A Second-Order Sliding Mode Controller Tuning Employing Particle Swarm Optimization

Sami Mahfoudhi[1]   **Mohammed Abdallah Khodja[2,3]***   **Faisal Omar Mahroogi[4]**

*[1] Department of Management Information Systems, College of Business and Economics,
Qassim University Buraidah, Al Qassim, Saudi Arabia*
*[2] Department of Electrical Engineering, College of Engineering,
Qassim University, P.O.B. 6677, Buraidah, 51452, Saudi Arabia*
[3]Department of Electrical Engineering, Faculty of Technology, M'sila University, Algeria.
[4]Department of Mechanical Engineering, Faculty of Engineering, Islamic University of Madinah, Saudi Arabia
* Corresponding author's Email: mohammed.khodja@qec.edu.sa

Abstract: The intention of this research is to provide tuning for the coefficients of the switching sliding manifold for the second-order sliding mode employing Particle Swarm Optimization. This research was carried out in noisy conditions with a homemade prototype apparatus employing cheap componentry. A Teensy development board was used as flight controller for a quadcopter, with the second-order sliding mode controller being used for attitude stabilization. Other research has offered confirmation for the control law in both theory and simulation where they identify the nonlinear coefficients based on Hurwitz stability analysis and linearization around equilibrium point. But in this research, we shall be focusing upon identification of the coefficients for the switching siding manifold of the second-order siding employing Particle Swarm Optimization, the obtained coefficient validated both practically and experimentally; this has never previously been undertaken. New data samples will also be provided regarding the short time execution for the physical system; this data will prove useful for future applications using artificial intelligence. The outcomes of the research demonstrate that the proposed tuning method for second order sliding mode controller confirm its robustness and effectiveness both in simulation and experiment.

Keywords: Flight controller, Sliding mode control, Teensy development board, Quadcopter.

1. Introduction

It is recognized that an increasing number of autopilot algorithms have been developed as a result of ongoing research in this field. Some commercial autopilot algorithms are Pixhawk [1] and CC3D [2]. Autopilot systems generally employ extremely complex algorithms for the best and smoothest flight control. The best understanding of flight systems can be gained by building a homemade quadcopter, in order to appreciate all the chief requirements of flight systems. We have used homemade autopilot systems in our previous published research [3, 4]. Our homemade autopilot system employs a Teensy 3.2 development board [5], centered around a Cortex-M4

microcontroller with 64 kB of RAM, 256 kB of Flash Memory, and a rate speed of 96 MHz.

Particle Swarm Optimization (PSO) represents a population-based methodology for deriving the best solutions for objective functions [6]. It was first proposed by Kennedy and Eberhart (1995) [7]. The "swarm" comprises "n" particles, with each particle's position representing the possible space solution. A particle varies in condition depending upon how its inertia, its position, and its velocity are controlled. Every particle in the swarm is influenced by its unique experience and also its optimal position within its environment. The PSO has a number of algorithms [8, 9]. For this research the fitness function (objective function) is the integral of square error (ISE) used with the construction coefficient

methodology [6, 10,11] as per our previous research [4, 12] where it obtained excellent outcomes.

There have been numerous proposals for SMC methodologies to be employed in flight controllers for quadcopters [13-15]. Researchers have also been attracted to the concept of employing sliding mode approaches to develop controllers; for example [16] suggested a fault-tolerant control methodology employing adaptive sliding mode control approaches for unmanned aerial vehicles (UAV), particularly quadcopters, to mitigate faults in actuators and the uncertainty in the system. In [17] investigation was undertaken of a dual-timescale autopilot for aircraft with damaged wings using nonlinear adaptive sliding mode approaches. Adaptive flight control strategies are employed for tracking outer-loop angle commands, allowing for the influence of wing damage. A pair of specific control strategies employing the adaptive sliding mode has been created for outer- and inner-loop dynamics. [18] suggests the deployment of a type-two fuzzy adaptive sliding mode control scheme to maintain stability for hypersonic aircraft when the parameters are uncertain by employing a combination of adaptive sliding mode control and the interval type-two fuzzy approach. We should never neglect the part that PSO can play in finding optimal parameters as shown in [19]. This research intends to examine the way an FOPID controller can be tuned with intelligent soft computing techniques, e.g. PSO and Differential Evolution (DE) to design a fractional order PID control unit. PSO can also be used to tune the parameters for controllers, e.g. in [20] where PSO was used to validate obtained PID parameters experimentally. [21] Proposed new strategies for employing a second sliding mode control with quadcopters; this requires obtaining nonlinear coefficients, they use Hurwitz Stability analysis and linearization theory for nonlinear coefficients identification but in our work suggest use Particle Swarm Optimization (PSO) for nonlinear coefficients obtaining. This research proposes that PSO should be used for the determining of nonlinear coefficients; we will undertake simulations and experimental testing to demonstrate that PSO can efficiently obtain coefficients for the switching sliding manifold.

This paper comprises the following sections: Section 1: Introduction; Section 2: Overview of Pixhawk, CC3D, and a description of the homemade quadcopter; Section 3: Discussion of results; Section 4: Conclusion.

2. Flight controller systems

2.1 Pixhawk

The Pixhawk flight controller (Fig. 1) [1] is regarded as a sophisticated flight controller that employs algorithms to achieve optimal smoothness in flight. The algorithm it employs was created by Lorenz Meier in 2008 at the ETH Zürich University. Pixhawk 1 is an open hardware project providing economical flight controllers for hobbyists and researchers; it can be used to control rovers, multicopters, and fixed-wing aircraft.

