ORIGINAL ARTICLE

A NEW SLIDING MODE CONTROLLER IMPLEMENTATION ON AN AUTONOMOUS QUADCOPTER SYSTEM

Vincent Delica[1], Alfred Orquia[1], Jayson Piquero[1, *], Joel Ilao[2], Eloise Reynaldo[1], Ma. Antonette Roque[1], Edwin Sybingco[1], Alvin Chua[3], Jayantha Katupitya[4], and Hiranya Jayakody[4]

[1]*De La Salle University. Department of Electronics and Communications Engineering*

[2]*De La Salle University. Department of Computer Technology*

[3]*De La Salle University. Department of Mechanical Engineering*

[4]*University of New South Wales. School of Mechanical and Manufacturing Engineering*

(**Received** 16 June 2018; **Accepted** 16 August 2018; **Published on line** 1 June 2019)

*Corresponding author: jayson_piquero@dlsu.edu.ph

DOI: 10.5875/ausmt.v9i2.1876

Abstract: This paper introduces Sliding Mode controller, a non-linear controller, for the implementation of an autonomous quadrotor platform, a non-linear system. The Sliding Mode controller was applied to a PixHawk Flight Controller using the Ardupilot firmware. The simulation testing using SITL shows the effectiveness of the controller before flight. The results imply the improvement when using Sliding Mode Control in comparison to PID controller. The results show that there is a reduction in attitude error when using Sliding Mode Control in comparison with PID control in all simulation and actual hardware results. The robustness of Sliding Mode Control was also tested by adding parameter uncertainties and disturbances to the system. In this study, the root-mean-square error obtained in the Sliding Mode Control is 1.546580%, 0.634243%, and 13.466256% for the roll, pitch, and yaw movements respectively, and the root-mean-square error obtained in the PID control is 2.588324%, 4.553838%, and 18.860183% for the roll, pitch, and yaw movements respectively. This shows that the quadrotor using Sliding Mode Control is less prone to attitude errors.

Keywords: quadrotor; sliding mode controller; nonlinear; pixhawk; ardupilot

Introduction

Unmanned Aerial Vehicles (UAV) has been a subject of many studies due to its many applications such as in military, security, emergency response and aerial mapping. It can take on, off and hover freely in the air at any time [1]. A Quadrotor UAV has six degrees of freedom; it can move and rotate along the three-dimensional space [2]. The movement of the quadrotor is controlled by four independent motors which allow the angular movements yaw, pitch, and roll. Unfortunately, having only four motors makes the vehicle underactuated [1]. Therefore, the quadrotor control and movement are limited. The type of controller plays a big factor on the effectivity and accuracy of the quadrotor movement. Recent quadrotor controllers are based on linear methods such as the Proportional Integral Differenctial (PID) and Linear Quadratic Regulator (LQR) [1]. However, linear controllers have limited performance due to the strong nonlinearity of the quadrotor UAV [1]. Also, one of the concerns in designing a quadrotor UAV is the stability. Consequently, nonlinear controllers are used to solve the problem regarding the instability of the system.

One of the most common and widely used controller is the PID controller. This type of controller is also commonly used for Unmanned Aerial Vehicles (UAV) to ensure stability and robustness of the system. However, the PID controller has poor performance in controlling nonlinear systems [3]. Using linear controllers will not be an effective way to improve the system behavior. Linear controllers like PID controllers cannot meet the desired requirements (e.g. stability and altitude gain) due to nonlinearity in the system. Software simulations of PID controllers appear to be asymptotically stable; however, in the actual implementation, there are difficulties in achieving stability due to external factors. Moreover, the presence of parameter uncertainties and exposure to intense disturbance will have an effect on the behavior of the controller. PID control can be easily implemented but does not guarantee robustness to parameter variation [12]. Thus, stability and robustness of the system is not easily achieved through the use of PID or other linear controllers.

Practically, there are always discrepancies between the mathematical model and the actual plant of the controller design [4]. This can be caused by external disturbances which are not taken in consideration in the mathematical model. This is a common problem among linear controllers [8]. According to [4], linear state feedback controllers can only reach asymptotic stability when there are no disturbances and other external factors. In reality, this is impossible. Therefore, due to these limitations, quadrotors tend to be unstable [9]. This paper aims to solve this problem by implementing the sliding mode controller in a quadrotor UAV. The sliding mode control is a better option for a quadrotor controller due to its insensitivity to model errors and parameter uncertainties [6]. Because drones have a wide variety of applications, improving the control with Sliding Mode Control is relevant, and a number of publications discussed Sliding Mode Control theoretically and in simulations. There are already existing researches in which Sliding Mode Control is implemented in a quadrotor and compared with PID but only in a simulation as seen in [10] [11]. In this paper, the effects of disturbance and parameter uncertainties are also tested, and the performance is compared to a PID Controller. It was concluded that the Sliding Mode Control is more robust to disturbance and parameter uncertainty than the PID Controller. This paper aims to be different from other researches by implementing Sliding Mode control for the roll, pitch, and yaw controllers and also to implement it in the actual hardware. This study also aims to test the robustness of Sliding Mode Control by inserting disturbances and parameter uncertainties. The Root-Mean-Square Error was obtained for both PID and Sliding Mode Control responses for statistical comparison. Chattering reduction is also discussed and implemented in simulation to maximize the features of Sliding Mode Control.

There have been researches wherein non-linear controllers, such as the sliding mode controllers, were implemented on unmanned aerial vehicles. According to [2], implementing the sliding mode control method will improve control precision and stabilize the movement of the quadrotor. The concept behind the sliding mode control is that it drives the state variables of all parameters unto a particular surface in the state space called the “sliding surface” [4]. Sliding Mode Control relies on the dynamic parameters of the system. As seen on equations (9), (11), and (12), the control inputs of Sliding Mode Controller use the dynamic equations of the quadrotor as seen on equations (1) to (6) to accurately minimize the error [2]. This is where the difference of Sliding Mode Control and PID Control can be seen. Since PID Control only minimizes the error, a slight error in tuning can cause oscillation due to its linearity [8]. The main advantage of using a sliding mode controller over the linear controller is that it is totally insensitive to particular parameter uncertainties such as external disturbances and changes in the quadrotor dynamics [4]. The main drawback on using sliding mode control is the chattering phenomenon which is the undesirable oscillations of parameters This destructive phenomenon may lower control accuracy or incur unwanted wear of mechanical components [6]. This problem has been addressed in [2] wherein a new control strategy that would eliminate the chattering problem without affecting the stability and robustness of the quadrotor.

