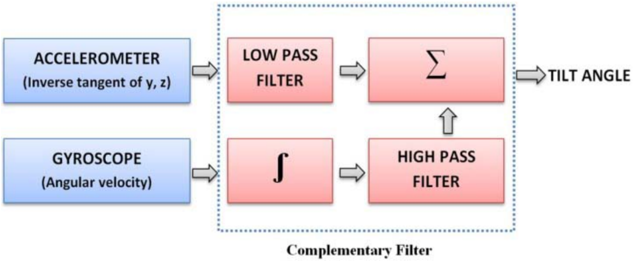


Figure 5: Complementary Filter Structure

Key things that affect the performance of the filtering process can be identified as the time constant and the filtering coefficient of the complementary filtering algorithm which are correlated with each other. Generally the time constant of the low and high pass filters are used to tweak the entire performance as it determines the filter coefficient of the filtering process.

- Complementary filter algorithm theory,
  Filtered angle = a * (current angle + gyro angle) + (1-a) * accelerometer angle [a = Filter coefficient]
- Complementary filter algorithm used for the self-balancing prototype,
  Filtered angle= 0.9934*(previous Angle + gyro_angle) + 0.0066*(accelerometer angle) [0.9934= Filter coefficient]

The value for the filter coefficient was selected as 0.0066 to obtain the most suitable filtered angle output from the complementary filtering process from a range of test data values for the specific prototype dimensions. Complementary filtered angle output was compared with the unfiltered angle values derived from IMU readings to ensure the elimination of horizontal acceleration noise and the gyroscope drift noise components respectively in accelerometer angle and gyroscopic angle.

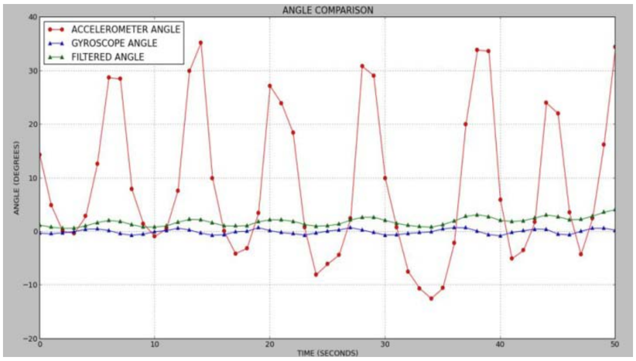


Figure 6: Comparing Complementary filtered angle (Green) with gyroscopic angle (Blue) and accelerometer angle (Red)

As the estimation provided by the complementary filtering process consisted of both the effects of accelerations acting on the prototype's frame and the changes in the angle of inclination (position), it was quite accurately providing the true angle of estimation of the prototype which depends on the entire stability maintaining.

d) Kalman Filter Algorithm Implementation

For a self-balancing platform application, Kalman filtering process can be defined as an iterative mathematical process that uses a set of equations made out of multi-dimensional matrices and data inputs to track objects by estimating the true values of velocity and position. Basically, it is focused on minimizing the variation or uncertainty in the continuous estimates with respect to the velocity and position data measurements. A state matrix (multi-dimensional) is formed to store the velocity and position data of the object which is being tracked. Process covariance (error) matrix contains the error in the estimation process.

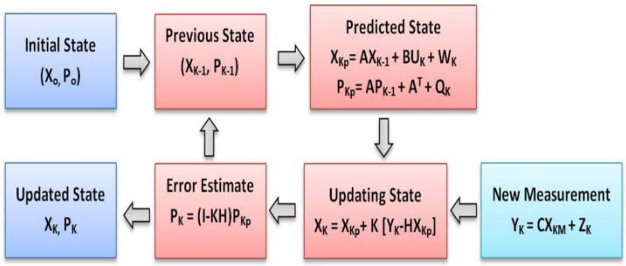


Figure 7: Iterative process for multiple dimensional Kalman filter model

U_k - Control variable matrix.

A, B, C - Adaptation matrices.

W_K - Predicted noise matrix.

X_{KM} - Measurement.

Q_K - Process covariance matrix.

Z_K - Measurement noise.

In the above process, U_K is used to combine a variable (acceleration) that affects both position and velocity to the predicted state. The intention of adaptation matrices is simply to ensure a common format between matrices. New estimate is processed for each data input by modifying the initial predicted state value with a portion multiplied by the Kalman gain (K) which determines the additional weight of sensor measurement and the predicted state value to be added. Kalman gain (K) can be explained with the sensor noise covariance matrix (R) which represents the measurement errors of relevant parameters of the IMU unit as,

$$K = P_{Kp.}H / (H.P_{Kp.}H^T + R)$$

The Kalman filtered angle of inclination was compared with the complementary filtered angle to observe the difference of true angle estimation to sort out the optimum filtering method. From the comparison result (Figure 8), it was clear that the predicted angle by the Kalman filter contains less variation from the true angle and more accurate response towards changes in velocity and position than the Complementary filter.