Main features are:

32bit STM32F427 Cortex-M4F® core with FPU , 168 MHz / 252 MIPS ,256 KB RAM ,2 MB Flash (fully accessible), 32 bit STM32F103 failsafe co-processor.

2.2 Flight controller F722-STD

Is open source autopilot, low cost controller and high flight performance. The CC3D [2] is an open-source autopilot that offers economical control and high levels of performance. In order to examine professional-standard flight controllers we have to have an understanding of all the various parts of this type of controller and their role in flight. We believe that the optimal way of achieving this is to create our own homemade flight controller.
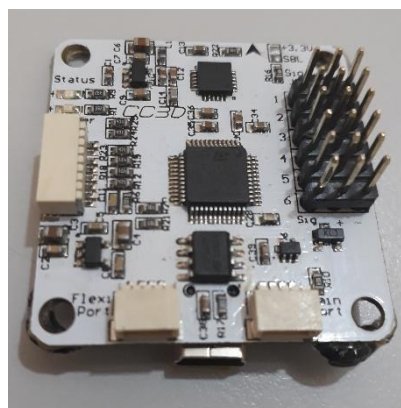

Figure. 1 Pixhawk flight controller

Figure. 2 CC3D flight controller

2.3 Homemade flight controller

The flight controller is the central nervous system of a quadcopter, undertaking all the required actions to allow for aircraft stability and controllability. We employed a Teensy development board and Arduino as illustrated by Figs. 3 and 4.

An identical experimental configuration was used in previous published work [3, 4], except that in this research a second-order sliding controller is employed; the figures illustrate the way in which the chief components are interconnected.

The Teensy V3.2 development board [5] is centered on the 32-bit ARM Cortex-M4 72 MHz CPU. It is a broadband-ready development board with many features. It is pre-flashed with a bootloader so can be programmed via the integral USB connector without the need for an external programmer. It can be programmed with the operator's user choice employing C or the Teensy duino addon for Arduino IDE can be used with Arduino sketches being written for Teensy by the operator.

Arduino due [26]: The Arduino Due is a microcontroller board using the Atmel SAM3X8E ARM Cortex-M3 CPU. It represents the first Arduino board to employ a 32-bit ARM core microcontroller.
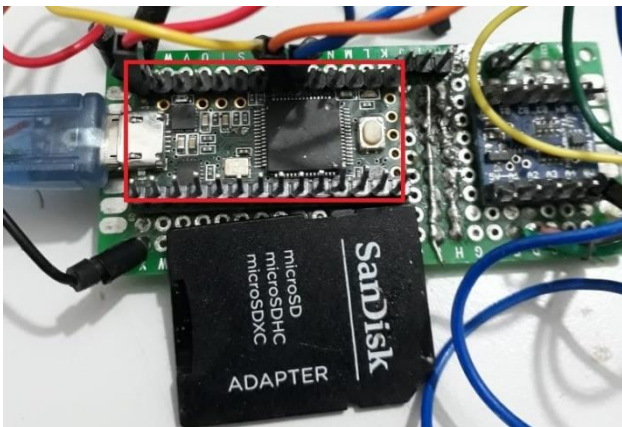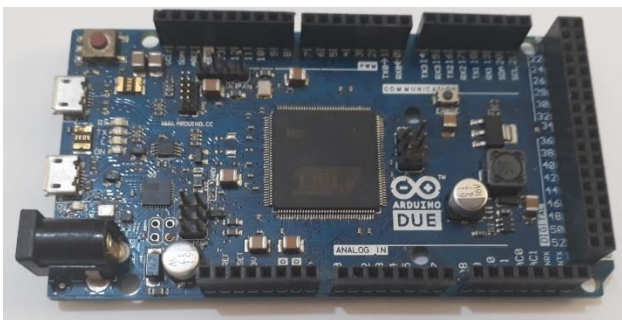
Our quadcopter system comprises three central electronic units, these being the microcontroller unit (MCU) to implement the controller, electronic speed controllers (ESCs) to control the speed of the brushless DC motors, and the inertial measurement unit (IMU) supplying orientation data allowing the state to be estimated. This chapter describes the reason for the selection of the MCU and ESCs, their functionality, and how they operate. The essential interconnections between various electronic hardware modules are illustrated in Fig. 5. The fully built quadcopter is shown in the pictures in Figs. 6 and 7.

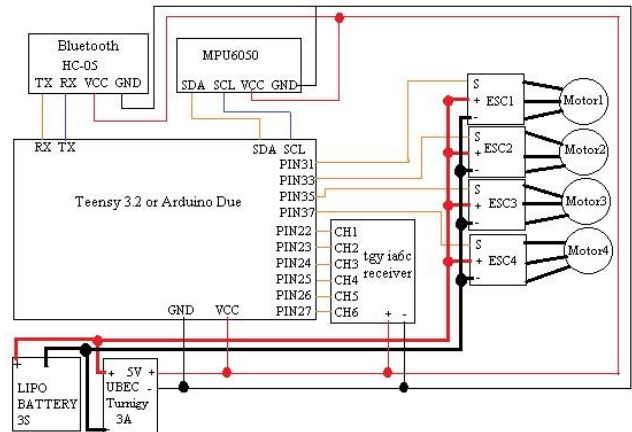Figure. 5 Hardware diagram of interconnection between electronics parts

Figure.3 Teensy development board in red rectangle

Figure. 4 Arduino due [26]

Figure. 6 The realized quadcopter insight view

Figure. 7 Top view of the realized quadcopter

3. Tuning switching coefficient using PSO

Kennedy and Eberhart first described Particle Swarm Optimization (PSO) in 1995 [22]. It is based on Swarm Intelligence methodologies to search for optimal values.