Vincent Delica graduated Bachelor of Science in Electronics Engineering in the Electronics and Communications Engineering Department at De La Salle University, Manila, Philippines 1004.

Alfred Orquia graduated Bachelor of Science in Electronics Engineering in the Electronics and Communications Engineering Department at De La Salle University, Manila, Philippines 1004.

Jayson Piquero graduated Bachelor of Science in Electronics Engineering in the Electronics and Communications Engineering Department at De La Salle University, Manila, Philippines 1004.

Eloise Reynaldo graduated Bachelor of Science in Electronics Engineering in the Electronics and Communications Engineering Department at De La Salle University, Manila, Philippines 1004.

Joel Ilao is an Associate Professor of the Computer Technology Department at De La Salle University, Manila, Philippines 1004.

Ma. Antonette Roque is an Assistant Professor of the Electronics and Communications Engineering Department at De La Salle University, Manila, Philippines 1004.

Edwin Sybingco is an Assistant Professor of the Electronics and Communications Engineering Department at De La Salle University, Manila, Philippines 1004.

Alvin Chua is a full Professor of the Mechanical Engineering Department at De La Salle University, Manila, Philippines 1004.

Jayantha Katupitya is an Associate Professor at University of New South Wales, Sydney, Australia 1466.

Hiranya Jayakody is a Postdoctoral Fellow of School of Mechanical and Manufacturing Engineering at the University of New South Wales, Sydney, Australia 1466.

In this research, a new Sliding Mode Control implementation is done in the PixHawk flight controller through the ArduPilot firmware. The ArduPilot firmware makes use of the SITL simulator which is one of the simulators of ArduPilot that can be connected to Mission Planner, an open source ground control software that connects to the drone and can create data logs for obtaining results from flights. Because of this, an improved attitude controller of the drone can be made available to users that want to use Sliding Mode Control algorithm for their flights. This research covers not only simulations but also actual hardware flights. A lot of researches such as [10] and [11] focused on simulations because Sliding Mode Control is theoretically more accurate than PID in a quadrotor system, however testing the controller in hardware confirms the advantages of Sliding Mode Control in the actual flight tests. A lot of researches such as [10] and [11] focused on simulations because Sliding Mode Control is theoretically more accurate than PID in a quadrotor system, however testing the controller in hardware confirms the advantages of Sliding Mode Control in the actual flight tests.

The effectiveness of the sliding mode controller is proven using a series of simulated and actual flight tests. To determine the improvement of the performance, similar set of flight tests are used for a PID controller which is then compared to the results of the Sliding Mode Controller. The flight tests are analyzed by comparing the actual position at each given time to the desired position. This determines which controller can traverse the waypoints of the path with the least deviation to the actual positions during flight. Using these data, the stability of the controller system is also determined by calculating the percent error. A small percent error means that the system can accurately follow the desired positions during flight even with the presence of external disturbances. Therefore, if a smaller percent error is attained, the controller is more stable. The contribution of this paper is the implementation of the Sliding Mode Controller and its comparison to PID controller. This paper also introduces improvement in Sliding Mode Control by adding chattering reduction and parameter uncertainty control.

Quadrotor Dynamic Model

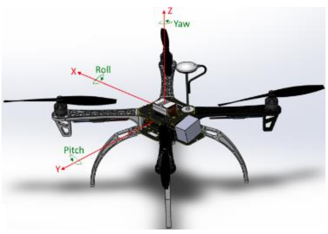

Figure 1. Quadrotor Dynamic Model.

The quadrotor model is shown in Figure 1. It illustrates the cartesian coordinate frame (x,y,z), which indicates the position of the quadrotor, and the Euler angles (Φ,θ,ψ), which indicate the orientation of the quadrotor. The system is divided into two frames namely, the earth fixed frame (E) and the body fixed frame (B). The earth fixed frame is the one attached to the earth, and the body fixed frame is attached to the center of mass. The quadrotor has four motors which operate in pairs. Motors 1 and 3 should be working in the clockwise direction that will create pitch angle and motion in x axis. On the other hand, motors 2 and 4 should also be working in the counterclockwise direction. Motors 2 and 4 are the ones responsible for creating roll angle and motion in y axis. The difference between the counter torque of motors 1 & 3 and counter torque of motors 2 and 4 dictates the yaw angle. Motion along z axis is determined by the total forces created by all motors.

Following Newton's laws of motion, the dynamic equations can be derived, considering the parameter uncertainties. The number of inputs in a quadrotor is four while the number of outputs is six, as defined in Equations 1 to 7. This means that not all parameters can be controlled simultaneously. Equations (1) to (7) are based on [4], [7], and [13].

$$\ddot{x} = \frac{(\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi)u_1 - K_{ftx}\dot{x}}{m} \tag{1}$$

$$\ddot{y} = \frac{(\cos\varphi\sin\theta\cos\psi + \sin\varphi\cos\psi)u_1 - K_{ftx}\dot{y}}{m} \tag{2}$$

$$\ddot{z} = \frac{(\cos\theta\cos\psi)u_1 - K_{ftx}\dot{z}}{m} \tag{3}$$

$$\ddot{\varphi} = \frac{I_{XX} + \Delta I_{XX} + I_{YY} + \Delta I_{YY} - I_{ZZ} - \Delta I_{ZZ}}{I_{XX} + \Delta I_{XX}}\dot{\psi}\dot{\theta} + \frac{u_2}{I_{XX} + \Delta I_{XX}} \tag{4}$$

$$\ddot{\theta} = \frac{-I_{XX} - \Delta I_{XX} - I_{YY} - \Delta I_{YY} + I_{ZZ} + \Delta I_{ZZ}}{I_{YY} + \Delta I_{YY}}\dot{\psi}\dot{\varphi} + \frac{u_3}{I_{YY} + \Delta I_{YY}} \tag{5}$$

$$\ddot{\psi} = \frac{I_{XX} + \Delta I_{XX} - I_{YY} - \Delta I_{YY} + I_{ZZ} + \Delta I_{ZZ}}{I_{ZZ} + \Delta I_{ZZ}}\dot{\theta}\dot{\varphi} + \frac{u_4}{I_{ZZ} + \Delta I_{ZZ}} \tag{6}$$

Sliding Mode Control of the Quadrotor

The sliding mode control is a nonlinear control method that drives the system dynamics to a desired set of trajectories called the sliding surface. Upon reaching the surface, a control law will maintain the dynamics within the sliding surface. The control aim is to make the output error variable which is the difference between the desired and the actual value to be close to zero as possible after an acceptable length of duration. All equations from (8) to (12) are based from [3].

