

## Control System for a Segway

Jorge Morantes, Diana Espitia, Olguer Morales, Robinson Jiménez, Oscar Aviles

*Davinci Research Group, Militar Nueva Granada University, Bogotá, Colombia.*

### Abstract

In order to implement the control of a segway, a program was developed in a 32-bit microcontroller of the STM32F746 family that, through a PID-type control and together with a Kalman filter, allows to keep the balance in the device, obtaining a response in a short time thanks to the PID control, thus achieving to maintain the vehicle at a certain point with great efficiency.

**Keywords:** SEGWAY, Digital filters, PID control, accelerometer, pulse width modulation

### INTRODUCTION

With the objective of developing the control of a Segway like the one shown in figure 1, which is a vehicle with two concentric wheels that, through different types of digital strategies, must keep its balance. For the case that will be implemented below, the following elements were used:

- Analog accelerometer MMA7361.
- Two DC motors with gearbox.
- Bridge H L298N (Module)
- Microcontroller of the STM32F746 family
- Cables or jumpers for connections



Figure 1. Segway

When searching in databases with respect to other Segway models, it was found different ways of dealing with the problem of maintaining balance. In the model developed by students of the Korea National University of Transportation, in the article,

"Implementation of Unicycle Segway using unscented Kalman filter in LQR control", it is observed that the implementation of the control is done through an LQR or linear quadratic regulator and a Kalman filter, in addition that the measurement of the angle was made through a gyroscope and accelerometer, apart from these sensors was used an absolute encoder with 24-bit resolution to obtain the position of the DC motors. [2]

Another article found with respect to the control of a Segway, from the Slovak University of Technology, shows the development of a simulator, which allows observing the behavior of a device by altering the variables that comprise it, likewise including or excluding a PID control and the device reaction to the variation of the control's own constants, all this through a 3D model made in the free Blender software, the mathematical model driven by the OpenModelica software and executed through a web page. [3]

Finally, in a final article from the mechanical engineering department of the University of Peradeniya, talks about the complete modeling of a PID control as a controller of a segway. [4]

### METHODS AND MATERIALS

#### Kalman filter

The Kalman filter provides an efficient computational method to estimate the state of a linear stochastic process by minimizing the measurement of squared errors [1], therefore, it is understood that the Kalman filter is a purely digital method that allows the estimation of the response of a system in addition to the elimination of random noise to which a signal may be subjected. For the realization of this, 5 matrix equations are used, this matrix being according to the number of system variables that are required and the number of ways of sensing said variables.

For this system, a one-dimensional filter was used, so all operations with matrices were reduced to scalar operations.

Equations 1-5 govern the Kalman filter:

$$\hat{x}_{k-1} = \Phi_k * \hat{x}_{k+1} \quad (1)$$

$$P_{k-1} = \Phi_k * P_{k+1} * \Phi_k^T + Q_t \quad (2)$$

$$K_k = P_{k-1} * H_k^T (H_k * P_{k-1} * H_k^T + R_k)^{-1} \quad (3)$$

$$\hat{x}_k = \hat{x}_{k-1} + K_k * (\hat{y}_k - H_k * \hat{x}_{k-1}) \quad (4)$$

$$P_k = (I - K_k * H_k) * P_{k-1} \quad (5)$$

Where  $\hat{x}_{k-1}$  is the estimated value of the variable for the next instant of time.

- $\Phi_k$ : Transition matrix of states that relates the current state and the next state.
- $\hat{x}_{k+1}$ : Current value of the variable.
- $P_{k-1}$ : Estimation of the covariance matrix.
- $P_{k+1}$ : Current value of the covariance matrix.
- $Q_t$  and  $R_k$ : Matrices of errors.
- $H_k$ : Matrix that indicates the relationship between the measurements and the state vector.
- $\hat{y}_k$ : Input of the measured variables
- $I$ : Identity matrix.
- $K_k$ : Kalman gain.

### Basic PID control

Integrative derivative closed loop proportional control is one of the most used process control techniques. Reacts to process violations or deviations from a desired behavior using a strategy based on knowledge about error values, error history and forecasting future errors. [5]

Therefore, the PID control is based on a closed loop in which, by measuring the error with respect to a desired equilibrium point, an action is carried out that seeks to bring the system back to its point of stability. Proportional control is the one that performs the action based on the current error, taking an action as its name indicates according to the magnitude of the measured error. On the other hand, the integrative error, through the numerical integration method, obtains the area under the error curve, thus allowing the calculation of the error's history, and in this way contributes to the system reaching a point of equilibrium. Finally, the derivative control uses the current and the previous value of the error to calculate the slope of the error curve, thus allowing to give a response that approaches the future, the latter is what allows the system to have a faster response being essential for the control of a device such as the Segway. The equation 6, show the PID programed into the microcontroller.

$$C_{PID} = E(k) \times (Kp + Ki + Kd) - E(k - 1) \times (kp - 2kd) + E(k - 2) \times kd + C_{PID} \times (k - 1) \quad (6)$$

### Development of the program

To develop the program that allows to control and stabilize a segway, the keil compiler was used for programming in C++ language and configuration of a card of the STM32F746 family. Along with the knowledge acquired on how to use the compiler, the configuration of the pins of the card that were used was made in the first place, as follows:

- PF3 and PF4 pins configured in analog mode, for reading ADC from an accelerometer.
- PC6, PC7, PC8 and PC9 pins configured as outputs to control turns of two motors with an H bridge.
- Pins A0 and A1 in alternating configuration to generate PWM.
- E7 and E8 pins for serial communication via a Bluetooth module.