Particle Swarm Optimization employs a number of algorithms [22-24]. In this research the fitness functions used the integral of square error (ISE) and we chose the constriction coefficient [22, 25] methodology. The algorithms have guaranteed convergence, clearly demonstrated in Fig. 9 [22], with the PSO being formulated in the following manner:

$$v_{ij}(t+1) = X\left[v_{ij}(t) + \phi_1\left(y_{ij}(t) - x_{ij}(t)\right) + \phi_2\left(\hat{y}_j - x_{ij}(t)\right)\right] \tag{1}$$

$$X = \frac{2k}{\left|2 - \phi - \sqrt{\phi(\phi - 4)}\right|} \tag{2}$$

With: $\phi = \phi_1 + \phi_2$, $\phi_1 = c_1 r_1$, $\phi_2 = c_2 r_2$, $\quad \phi \geq 4$, $k \in [0\ 1]$

c_1 and c_2 are the acceleration coefficients, and r_1 and r_2 are random values in [0 1].

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{3}$$

For performance verification we employed the integral of square error (ISE) as comparison criteria, defined thus:

$$ISE = \int e^2(t)dt \tag{4}$$

e= The desired angle –The actulle angle

Thus in order to optimize the controller tuning we employed criteria (4) for the objective function to find a collection of parameters *(η3, ε3, c3, c4)* that would permit the control system to meet the necessary performance criteria.

Fig. 8 shows how PSO was deployed for tuning four parameters. Every particle represents a controller solution. When all four parameters were good then the system responded well, achieving the minimum performance criteria as shown in Eq. (4). Fig. 8 shows the overall strategy for dealing with optimization difficulties.

Particle Swarm Optimization it should initialized by searching interval in our case [0.1 8] sets as searching interval for four coefficients *(η3, ε3, c3, c4)*, PSO will generates many values and star search the best one. For each four values of the coefficients will

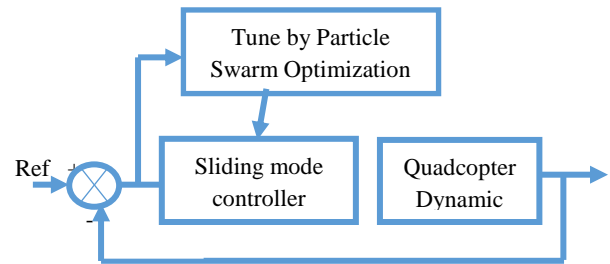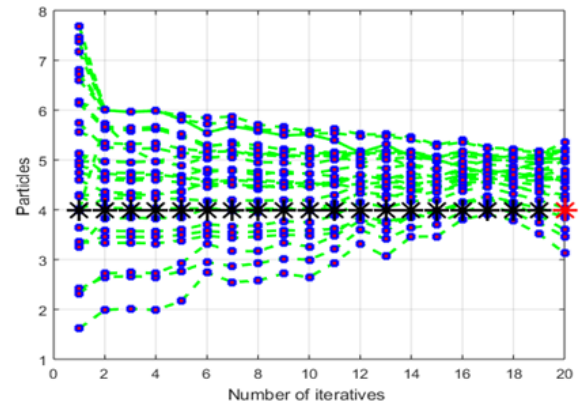Figure. 8 The general strategy of optimization
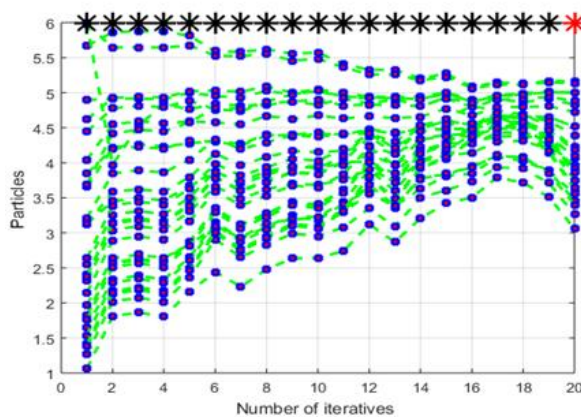
Figure. 9 Movement of particles during the search for the optimal parameters (η_3, ε_3): (a) η_3 and (b) ε_3

substituted in the controller for get response and amount of quadratic error of trajectory tracing and sure lower error will be best coefficients values.

The PSO has to calculate particle numbers and the iterative number for calculation of PID gains. In this instance particle numbers were set at 30 with the iterative number at 20. The particle movements as they search for optimized second order sliding mode control output parameters in accordance with objective function is illustrated in Fig. 9. Swarms are represented by blue points, local base particles are represented by black points, and global best particles are represented by red points.

4. Results and discussion

The methods suggested here for identification of nonlinear coefficients has been subjected to testing in the experiments and simulations detailed to find whether the control scheme employed by this research is both valid and efficient. [21] Suggests the use of a sliding mode controller; however, this form of controller has to be able to undertake identification of the coefficient with switching sliding manifold of second-order sliding. In [21] Hurwitz stability analysis and linearization theory are employed to resolve this problem. In this research we proposed using Particle Swarm Optimization (PSO) for identification of the nonlinear coefficients of the second sliding mode controller in order to achieve attitude stabilization of the realized quadcopter. Table 1 illustrates the parameters of the completed quadcopter [3, 4]. Eq. (5) shows the control law where proposed in [21]; Fig. 10 shows the control strategy.