The sliding mode control has two phases:

- Sliding Surface design
- Control Law Design

Sliding Surface Design: The first phase involves the design of a sliding surface so that the sliding motion satisfies design specifications. In general, the sliding surface design can be described by the function:

$$S_X = \dot{e}_X + \lambda_X e_X \tag{7}$$

Where λ is a constant positive value and e=x-xd.

Control Law: The first phase involves the design of a sliding surface so that the sliding motion satisfies design specifications. In general, the sliding surface design can be described by the function:

$$U = U_{eq} + U_s \tag{8}$$

Where U_{eq} is the corrective control, U_s is the switching control.

Corrective control is used to compensate the deviations from the sliding surface to reach the sliding surface whereas equivalent control is used to make the derivative of the sliding surface equal zero to stay on the sliding surface.

According to [3], the Roll controller is derived into:

$$u_\varphi = (I_{XX} + \Delta I_{XX})\begin{pmatrix} \ddot{\varphi}_d - \dfrac{I_{XX} + \Delta I_{XX} + I_{YY} + \Delta I_{YY} - I_{ZZ} - \Delta I_{ZZ}}{I_{XX} + \Delta I_{XX}}\dot{\psi}\dot{\theta} \\ -\lambda(\dot{\varphi} - \dot{\varphi}_d) - u_D \end{pmatrix} \tag{9}$$

Where:

$$u_D = K_\varphi \frac{S_\varphi}{|S_\varphi| + \varepsilon_\varphi} \tag{10}$$

The same process of derivation will be made for pitch and yaw.

$$u_\theta = (I_{YY} + \Delta I_{yy})\begin{pmatrix} \ddot{\theta}_d - \dfrac{-I_{XX} - \Delta I_{XX} - I_{YY} - \Delta I_{yy} + I_{ZZ} + \Delta I_{ZZ}}{I_{YY} + \Delta I_{yy}}\dot{\psi}\dot{\varphi} \\ -\lambda(\dot{\theta} - \dot{\theta}_d) - u_D \end{pmatrix} \tag{11}$$

$$u_\psi = (I_{ZZ} + \Delta I_{ZZ})\begin{pmatrix} \ddot{\theta}_d - \dfrac{I_{XX} + \Delta I_{XX} - I_{YY} - \Delta I_{yy} + I_{ZZ} + \Delta I_{ZZ}}{I_{ZZ} + \Delta I_{ZZ}}\dot{\theta}\dot{\varphi} \\ -\lambda(\dot{\psi} - \dot{\psi}_d) - u_D \end{pmatrix} \tag{12}$$

Mass Moment of Inertia

The mass moment of Inertia is a vital part in the performance of the Sliding Mode Controller. An experimental method was used to determine the mass moment of Inertia of the quadrotor. It involves the use of a trifilar pendulum that will measure the oscillations along each of the quadrotor body axis. Using the measured oscillations, the moment of inertia can be calculated using (13). Equation (13) is based from the method of calculating moment of inertia in [5].

$$I_{XX,YY,ZZ} = \frac{WR^2 T^2_{X,Y,Z}}{4\pi^2 L} \tag{13}$$

Where:

$I_{xx,yy,zz}$ = mass moment of inertia of object in x, y, and z-axis,
T = Period of one oscillation in s,
W= weight of the disc and quadrotor in N,
R = radius of disc in m,
L = length of wire suspending disc from ceiling in m.

Three of these parameters were obtained by measuring and weighing. These values can be seen below:
L = 30cm W = 14.715N R = 17.75cm

The pendulum which holds the quadrotor according to a certain axis will be rotated measuring the period of time it took to complete a number of oscillations:

Figure 2. Measurement of Moment of Inertia along the x-axis.

As seen on Figure 2, the drone is hung from its front and tied to a circular wire. This allows the user to insert a force that can cause rotations of the quadrotor around its x-axis.

Figure 3. Measurement of Moment of Inertia along the y-axis.

As seen on Figure 3, the drone is hung from its right side and tied to a circular wire. This allows the user to insert a force that can cause rotations of the quadrotor around its y-axis.

Figure 4. Measurement of Moment of Inertia along the z-axis.

As seen on Figure 4, the drone is hung from its top and tied to a circular wire. This allows the user to insert a force that can cause rotations of the quadrotor around its y-axis.

By applying a small force into the drone, the drone will swing back and forth. The period of 10 oscillations is recorded 10 times to lessen the effects of human and equipment error.

Table 1. Periods of Oscillation.

T_x (seconds per 10 oscillations)	***T_y (seconds per 10 oscillations)***	***T_z (seconds per 10 oscillations)***
6.51	7.01	8.19
6.16	7.04	8.01
6.37	7.11	8.13
6.45	7.07	8.04
6.71	6.84	8.11
6.77	7.20	8.10
6.61	6.91	7.93
6.28	7.19	7.86
6.25	7.06	8.2
6.38	6.82	7.884

$$\boldsymbol{T_x = 0.6649\ second/oscillation}$$
$$\boldsymbol{T_y = 0.70248\ second/oscillation}$$
$$\boldsymbol{T_z = 0.80214\ second/oscillation}$$

From there, the period for one oscillation of the quadrotor will be calculated. Once the period of oscillation is obtained and the other parameters are measured, the mass moment of inertia can now be finally calculated using equation (13).