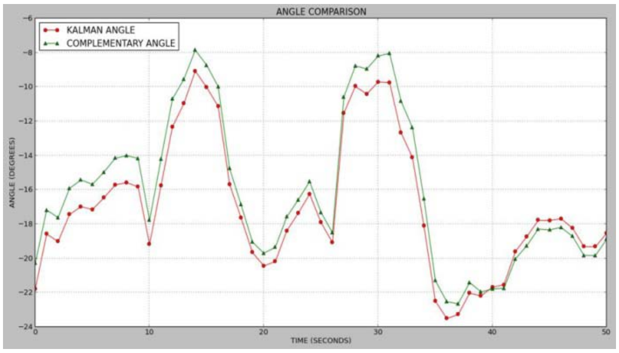


Figure 8: Comparing Kalman filter angle (Red) with Complementary filtered angle (Green)

V. Implementation of the Control System

a) Structure of the PID Control System

The intention of the PID control system is simply to control the motors of the self-balancing prototype according to rapid changes in the position. The basic algorithm to represent a PID control system can be given as [6],

$$u(t) = K_p\, e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^1 e(\tau) d\tau$$

The most important component of a PID control system can be considered as the feedback error value as it's combined with all of the constant values and used to generate the control signal output of the system. In the self-balancing platform, target or the reference angle can be calculated by positioning the robot in the upright position and therefore the feedback error value can be calculated as

e(t) = Current (Filtered) Angle–Target (Reference) Angle

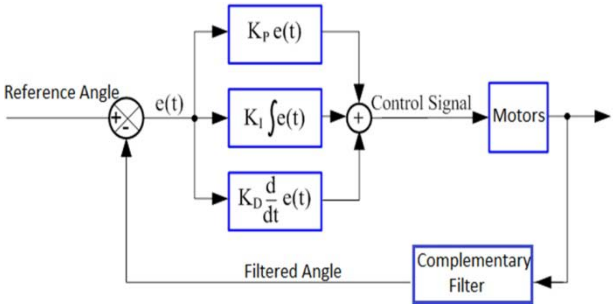
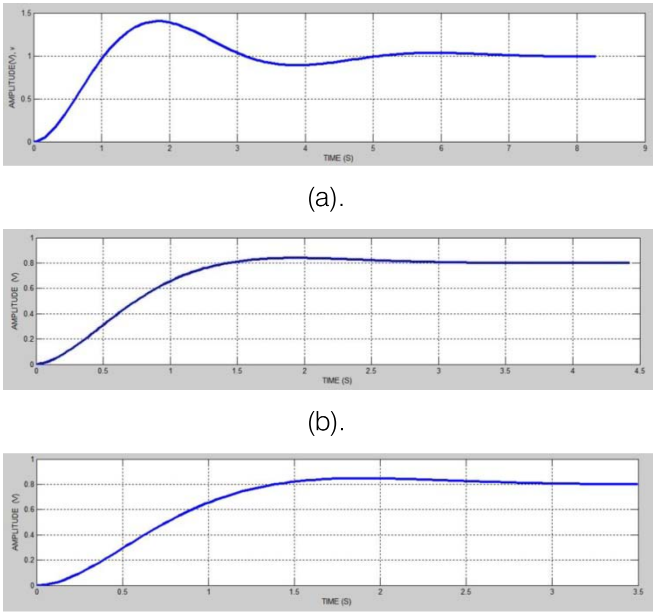


Figure 9: PID Control System Structure

As shown in Figure 8, the output signal of the PID control system is simply fed as the motor power to control the motors of the prototype according to the calculated error (difference) between the reference and the current (filtered) angle. Reference angle of the PID system is found out by measuring the angle of inclination of the platform when the robot frame is placed in an upright position.

b) PID Simulations

Matlab software-based simulations were carried out to find out the optimal values for the control terms (K_P, K_I and K_D) of a P, PD, and a PID controller. Unit step input (error value) for the simulation was generated by inputting a set of random angle value data. The system performance characteristics such as settling time, overshoot and rising time were observed by plotting the step response of the forms of the PID system with different sets of control term values. Depending on the characteristics of the response curve (Unit step response) of the PID, PD and P control systems, some value sets for the control terms were tested to sort out the best possible value range to shorten the settling time and to reduce oscillations in the control signal.



(a).

(b).

(c).