As this research is only focused on attitude stabilization. d_1, d_2, d_3, d_4, are disturbance terms that will be considered as zero.

$$
\begin{cases}
u_1 = m\dfrac{c_z(\dot{z}_d - \dot{z}) + \ddot{z}_d + g + d_1 + \varepsilon_1\,\mathrm{sgn}(s_1) + n_1 s_1}{\cos\varphi\cos\theta} \\[2mm]
u_4 = \dfrac{I_z}{C}\begin{bmatrix} c_\psi(\psi_d - \psi) + \ddot{\psi}_d + \\ d_2 + \varepsilon_1\,\mathrm{sgn}(s_2) + n_2 s_2 \end{bmatrix} \\[3mm]
u_3 = \dfrac{I_y}{l}\begin{bmatrix} \dfrac{c_1}{c_3}(\ddot{x}_d - \ddot{x}) + \dfrac{c_2}{c_3}(\dot{x}_d - \dot{x}) + \ddot{\theta}_d + \\ \dfrac{c_4}{c_3}(\dot{\theta}_d - \dot{\theta}) + d_3 + \dfrac{1}{c_3}(\varepsilon_3\,\mathrm{sgn}(s_3) + n_3 s_3) \end{bmatrix} \\[4mm]
u_2 = \dfrac{I_x}{l}\begin{bmatrix} \dfrac{c_5}{c_7}(\ddot{y}_d - \ddot{y}) + \dfrac{c_6}{c_7}(\dot{y}_d - \dot{y}) + \ddot{\varphi}_d + \\ \dfrac{c_8}{c_7}(\dot{\varphi}_d - \dot{\varphi}) + d_3 + \dfrac{1}{c_7}(\varepsilon_4\,\mathrm{sgn}(s_4) + n_4 s_4) \end{bmatrix}
\end{cases}
$$
(5)

Where:

$$
\begin{cases}
s_1 = c_z(z_d - z) + (\dot{z}_d - \dot{z}) \\
s_2 = c_\psi(\psi_d - \psi) + (\dot{\psi}_d - \dot{\psi}) \\
s_3 = \begin{bmatrix} c_1(\ddot{x}_d - \ddot{x}) + c_2(\dot{x}_d - \dot{x}) + \\ c_3(\dot{\theta}_d - \dot{\theta}) + c_4(\theta_d - \theta) \end{bmatrix} \\
s_4 = \begin{bmatrix} c_5(\ddot{y}_d - \ddot{y}) + c_6(\dot{y}_d - \dot{y}) + \\ c_7(\dot{\varphi}_d - \dot{\varphi}) + c_4(\varphi_d - \varphi) \end{bmatrix}
\end{cases}
$$

Table 1 [3, 4] illustrates the finished quadcopter's dynamic parameters. The second-order sliding mode's parameters employ PSO only for the roll and

Table 1. Quadrotor parameters

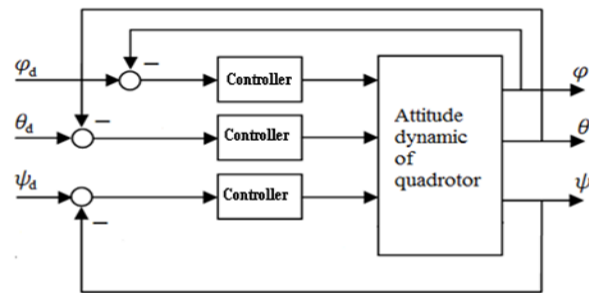|	
Calculated Trust Coefficient	$8.5*10^{-7}$
Trust Coefficient C_T	$8.8*10^{-7}$
Drag Coefficient C_D	$11.3*10^{-8}$
l	0.127 meter
Mass	0.49Kg
I_x	0.00102 Kg*m^2
I_y	0.00059 Kg*m^2
I_z	0.00138 Kg*m^2

Figure. 10 Control strategy for attitude stabilization [22]: θ_d :is the desired roll Angle. φ_d: is the desired pitch Angle. Ψ_d:is the desired yaw angle

Table 2. Roll/pitch controller's parameters using PSO

Variables	Values	Variables	Values
c_3	0.3	c_7	0.3
c_4	1.8	c_8	1.8
e_3	6	e_4	6
n_3	4	n_4	4

Table 3. Controllers parameters [21]

Variables	Values	Variables	Values
c_z	1	c_z	1
e_1	0.8	e_2	0.8
n_1	2	n_2	2
c_1	$11m/(u_1\,cos\varphi\,cos\psi)$	c_5	c_1
c_2	$6m/(u_1\,cos\varphi\,cos\psi)$	c_6	c_2

pitch axes as quadcopters are symmetrical so the values will be equal. For the other axes (z /yaw axis) the parameters will be retained as [21]. The following Table 2 illustrates the parameters that were obtained by employing PSO. Table 3 offers the others parameters identified in [21].

Figs. 11 and 12 illustrate the results of the simulation of simple directory tracking with smooth tracking related to the attitude angle and x-axis. It can be seen that the x-axis took five seconds to acquire the optimal position but the attitude angle was acquired in under 0.9 seconds.

The results tested for 100 seconds of run time are shown in Figs. 13, 14, and 15: the controller achieves stabilization of attitude for the quadcopter on the equilibrium point and actually managed to track the trajectory effectively. The simulation results in the original paper demonstrates that the angles returned

to equilibrium within a second, but in these experimental findings the equilibrium angles are found in 0.9 seconds. The appendix contains sample data from Fig. 14 related to durations between 20 seconds and 33 seconds for use in future AI projects. The suggested controller has been proved through experimentation to be both robust and effective.