Table 2. Moment of Inertia.

I_{xx}	0.017305
I_{yy}	0.019317
I_{zz}	0.025187

Chattering Reduction Equation

Chattering phenomenon is the high frequency oscillations in the control input which is present in Sliding Mode Control. This is undesirable because of the energy consumption and this can also cause vibrations in the system. To reduce the chattering phenomenon, the switching control of the control law design was replaced with equation:

$$K\frac{S}{|S|+\varepsilon} \qquad (14)$$

Where ϵ is the tuning parameter of the chattering effect and s is the sliding surface

Using a series of simulated and actual tests, the control law with chattering reduction equation was compared with the one without to prove the efficiency of the equation. The tests made were movements in the horizontal plane which are the movements affected by the chattering phenomenon.

Error Computation

In this study, error was computed between the desired and measured values of the roll, pitch, and yaw angles for the comparison and analysis of the performance of the quadrotor. The Root-Mean-Square-Error (RMSE) was taken as the percentage error of the system.

$$RMSE = \sqrt{\frac{\sum(X_{desired} - X_{measured})^2}{T}} \qquad (15)$$

Simulation Results

The Sliding Mode Controller is implemented using the Ardupilot source code. To test the controller in a simulation, Software-in-the-loop (SITL) is used. From every set of waypoints, a detailed comparison of the performance of PID and Sliding Mode Control Controllers is measured, showing all the roll, pitch, and yaw values (desired and actual) for PID and Sliding Mode Control.

These mentioned parameters are graphed into MATLAB, and the accuracy was computed in this software. In measuring the accuracy, the root mean-square error between the desired and measured values (roll, pitch, and yaw) were taken as seen in equation (15). These values were implemented in MATLAB and were used as the measure of accuracy of the quadrotor.

Figure 5. Waypoint Set.

Waypoints are simply points in the world map where the drone is required to navigate to, using the GPS module as its guide. Each waypoint consists of the longitude, latitude, and altitude information, in order to properly command the drone to follow the desired positions. In simpler terms, waypoints are simply the path the drone is required to follow. Both PID and Sliding Mode Control Algorithms are used in the simulation for comparison. The software, Software-In-The-Loop (SITL), is mostly used for the simulations of Ardupilot-based codes. As seen on Figure 5, the waypoints set is the set of points in the world map where the drone is set to follow. This is done in the simulations using SITL connected to Mission Planner. The quadrotor dynamic parameters used are listed on Table 3, initially ignoring parameter uncertainties.

Table 3. Simulation Dynamic Parameters.

I_{xx}	0.05
I_{yy}	0.05
I_{zz}	0.24
ΔI_{xx}	0
ΔI_{xx}	0
ΔI_{xx}	0

PID Control Algorithm

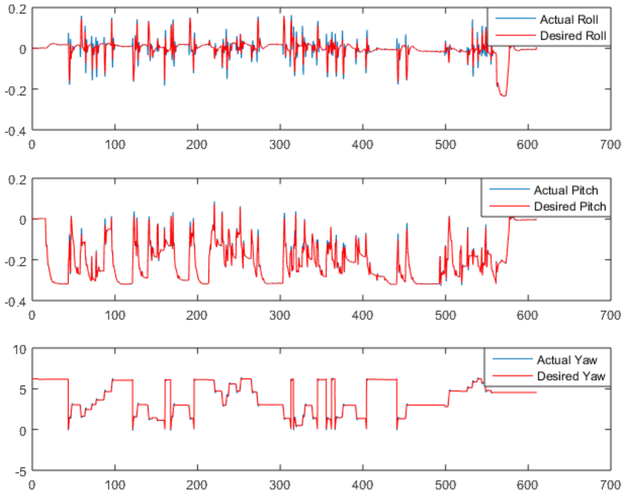

Figure 6. Simulated Roll, Pitch, and Yaw Response for PID.

Calculated Roll RMSE: 0.9814%
Calculated Pitch RMSE: 0.5722%
Calculated Yaw RMSE: 5.9883%

As it can be seen in Figure 6, the drone used PID control to follow the waypoints seen in Figure 5. The roll, pitch, and yaw movements are displayed in MATLAB graphs after the drone has finished following the path.

Sliding Mode Control Algorithm

A set of tuning parameters were initialized in this simulator. Since the error of SMC yielded a smaller error compared to PID, the tuning process was no longer required as the initialized tuning parameters were already functional.

Table 4. Measurement of Moment of Inertia along the z-axis.

	λ	K	ε
φ	1	1	0.5
θ	2.5	3	1
ψ	1	1	0.1

As it can be seen in Figure 7, the drone used Sliding Mode Control to follow the waypoints seen in Figure 5. The roll, pitch, and yaw movements are displayed in MATLAB graphs after the drone has finished following the path. Based from the simulated graphs of both controllers (Figure 6 and Figure 7), it can be seen that the percentage error for roll, pitch and yaw of the Sliding Mode Control is smaller than that of PID. It can also be seen that there is a significant difference between the Yaw of PID and Sliding Mode Control. Therefore, it can be concluded that the Sliding Mode Control can track the desired values with more accuracy based on the simulations provided in Figure 5.

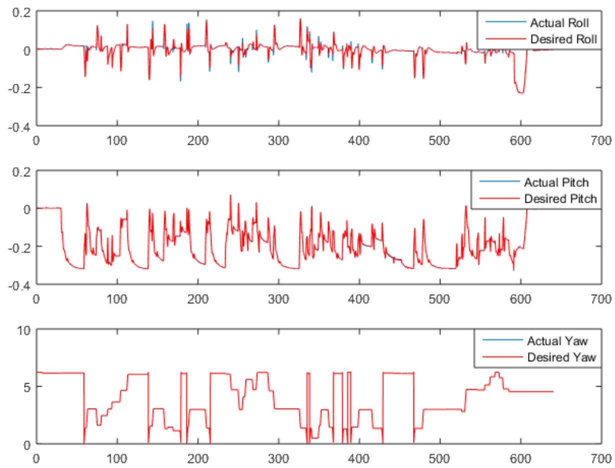

Figure 7. Simulated Roll, Pitch, and Yaw Response for SMC.