Once the pins were configured, the characterization of the analog accelerometer used to determine the inclination angle of the segway was performed and in this way, a movement in the corresponding motors was generated. For this case, the axes of the accelerometer that were used were X and Z. To do this, an ADC configuration code was prepared with pins PF3 and PF4 declared in analogous mode previously (see figure 2).

```
RCC->APB2ENR |= RCC_APB2ENR_ADC3EN;
ADC3->CR1 &=~ (1UL<<25);
ADC3->CR1 &=~ (1UL<<24);
ADC3->CR1 |= (1UL<<24);
ADC3->CR2 &= (1UL<<0);
ADC3->CR2 &=~ (1UL<<1);
ADC3->CR2 &=~ (1UL<<11);
ADC3->CR2 |= (1UL<<0);
```

Figure 2. ADC configuration for accelerometer reading on X and Z axes.

For the configuration, one of the ADCs that the card has, the data bus and the pins that are activated are configured based on the information indicated in the manual of the card.

After making this configuration, through a USB module, serial communication was made between the microcontroller and an interface known as -TERMINAL- from which the value indicated by the ADC was displayed. With the obtained data, the equation that gives the value of the component of gravity force in each of the axes of the accelerometer used to perform the measurements was determined (see equation 7):

$$Force = -0,03939759036 * data + 19,13722892 \quad (7)$$

Where:

- data  $\rightarrow$  corresponds to the value read from ADC by the accelerometer.

With the accelerometer characterized, and the calculated force components, it is proceeded to determine the corresponding angle of inclination, with the use of trigonometric ratios (see equation 8):

$$\theta = \tan^{-1} \left( \frac{ForceZ}{ForceX} \right) \quad (8)$$

After determining the value of the angle according to the inclination of the accelerometer, it is proceeded to elaborate the code for the configuration of timers with PWM, and thus control the movement of the motors to prevent the segway from falling and achieving a state of stability (see figure 3).

```
RCC->APB1ENR |= RCC_APB1ENR_TIM5EN;
TIM5->PSC = 31;
TIM5->ARR = 999;
TIM5->DIER |= 0x1;
TIM5->CR1 |= 0x1;
TIM5->CCMR1 |= 0x6060;
TIM5->CCER |= 0x11;
TIM5->CCR1 = cicloUtil1;
TIM5->CCR2 = cicloUtil2;
NVIC_EnableIRQ(TIM5_IRQn);
```

Figure 3. Timer configuration to generate PWM signal.

One of the factors that affected the stability of the robot was the noise in the ADCs to calculate the value of the angle. To reduce this factor and thus improve the behavior of the segway, two filters were applied, which allowed obtaining a more accurate angle value and therefore a better reaction in the movement of the motors, in the following way:

**Moving average filter:**

20 angle values were taken at the same point and the average between the data was determined (see figure 4).

```
void llenarFiltro(double angle){
    for(int i = 19; i>0; i--){
        filtro[i] = filtro[i-1];
    }
    filtro[0] = angle;
}
void valorFiltrado(){
    double promAngle = 0;
    for(int i=0; i<20; i++){
        promAngle += filtro[i];
    }
    angulo = promAngle/20;
}
```

Figure 4. Moving average filter.

**Kalman filter:**

This digital filter was applied, which makes it possible to estimate the future states of a variable. In this case, the observed variable was the angle, and the entire process and application of equations for Kalman prediction and correction were performed to the angle value obtained with the moving average filter. In Figure 5, the code developed to perform this filter is shown.

```
Yt = angulo;
Xt = Ft * Xt;
Pt = Ft * Pt * Ft + Qt;
Kt = (Pt * Ht)/(Ht * Pt * Ht + Rt);
Xt = Xt + Kt*(Yt - Ht * Xt);
Pt = (Pt - Kt * Ht * Pt);
```

Figure 5. Kalman filter.

Figures 6 and 7 show the variations in the filtered angle data after Kalman application.



Figure 6. Angle measurement in steady state.

In Figure 6, the comparison of the angle data determined with the accelerometer can be observed. In blue, the calculated value is found before the filter application, while in red the angle data appears with the Kalman filter. This graph was obtained at the moment in which the segway was in stable state, in this way, it can be verified the importance of the applied filter, achieving a more accurate value of the measured variable.



Figure 7. Angle measurement, segway in motion.

In Figure 7, the comparison chart of the angle values is shown when the segway is in motion, seeking stability against the presented variations. In blue, the data without filter is observed while in red, the calculated value appears once the Kalman filter has been applied. As shown in the graph, the filter helps to considerably reduce the noise present during the measurement, although the response becomes a little slower.