Figs. 16, 17, and 18 illustrate the sliding surface as it varies over time in the course of the response recording illustrated in Figs. 13, 14, and 15. It is clear that the obtained sliding surface's global experimental behavior has similarities to the behavior encountered in the simulations detailed in our original paper [21].
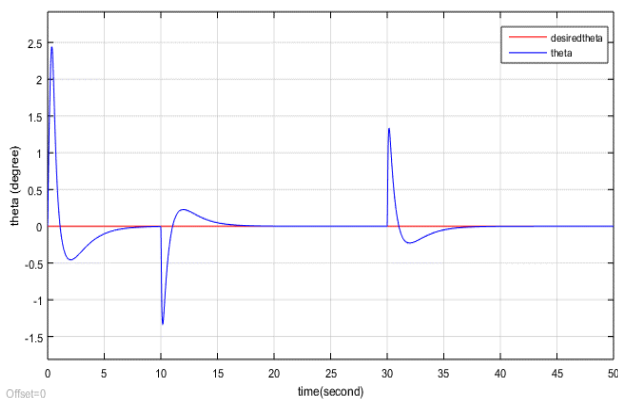

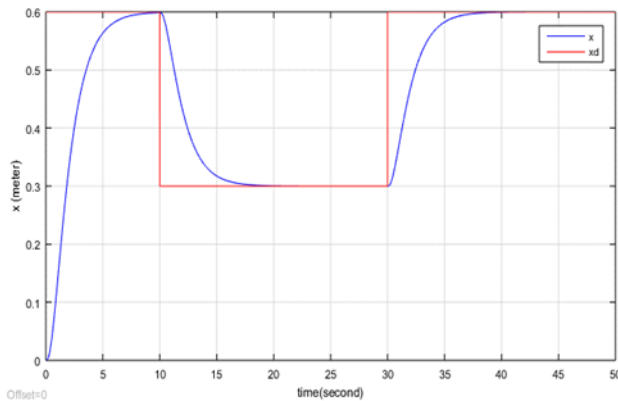Figure. 11 Roll angle stabilization
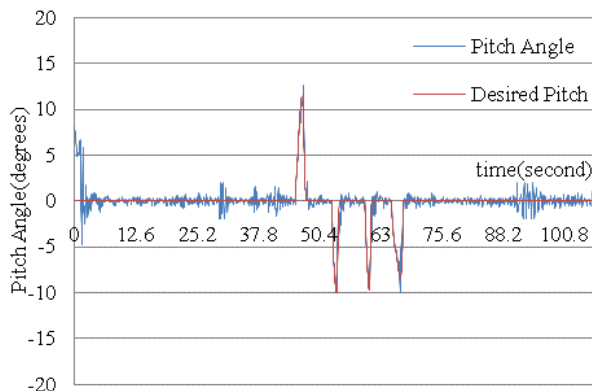

Figure. 12 Trajectory tracking on X-axis
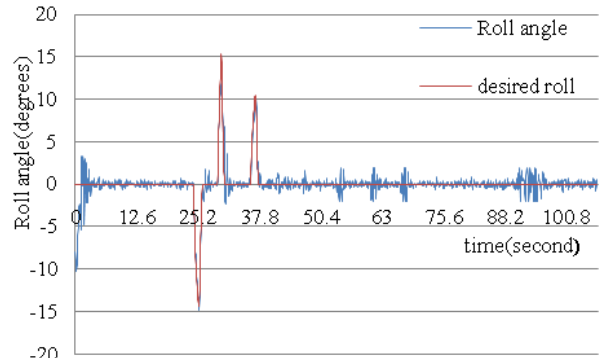

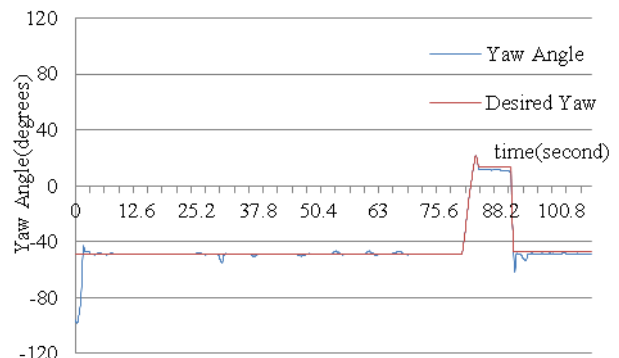Figure. 13 Pitch angle response

Figure. 14 Roll angle response

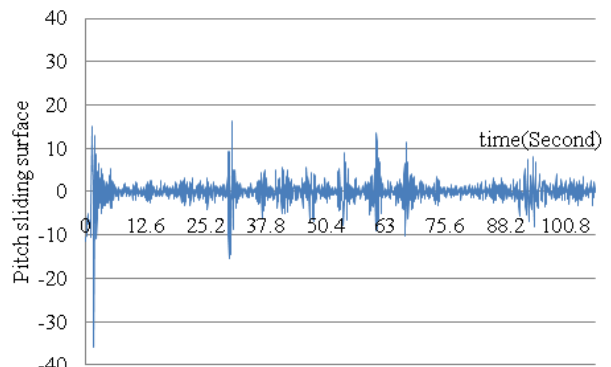Figure. 15 Pitch angle response

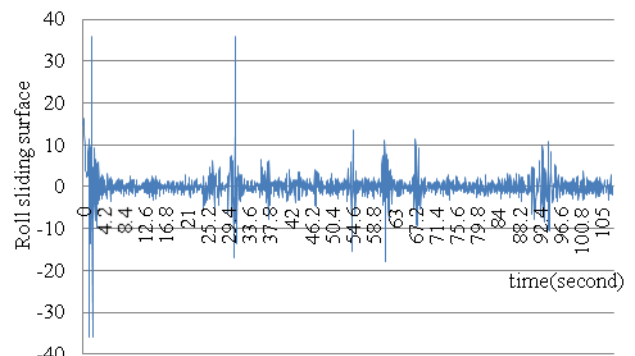Figure. 16 Surface (s_3) of Pitch control (u_3) in real time