Calculated Roll RMSE: 0.6334%
Calculated Pitch RMSE: 0.4484%
Calculated Yaw RMSE: 0.1437%

Parameter Uncertainties

Since Sliding Mode Control relies on the dynamics of the system, the control inputs may be varied depending on the parameter uncertainties. In Figure 8, the quadrotor is to be simulated using the waypoints shown. This will involve simple movements: Forward-Right which is a simultaneous Roll and Pitch movement.

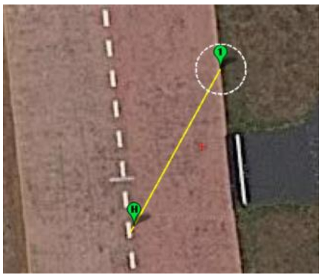

Figure 8. Simulation Waypoints with Parameter Uncertainties.

The following Moments of Inertia denoted by I_{xx}, I_{yy}, I_{zz} are initialized in Table 5. The response of the system with parameter uncertainties are tested using Monte Carlo method. The flight was executed 40 times with the parameter uncertainties ΔI_{xx}, ΔI_{yy}, and ΔI_{zz} randomly generated with values ranging from $-I_{xx}$, $-I_{yy}$, and $-I_{zz}$ to I_{xx}, I_{yy}, and I_{zz}.

Table 5. Simulation with Parameter Uncertainties

$\boldsymbol{I_{xx}}$	0.05
$\boldsymbol{I_{yy}}$	0.05
$\boldsymbol{I_{zz}}$	0.24
$\boldsymbol{\Delta I_{xx}}$	$-I_{xx}$ to I_{xx}
$\boldsymbol{\Delta I_{yy}}$	$-I_{yy}$ to I_{yy}
$\boldsymbol{\Delta I_{zz}}$	$-I_{zz}$ to I_{zz}

The response of the quadrotor without parameter uncertainties is plotted in Figure 9. As seen in Figure 9, there are green and magenta vertical lines dividing the graph.

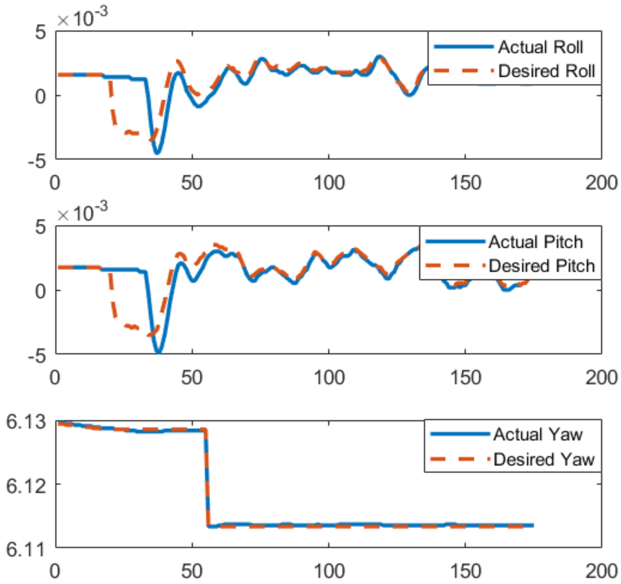

Figure 9. Simulated Roll, Pitch, and Yaw Response for SMC.

The average Roll, Pitch, and Yaw RMSE values are obtained with respect to the parameter uncertainties. As seen from Figure 10, the range of parameter uncertainties used in this setup yielded a maximum and minimum RMSE values of 0.141722% and 0.05291% respectively. Because of the low RMSE values, the quadrotor is considered to be robust and stable within the range of parameter uncertainties used.

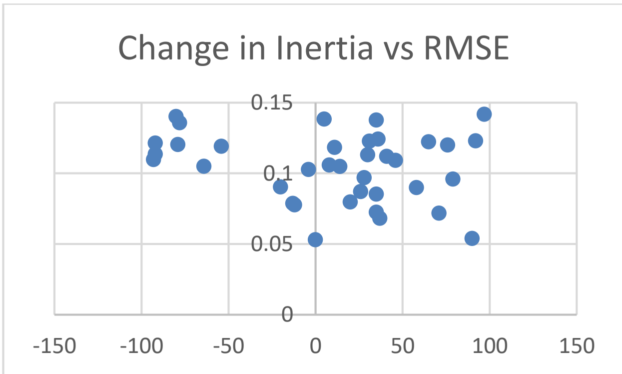

Figure 10. Plot of RMSE with respect to the Parameter Uncertainties.

Actual Implementation Results

After proving the robustness and stability of the Ardupilot source code with Sliding Mode control in the simulation, the source code is then implemented in an actual quadrotor. This is done because the testing of Sliding Mode Control in the actual hardware is highly relevant. Simulations are highly ideal as they do not include sensor errors, GPS deviations, outside disturbances, motor inconsistencies, and several others that occur in the actual flights. The purpose of the actual hardware flights is to confirm the usefulness and advantages of Sliding Mode Control in both theory and applications.

The frame used is the F450 multirotor frame kit. For the microcontroller, Pixhawk was used due to its compatibility with Ardupilot. Pixhawk version 2.4.6 was the model used for the quadrotor. After assembling the quadrotor, the Ardupilot source code is then installed into the Pixhawk controller. The code was modified in a way that the Sliding Mode Control and the PID can be easily interchanged via the modification of values of certain parameters in the parameter list of the Mission Planner.

Figure 11. Actual Quadrotor Testing

Both PID and SMC controllers were tested in an actual environment with constant external disturbance. The disturbance will be caused by the industrial fan placed beside the flight path of the quadrotor. This test would determine the stability of the system when disturbance is present. Presented is the roll, pitch, yaw response of PID and Sliding Mode Controller for backward movement. The similar flights are repeated four times as seen in Figures 12, 13, and 14 to reduce inconsistencies brought by GPS and sensors error. Figures 12-17 are all attitude graphs for both the PID and Sliding Mode Control algorithms that are used when flying the drone approximately 10 feet backwards (set in autopilot mode).