The next step to the application of the digital filters was to develop a basic PID controller to improve and feedback the system response. The values of the constants were calculated empirically, since the model of the physical plant was not determined. The obtained values are:

**- Proportionality constant (Kp) = 80.**

There is a limit value in the proportional constant and, in some cases, the system reaches values that are higher than the desired data. The phenomenon that governs this constant is the over oscillation of the system. Since this value does not consider time, it is necessary to include integral and derivative constants. For the manual calculation, a value was established that allowed the oscillation of the physical plant with the values of

the other constants at 0. To calculate the value of integral and derivative constants, it was proceeded to assign to the proportional constant half of the calculated value to generate oscillation.

This PID constant allows the system to react to a change in the reference value, so if the value is very small, the force exerted on the system is not sufficient to allow it to return to the equilibrium position, while, if the value is very large, to small variations, the force applied is very high, causing the segway to lose balance. In comparison with the values of the other constants, the proportional is greater.

#### - Integral constant (Ki) = 1.2

This constant of the PID indicates the speed with which the proportional action is repeated. To manually determine the value, once the proportional part is set, the integral value is increased until the process is adjusted in the required time. It is important that the value is not too big to avoid system instability, since the sum of the errors that are made for the integrative part often generates a slow response of the system.

#### - Derivative constant (Kd) = 13.

The aim of the derivative action is to maintain a minimum error, correcting it proportionally with the same speed that produces it, seeking to avoid an increase in the error. If the value of this constant is very small, there is a slow reaction to large variations in position, therefore, the system cannot return to the equilibrium position, while if the value is very large, high sensitivity to small variations are presented, generating sudden changes and loss of balance.

Additionally, to perform the summation of the constants, the sampling time was taken into account, that was 5 ms (this sampling was done using a timer at a frequency of 200 Hz), as well as the error.

With this controller the useful cycle value of the motors was determined in order to make faster or slower the response of these at a certain angle of inclination. In figure 8, the code developed for PID is shown.

```
error = Ref - Xt;
errorPass = error*TiempoMuestreo + errorPass;
errorD = (error - errorAnt) / TiempoMuestreo;
P = kp*error;
I = ki*errorPass;
D = kd*errorD;
errorAnt = error;
cicloUtil1 = P + D + I;
cicloUtil2 = P + D + I;
```

**Figure 8.** Basic PID controller on segway

Where:

- Ref → is the variable that indicates the degree to which the segway maintains equilibrium, that is, the 0 of the system. For the segway, the reference is 18°.
- Xt → is the angle value obtained once the Kalman filter has been applied.

- TiempoMuestreo → variable that represents the period in which the sampling is carried out.
- error → physical plant error.
- errorPass → sum of the error for the integral part.
- errorD → error for the derivative calculation.
- errorAnt → variable to update the error.
- cicloUtil1 / cicloUtil2 → Value calculated for the basic PID control.
- P → proportional.
- D → derivative.
- I → integral.

## CONCLUSIONS

The data of the angles delivered by the analog accelerometer present a great variety of difficulties, from the internal error of the sensor itself, the sensitivity of the same to impacts, since these when applying a force, interfere with the measurements.

The plant and the distribution of the weight represented a great problem at the time of achieving the equilibrium of the system, since due to a center of mass not aligned with respect to the center, the body tended to lean.

The application of a Kalman filter was of great importance to achieve the stability of the system, since the sensor used had a high susceptibility to noise, causing the PID response was not appropriate. On the other hand, the use of a PID control is the basis of the system, since without it, it is impossible to make the response of the system efficient enough.

## ACKNOWLEDGMENT

The research for this paper was supported by Davinci research Group of Nueva Granada Military University.

## REFERENCES

- [1] H. Ferdinando, H. Khoswanto and D. Purwanto, "Embedded Kalman Filter for Inertial Measurement Unit (IMU) on the ATmega8535," 2012 International Symposium on Innovations in Intelligent Systems and Applications, Trabzon, 2012, pp. 1-5. doi: 10.1109/INISTA.2012.6246978.
- [2] Han, M., Kim, K., Do Youn Kim and Lee, J., 2013, "Implementation of unicycle segway using unscented Kalman filter in LQR contro,". In URAI , pp. 695-698. doi: 10.1109/URAI.2013.6677427.
- [3] Reza Ezuan Samin, Lee Ming Jie and M. A. Zawawi, "PID implementation of heating tank in mini automation plant using Programmable Logic Controller (PLC)," International Conference on Electrical, Control and

Computer Engineering 2011 (InECCE), Pahang, 2011, pp. 515-519.

- [4] Basnayake, I.D., Madhushani, T.W.U. and Maithripala, D.H.S., 2017, "Intrinsic PID controller for a segway type mobile robot," In Industrial and Information Systems (ICIIS), 2017 IEEE International Conference on, pp. 1-6. doi: 10.1109/ICIINFS.2017.8300402.
- [5] Dr. Eman Kamel; Dr. Khaled Kamel: Hands-On PLC Programming with RSLogix 500 and LogixPro®. Analog Programming and Advanced Control, Chapter (McGraw-Hill Professional, 2016), AccessEngineering