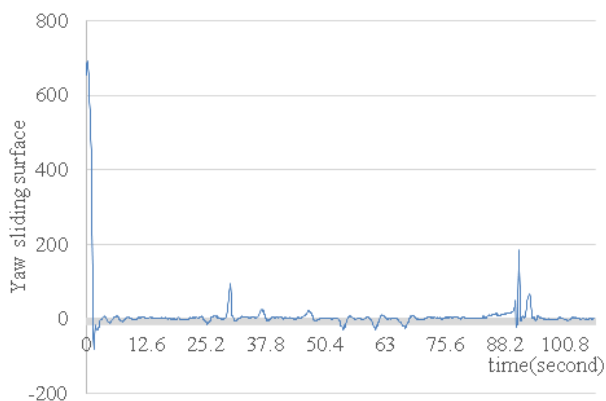Figure. 17 Surface(s_4) of Roll control (u_2) in real time

Figure. 18 Surface (s_2) of Pitch control (u_4) in real time

5. Conclusion

This paper has presented an illustration of the ways in which building a quadcopter is useful in gaining an understanding of the foundational technology; it provides insider knowledge of the operation of flight controllers that will allow us to work with the most sophisticated open hardware flight controllers. The homemade autopilot system, using the Teensy development board with a clock frequency of approximately 100 MHz, underwent testing using a second-order sliding mode control to control a quadcopter's attitude. The data harvested in real time from our prototype has been provided as a dataset to be employed with Artificial intelligence applications in future and with same law control change it to fractional second order sliding mode as another contribution. These results have shown that Particle Swarm Optimization is effective as a means of finding coefficients for a nonlinear switching sliding manifold where in other research they employ Hurwitz stability analysis and linearization theory. This suggested Particle Swarm Optimization tuning of the coefficients of the switching sliding manifold for the second-order sliding mode control strategy has been shown to be both robust and effective through simulation and experimentation.

References

[1] https://pixhawk.org
[2] https://opwiki.readthedocs.io/en/latest/user_ma nual/cc3d/cc3d.html
[3] M. Khodja, C. Larbes, N. Ramzan, and A. Ibrahim, "Implementation of Heuristical PID Tuning for Nonlinear System Control", *International Review of Automatic Control*, Vol.12, No.2, pp. 108-114, 2019.
[4] M. A. Khodja, M. Tadjine, M. S. Boucherit, and M. Benzaoui, "Experimental dynamics identification and control of a quadcopter", In: *Proc. of 2017 6th International Conference on Systems and Control*, pp. 498-502, 2017.
[5] https://www.pjrc.com/store/teensy32.html
[6] Engelbrecht AP. *Computational Intelligence: An Introduction*. John Wiley & Sons; 2007.
[7] R. Eberhart and J. Kennedy, "Particle swarm optimization", In: *Proc. of the IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942-1948, 1995.
[8] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial", *Chemometrics and Intelligent Laboratory Systems*, Vol. 149, Part. B, pp.153-65, 2015.
[9] P. R. Dian, M. S. Siti, and S. Y. Siti, "Particle swarm optimization: Technique, system and challenges", *International Journal of Computer Applications*, Vol. 14, No. 1, pp. 19-26, 2011.
[10] G. Štimac, S. Braut, and R. Žigulić, "Comparative analysis of PSO algorithms for PID controller tuning", *Chinese Journal of Mechanical Engineering*, Vol. 27, No. 5, pp. 928-36, 2014.
[11] K. Menghour and L. Souici-Meslati, "Hybrid aco-pso based approaches for feature selection", *International Journal of Intelligent Engineering and Systems*, Vol. 9, No. 3, pp. 65-79, 2016.
[12] M. A. Khodjal, M. Tadjine, M. S. Boucherit, and K. Busawon, "Optimization of a proportional derivative (PD) fuzzy controller using the particle swarm optimization (PSO) technique for a 3DOF robot manipulator", *The Mediterranean Journal of Measurement and Control*, Vol. 12, No. 4, pp. 670-7, 2016.
[13] O. V. Dhakad and V. Kumar, "Fractional Order Sliding-Mode Controller for Quadcopter", *Advances in Interdisciplinary Engineering*, pp. pp. 381-392, 2019.
[14] A. Alkamachi and E. Erçelebi, "A proportional derivative sliding mode control for an overactuated quadcopter", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 4, pp. 1354-63, 2019.
[15] A. Imani and B. Beigzadeh, "Robust control of spacecraft rendezvous on elliptical orbits: Optimal sliding mode and backstepping sliding mode approaches", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 230, No. 10, pp. 1975-89, 2016.
[16] N. P. Nguyen and S. K. Hong, "Fault-tolerant control of quadcopter UAVs using robust adaptive sliding mode approach", *Energies*, Vol. 12, No. 1, pp. 95, 2019.

[17] D. Asadi and S. A. Bagherzadeh, "Nonlinear adaptive sliding mode tracking control of an airplane with wing damage", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 232, No. 8, pp. 1405-20, 2018.

[18] X. Jiao, B. Fidan, J. Jiang, and M. Kamel, "Type-2 fuzzy adaptive sliding mode control of hypersonic flight", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 8, pp. 2731-44, 2019.