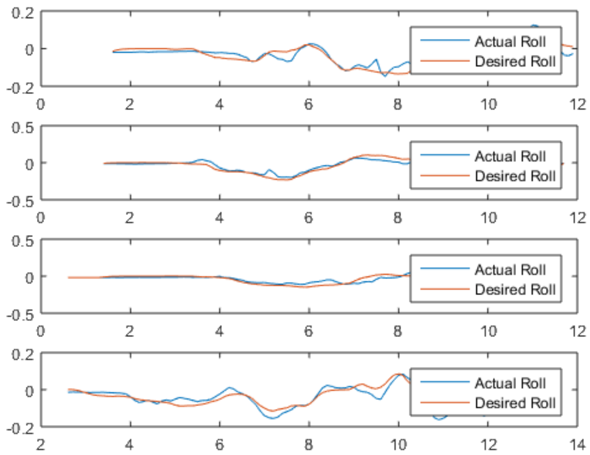

Figure 12. Roll Response for PID.

Figure 12 shows the roll movements of all the 4 flights that were done for the redundancy testing using PID control. This shows the angular movements around the x-axis of the drone as it follows the desired path.

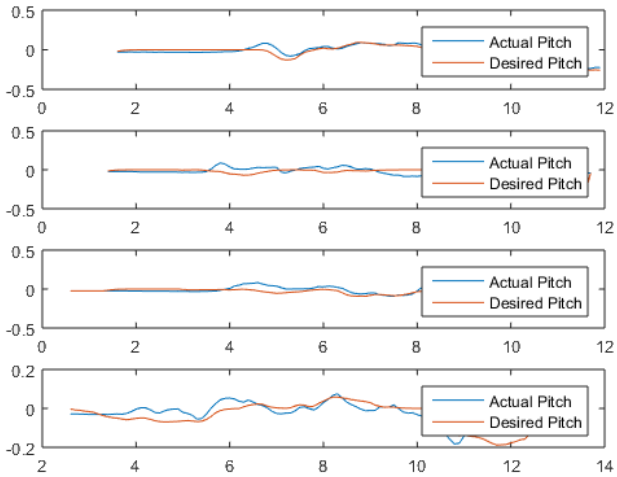

Figure 13. Pitch Response for PID.

Figure 13 shows the pitch movements of all the 4 flights that were done for the redundancy testing using PID control. This shows the angular movements around the y-axis of the drone as it follows the desired path.

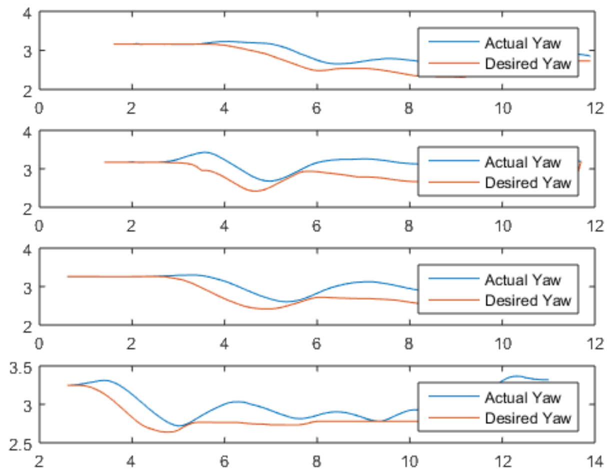

Figure 14. Yaw Response for PID.

Figure 14 shows the yaw movements of all the 4 flights that were done for the redundancy testing using PID control. This shows the angular movements around the z-axis of the drone as it follows the desired path.

A set of tuning parameters were used in implementing the Sliding Mode Control into the Quadrotor. A lot of researches such as [4] make use of tuning parameters that are obtained through trial and error. The use tuning parameters in these flights are also obtained using trial and error.

Table 6. SMC Tuning Parameter

	Λ	K	ε
φ	3	1	1
θ	4	0.5	1
ψ	8	0.5	1

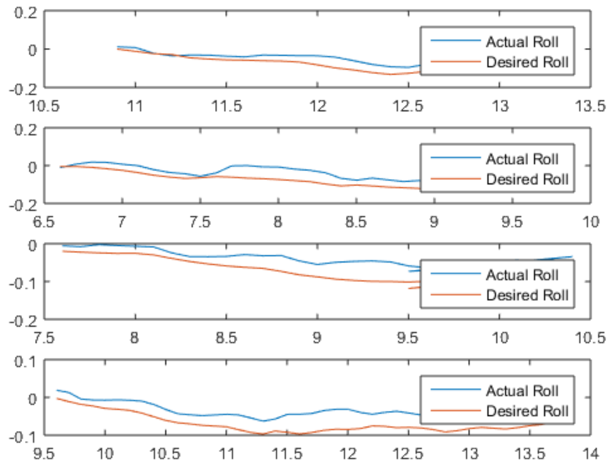

Figure 15. Roll Response for SMC.

Figure 15 shows the roll movements of all the 4 flights that were done for the redundancy testing using Sliding Mode Control. This shows the angular movements around the x-axis of the drone as it follows the desired path.

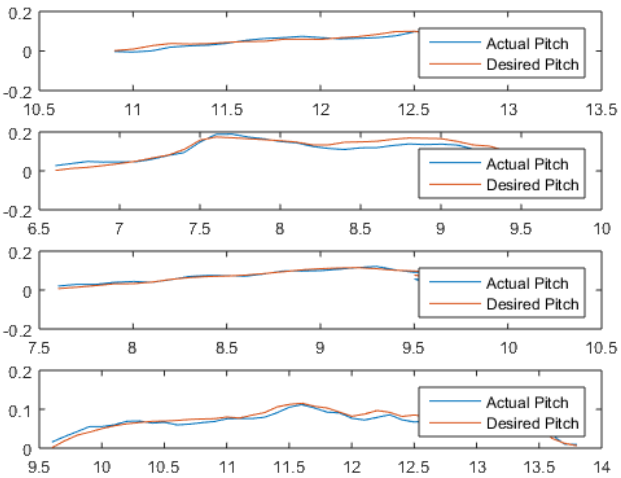

Figure 16. Pitch Response for SMC.