Figure 10: Step response of the control systems corresponding to optimal constant value sets, a). PID control system (Kp=60, Ki=4, Kd=0.3), b). PD control system (Kp=80, Kd=1.2), c). P control system (Kp=80)

Table 1: Performance Chart of Control System Simulations

Controller Type	Overshoot	Settling Time	Steady state error
PID	High	Very High	Very Low
PD	Low	Low	High
P	Low	Low	High

c) PID Tuning

However throughout practical experiments where manual PID tuning method was used to lock down the optimum control term values, the PID controller's performance with the minimum 'rise time' was not as stable as expected through the above simulation result. On the other hand, the PD controller provided a better stability for the prototype with a minimized steady state error which produced a negligible real-time effect to the overall balancing performance. Even though the performance of both P and PD control systems contain major similarities, the simulation result highlighted the slight increase in the 'rise time' in the P controller compared to the PD controller.

Table 2: Effect of Control Term Values in Tuning Process

Increased Control Variable	Improved Performance
K_P	Stability, Rise time
K_D	Overshoot, Settling time
K_I	Steady state error

As a result, the P controller presented a considerable stable balancing performs with slight oscillations and by assigning a suitable value for the K_D, the controller type was converted into a PD system and the overall performance was improved in to a better standard at the end.

VI. CAD Design and Hardware Implementation

CAD design of the self-balancing prototype was modeled through the 'Sketchup' software to secure the best possible weight distribution of the frame which directly affects to the balancing performance before the hardware implementation of the prototype.

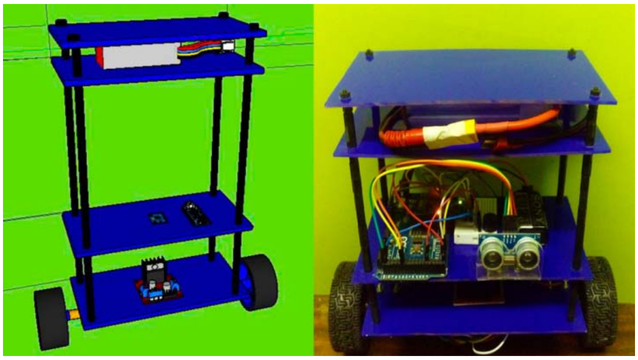


Figure 11: CAD design with the actual hardware implementation

VII. Conclusion and Future Work

The overall performance of the PD controller was ideal for the prototype to reach stability (upright position) with minimized oscillations and the shortest settling period. Further, the prototype was comfortable in responding rapidly to compensate the angle differences (errors) that occurred by various external forces. The control unit built through this research can be reused with relevant PID tuning parameters for differently scaled prototypes or Segway clones.

For similar experiments with self-balancing transporter prototypes, the safety system which was initially implemented through this research can be further improved. The sampling rate used to obtain the IMU readings and for the filtering process was 0.005 milliseconds and it was produced by internal interrupts of the Atmega128 chip. However the requirement of this rapid sampling rate prevented the flexibility of the microcontroller usage to carry out safety system experiments along with the balancing and filtering processes. Therefore as an improvement which is required for further experiments to implement a solid safety system for self-balancing transporter platforms, a separate microprocessor chip can be reserved to avoid conflicts between the priorities of each task. Further to preserve the compatibility of the circuit, both of the chips can be located in the same PCB with proper power distributions.

References Références Referencias

1. J. Fang, "A Posture Control System Design for a Two-wheeled and Self-balancing Robot", *Journal of IJ3C*, vol. 4, no.3, pp. 26-34, 2015.
2. "Segway - Simply Moving - Welcome to the Official Website of Segway Europe", Segway.com, 2018. [Online]. Available: http://www.segway.com/. [Accessed: 19- Apr- 2018].
3. J. Golson, J. Golson, A. Powell, W. Staff and A. Pardes, "Well, That Didn't Work: The Segway Is a Technological Marvel. Too Bad It Doesn't Make Any Sense", WIRED 2018.
4. K. Heong Ang and G. Chong, "PID Control System Analysis, Design, and Technology," IEEE Transactions on Control Systems Technology, vol. 13, No. 4, 2005, pp. 559-576.
5. Walter T Higgins, J R, "A Comparison of Complementary and Kalman Filtering", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES- 11, No. 3 , 1975.
6. A. Visioli., "Tuning of PID controllers" - Control Theory and Applications, 2001, vol.148, Issue1, pp. 1-8. "MPU-6050 | TDK", Invensense.com, 2018. [Online]. Available: https://www.invensense.com/ products/motiontracking/6-axis/mpu-6050/. [Accessed: 11- Jul- 2018].