[19] A. Idir, M. Kidouche, Y. Bensafia, K. Khettab, and S. A. Tadjer, "Speed Control of DC Motor Using PID and FOPID Controllers Based on Differential Evolution and PSO", *International Journal of Intelligent Engineering and Systems*, Vol. 11, No. 4, pp. 241-249, 2018.

[20] M. A. Khodja, M. Tadjine, M. S. Boucherit, and M. Benzaoui, "Tuning PID attitude stabilization of a quadrotor using particle swarm optimization (experimental)", *International Journal for Simulation and Multidisciplinary Design Optimization*, Vol. 8, pp. A8, 2017.

[21] E.H. Zheng, J. J. Xiong, and J. L. Luob, "Second order sliding mode control for a quadrotor UAV", *ISA Transactions*, Vol. 53, No. 4, pp. 1350-1356, 2014.

[22] M. I. Menhas, L. Wang, M. Fei, and H. Pan, "Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design", *Expert systems with applications*, Vol. 39, No. 4, pp. 4390-401, 2012.

[23] J. M. Mendel, *Uncertain rule-based fuzzy logic systems: Introduction and new. Directions*, Ed. USA: Prentice Hall. 25-00, 2000.

[24] B. Nadir and O. Mohammed, "Optimization the Trajectories of Robot Manipulators A long Specified Task", *International Journal of Intelligent Engineering and Systems*, Vol. 11, No. 1, pp. 11-19, 2018.

[25] R. H. Al-Rubayi, M. K. Abd, and F. M. Flaih, "A New Enhancement on PSO Algorithm for Combined Economic-Emission Load Dispatch Issues", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 1, pp. 77-85, 2020.

[26] https://store.arduino.cc/usa/due

Appendix

$e = \theta_d - \theta, \ de \dot{=} \theta_d - \dot{\theta}$

time (s)	θ	θ_d	e	de	U_3
20.1	-0.08	0	0.08	-0.94	-0.33
20.16	0.11	0	-0.11	-4.23	-2.25
20.22	0.15	0	-0.15	-0.48	-0.47
20.28	0.16	0	-0.16	-1.31	-0.91
20.34	0.41	0	-0.41	-4.29	-2.78
20.4	0.42	0	-0.42	2.06	0.35
20.46	0.11	0	-0.11	2.34	0.99
20.52	0.14	0	-0.14	-3.74	-2.08
20.58	0.33	0	-0.33	0.06	-0.5
20.64	-0.05	0	0.05	8.69	4.38
20.7	-0.48	0	0.48	2.2	1.86
20.76	-0.36	0	0.36	-3.92	-1.36
20.82	-0.11	0	0.11	-2.16	-0.9
20.88	-0.15	0	0.15	0.13	0.3
20.94	-0.19	0	0.19	-1.46	-0.42
21	-0.1	0	0.1	-2.64	-1.15
21.06	-0.09	0	0.09	-0.22	0.03
21.12	-0.01	0	0.01	-2.12	-1.02
21.18	0.14	0	-0.14	-0.21	-0.33
21.24	0.04	0	-0.04	2.65	1.24
21.3	-0.09	0	0.09	0.09	0.18
21.36	-0.08	0	0.08	1.38	0.81
21.42	-0.4	0	0.4	5.19	3.21
21.48	-0.66	0	0.66	-0.76	0.68
21.54	-0.38	0	0.38	-7.04	-2.86
21.6	0.22	0	-0.22	-7.26	-3.92
21.66	0.66	0	-0.66	-1.61	-1.85
21.72	0.53	0	-0.53	1.34	-0.19
21.78	0.38	0	-0.38	1.68	0.22
21.84	0.27	0	-0.27	3.82	1.45
21.9	-0.07	0	0.07	4.99	2.58
21.96	-0.4	0	0.4	2.81	2.04
22.02	-0.33	0	0.33	-3.97	-1.44
22.08	-0.03	0	0.03	-2.71	-1.29
22.14	0.09	0	-0.09	-0.82	-0.55
22.2	0.18	0	-0.18	-2.1	-1.32
22.26	0.39	0	-0.39	-3.35	-2.27
22.32	0.4	0	-0.4	1.04	-0.13
22.38	0.1	0	-0.1	4.34	1.99
22.44	-0.11	0	0.11	-0.18	0.08
22.5	0.07	0	-0.07	-4.03	-2.1
22.56	0.21	0	-0.21	0.07	-0.31
22.62	0.09	0	-0.09	1.6	0.65
22.68	-0.05	0	0.05	0.02	0.09