Figure 16 shows the pitch movements of all the 4 flights that were done for the redundancy testing using Sliding Mode Control. This shows the angular movements around the y-axis of the drone as it follows the desired path.

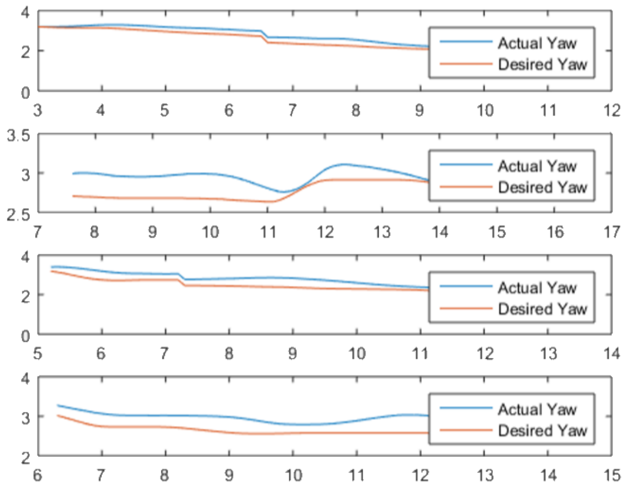

Figure 17. Yaw Response for SMC.

Figure 17 shows the yaw movements of all the 4 flights that were done for the redundancy testing using Sliding Mode Control. This shows the angular movements around the z-axis of the drone as it follows the desired path.

Table 7. SMC RMSE Results.

	PID	*SMC*
Roll	2.588324 %	1.546580 %
Pitch	4.553838 %	0.634243 %
Yaw	18.860183 %	13.466256 %

Based from the Table 7, it can be seen that the average error for roll, pitch and yaw of the Sliding Mode Control is smaller than that of PID. It can also be seen that there is a significant difference between the Roll and Pitch of PID and Sliding Mode Control. Therefore, it can be concluded that the Sliding Mode Control can track the desired values with more accuracy especially when there is disturbance.

Chattering Reduction

The high-frequency oscillations were tested with and without the chattering reduction equation. The response of the quadrotor movements, with and without chattering reduction, were plotted in Figure 18 to Figure 23.

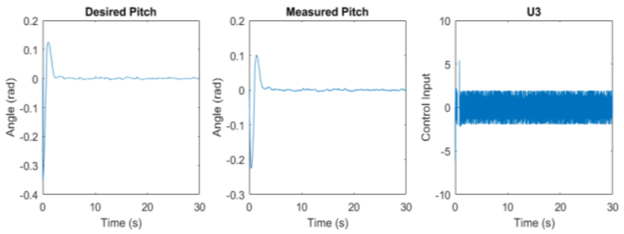

Figure 18. Roll Rotation without Chattering Reduction.

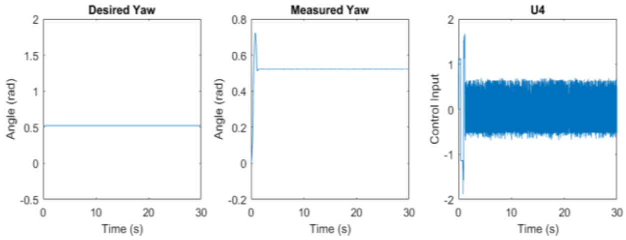

Figure 19. Pitch Rotation without Chattering Reduction.

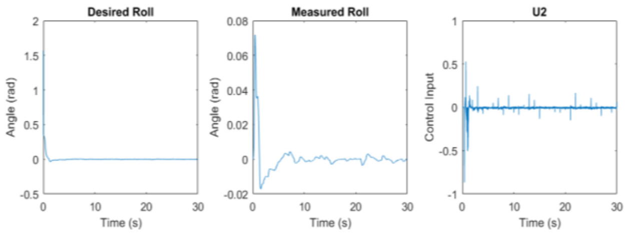

Figure 20. Yaw Rotation without Chattering Reduction.

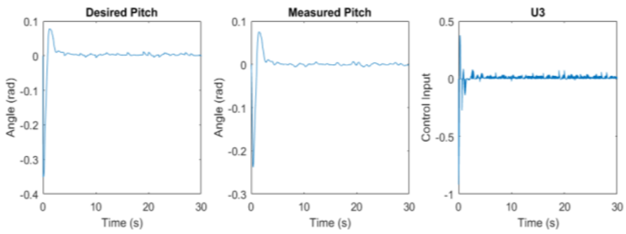

Figure 21. Roll Rotation with Chattering Reduction.

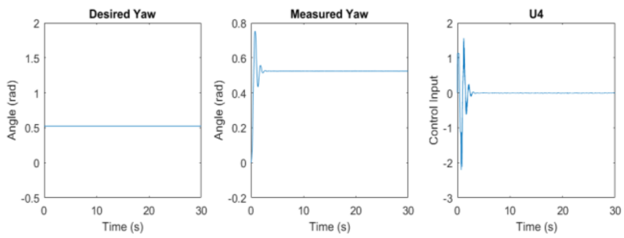

Figure 22. Pitch Rotation with Chattering Reduction.

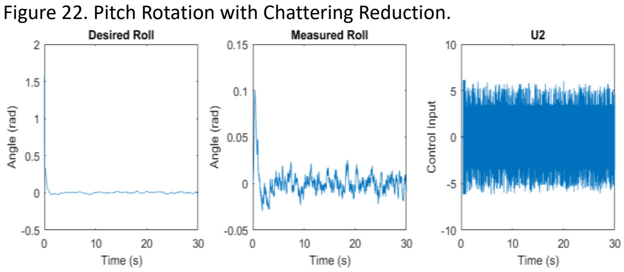

Figure 23. Yaw Rotation with Chattering Reduction.

As seen on the comparisons on Figure 18 to Figure 23, the inclusion of the Chattering Reduction equation reduces the high-frequency oscillations or chattering that occurred in the control inputs U2 – U4 in Figure 18 to Figure 20, however with the Chattering Reduction, the control inputs U2 – U4 in Figure 21 to Figure 23. The high-frequency oscillations in the control inputs are reduced while maintaining its accuracy in following the desired roll, pitch, and yaw angles. This is an improvement in the Sliding Mode Control that is useful in energy consumption.