22.74	-0.05	0	0.05	-1.2	-0.51	25.38	-14.8	-14.2	0.52	5.63	2.23
22.8	-0.05	0	0.05	0.02	0.09	25.44	-14.6	-13.6	1.01	0.77	0.67
22.86	-0.15	0	0.15	3.35	1.9	25.5	-13.2	-12.5	0.76	-10.35	-5.1
22.92	-0.47	0	0.47	4.47	2.96	25.56	-11.4	-10.0	1.34	8.79	5.52
22.98	-0.63	0	0.63	-1.14	0.45	25.62	-9.21	-8.36	0.84	-12.74	-5.75
23.04	-0.37	0	0.37	-5.96	-2.35	25.68	-7.09	-6.24	0.85	2.73	2.1
23.1	-0.03	0	0.03	-3.45	-1.66	25.74	-5.53	-5.06	0.48	-7.94	-3.64
23.16	0.17	0	-0.17	-2.1	-1.31	25.8	-4.24	-3.55	0.69	2.74	2.11
23.22	0.21	0	-0.21	0.1	-0.28	25.86	-3.32	-2.71	0.61	-2.02	-0.29
23.28	0.1	0	-0.1	1.37	0.51	25.92	-2.66	-1.65	1.01	5.44	4.15
23.34	0.09	0	-0.09	-1.4	-0.84	25.98	-2.01	-1.05	0.95	-5.87	-1.47
23.4	0.17	0	-0.17	-1.46	-0.99	26.04	-1.03	-0.3	0.72	-6.66	-2.16
23.46	0.06	0	-0.06	1.92	0.86	26.1	-0.15	0	0.15	-7.09	-3.26
23.52	0.04	0	-0.04	-1.19	-0.66	26.16	-0.32	0	0.32	10.78	5.83
23.58	0.11	0	-0.11	-0.62	-0.47	26.22	-1.5	0	1.5	13.08	8.87
23.64	0.08	0	-0.08	1.8	0.76	26.28	-1.99	0	1.99	-1.52	2.44
23.7	-0.1	0	0.1	0.85	0.57	26.34	-1.6	0	1.6	-6.39	-0.59
23.76	-0.07	0	0.07	-0.74	-0.25	26.4	-1.24	0	1.24	-1.94	1.04
23.82	-0.06	0	0.06	0.55	0.36	26.46	-1	0	1	-5.7	-1.21
23.88	-0.19	0	0.19	1.73	1.16	26.52	-0.16	0	0.16	-16.05	-7.67
23.94	-0.16	0	0.16	-2.6	-1.04	26.58	1.16	0	-1.16	-15.11	-9.32
24	0.13	0	-0.13	-4.92	-2.64	26.64	1.73	0	-1.73	1.08	-2.23
24.06	0.25	0	-0.25	-0.66	-0.72	26.7	1.13	0	-1.13	13.37	4.8
24.12	0.17	0	-0.17	0.62	0.03	26.76	0.28	0	-0.28	10.33	4.65
24.18	0.11	0	-0.11	2.17	0.9	26.82	-0.32	0	0.32	7.6	4.26
24.24	-0.1	-0.81	-0.71	-15.35	-8.8	26.88	-0.82	0	0.82	6.13	4.35
24.3	-0.88	-1.9	-1.02	8.71	2.47	26.94	-0.99	0	0.99	-0.55	1.33
24.36	-3.06	-3.48	-0.42	10.56	4.19	27	-0.77	0	0.77	-2.83	-0.15
24.42	-4.99	-4.54	0.46	6.01	3.26	27.06	-0.46	0	0.46	-2.8	-0.64
24.48	-5.74	-5.92	-0.18	-13.64	-7.59	27.12	-0.04	0	0.04	-7.99	-3.87
24.54	-6.13	-6.75	-0.61	-0.12	-1.7	27.18	0.62	0	-0.62	-5.59	-3.75
24.6	-7.07	-7.9	-0.83	-0.57	-2.37	27.24	0.76	0	-0.76	0.7	-0.88
24.66	-7.94	-8.56	-0.62	2.04	-0.81	27.3	0.82	0	-0.82	-2.47	-2.53
24.72	-8.33	-9.46	-1.13	-7.44	-6.4	27.36	1.35	0	-1.35	-8.08	-6.15
24.78	-8.8	-10.1	-1.3	2.33	-1.92	27.42	1.65	0	-1.65	3.56	-0.88
24.84	-9.77	-10.9	-1.16	4.47	-0.71	27.48	0.81	0	-0.81	15.76	6.49
24.9	-10.8	-11.4	-0.58	5.89	0.87	27.54	-0.33	0	0.33	14.06	7.47
24.96	-11.4	-12.0	-0.64	-1.23	-2.79	27.6	-0.85	0	0.85	0.82	1.76
25.02	-12.0	-12.5	-0.54	2.76	-0.73	27.66	-0.59	0	0.59	-4.92	-1.49
25.08	-12.6	-13.2	-0.58	-2.23	-3.3	27.72	-0.28	0	0.28	-1.59	-0.33
25.14	-13.1	-13.6	-0.45	-0.49	-2.28	27.78	-0.21	0	0.21	-3.45	-1.37
25.2	-13.2	-14.0	-0.81	-6.62	-5.93	27.84	0.14	0	-0.14	-6.58	-3.47
25.26	-13.4	-14.3	-0.85	2.89	-1.32	27.9	0.42	0	-0.42	-1.3	-1.32
25.32	-14.1	-14.4	-0.28	9.53	2.87	27.96	0.32	0	-0.32	3.78	1.35

28.02	-0.05	0	0.05	2.66	1.39
28.08	-0.07	0	0.07	-1.17	-0.46
28.14	-0.13	0	0.13	1.72	1.05
28.2	-0.36	0	0.36	2.12	1.62
28.26	-0.21	0	0.21	-3.87	-1.57
28.32	-0.08	0	0.08	-1.28	-0.51
28.38	-0.06	0	0.06	-0.9	-0.35
28.44	0	0	0	-0.98	-0.49
28.5	0.09	0	-0.09	-1.29	-0.78
28.56	0.03	0	-0.03	-0.42	-0.26
28.62	0.1	0	-0.1	-2.99	-1.63
28.68	0.23	0	-0.23	0.25	-0.26
28.74	0.11	0	-0.11	1.5	0.57
28.8	0.09	0	-0.09	-1.02	-0.65
28.86	-0.04	0	0.04	1.24	0.68
28.92	-0.15	0	0.15	0.08	0.28
28.98	-0.05	0	0.05	-0.81	-0.32
29.04	-0.06	0	0.06	2.42	1.29
29.1	-0.17	0	0.17	-1.29	-0.36
29.16	-0.03	0	0.03	-2.67	-1.27
29.22	0.27	1.11	0.84	4.87	3.86