Conclusion

Sliding mode control is highly compatible to a quadrotor system because this is a nonlinear system. Sliding mode control exhibits a more robust and stable system compared to proportional integral differential controller based on the simulation results. Because the Sliding Mode Controller controls the response of the non-linear system using its dynamic parameters to minimize the error, Sliding Mode Control produces a much more accurate system response whereas PID is a linear controller that do not depend on the dynamics of the system and only minimizes the error. However, because the Sliding Mode Controller relies on the dynamic parameters of the quadrotor, any changes in these parameters can cause a decrease in the accuracy of the system. This is tested by applying parameter uncertainties. By applying a range of values to the dynamic parameters of the system and observing the response, the performance of the quadrotor with these changes can be observed. As seen on Figure 9 and Figure 10, the range of parameter uncertainties applied on the system has very minimal change in the obtained RMSE. This implies that the system is robust within the specified range of parameter uncertainties.

Additionally, simulation results showed great results due to the absence of external disturbances. Because of this, tuning the parameters became easier. However, tuning in the actual became difficult because there are a lot of factors that affects the flight of the quadrotor like its asymmetrical figure. Nevertheless, the actual results still show that the Sliding Mode Controller performs better than PID when there is a disturbance.

Sliding Mode Control showed a lower roll RMSE with a difference of approximately 1% from the PID roll RMSE. Additionally, SMC pitch RMSE is also lower by approximately 3.92% from PID pitch RMSE, and SMC yaw RMSE is also lower than PID yaw RMSE by approximately 5.4%. Because of the values presented in Table 7, it can be concluded that Sliding Mode Control is more desirable in certain applications that are used in the presence of disturbances such as strong winds. These applications include photography, surveying, mapping, agriculture, and many more. In conclusion, using Sliding Mode Control in the ArduPilot code can allow PixHawk users to access a better-performing control algorithm that allows them to be used for the improvement of their applications. For future studies, because the difficulty in Sliding Mode Control is its trial-and-error tuning, an auto-tuning algorithm for the tuning parameters may be highly desirable for ease in use and quicker setup.

Acknowledgements

The authors would like to acknowledge the Department of Science and Technology – PCIEERD with project number 04254 and the University Research Coordination Office in De La Salle University with project number 01 IR 1TAY16-1TAY17 for funding and supporting this research.

References

[1] C. Cömert and C. Kasnakoğlu, "Comparing and Developing PID and Sliding Mode Controllers for Quadrotor", *International Journal of Mechanical Engineering and Robotics Research*, vol. 6, no. 3, pp. 194-199, 2017.
doi: 10.18178/ijmerr. 6.3.194-199

[2] D. Ye, Q. Wang and J. Li, "Fast Sliding Mode Tracking Control of Micro Quadrotor UAV", *2016 28th Chinese Control and Decision Conference (CCDC)*, Yinchuan , 28-30 May, 2016, pp. 632-636.
doi: 10.1109/CCDC.2016.7531062

[3] M. J. Reinoso, L. I. Minchala, P. Ortiz, D. F. Astudillo and D. Verdugo, "Trajectory Tracking of a Quadrotor Using Sliding Mode Control", *IEEE Latin America Transactions*, vol. 14, no. 5, 2016, pp. 2157-2167.
doi: 10.1109/TLA.2016.7530409

[4] W. Selby, "Sliding Mode Control for Translational Trajectory Following for a Quadrotor Vehicle", 2011.

[5] W. Z. Fum, "Implementation of Simulink Controller Design on Iris+ Quadrotor", Calhoun: The NPS Institunional Archive, 2015, pp. 58-66.
doi: 10.21236/ad1008930

[6] Y. Li and S. Song, "A Survey of Control Algorithms for Quadrotor Unmanned Helicopter", *2012 IEEE fifth International Conference on Advanced Computational Intelligence(ICACI)*, Nanjing, 18-20 Oct., 2012, pp. 365-369.
doi: 10.1109/ICACI.2012.6463187

[7] Z. Weidong, Z. Pengxiang, W. Changlong, C. Min, "Position and Attitude Tracking Control for a Quadrotor UAV Based on Terminal Sliding Mode Control", In proceedings of *the 34th Chinese Control Conference*, Hangzhou, 28-30 July, 2015, pp. 3398-3404.
doi: 10.1109/ChiCC.2015.7260164

[8] Z. Samir, "Sliding Mode Control Strategy for a 6 DOF Quadrotor Helicopter", *Journal of Electrical Engineering*, pp. 1-5.

[9] H. Jayakody and J. Katupitiya, "Sliding Mode Based Position and Attitude Controller for a Vectored Thrust Aerial Vehicle", In proceedings of *Australasian Conference on Robotics, University of New South Wales*, Sydney, 2-4 Dec., 2013.

[10] M. K. Shaik and J. F. Whidborne, "Robust Sliding Mode Control of a Quadrotor", *2016 UKACC 11th International Conference on Control*, Belfast, 31 Aug.-2 Sept., 2016.
doi: 10.1109/CONTROL.2016.7737529

[11] P. Dey, S. R. Kurode, and R. Ramachandran, "Robust Attitude Control of Quadrotor using Sliding Mode", *2016 International Conference on Automatic Control and Dynamic Optimization Techniques,* Pune, 9-10 Sept., 2016 , pp. 268-272.
doi: 10.1109/ICACDOT.2016.7877592

[12] A. Bartoszewicz and J. Żuk, "Sliding mode control — Basic concepts and current trends", *2010 IEEE International Symposium on Industrial Electronics,* Bari, 4-7 July, 2010, pp. 3772-3777.
doi: 10.1109/ISIE.2010.5637990

[13] J. J. Xiong and G. Zhang, " Sliding Mode Control for a Quadrotor UAV with Parameter Uncertainties", *2016 2nd International Conference on Control, Automation and Robotics (ICCAR),* 28-30 April, 2016, pp. 207-212.
doi: 10.1109/ICCAR.2016.7486727