

A Nonlinear Discrete-Time Sliding Mode Controller for Autonomous Navigation of an Aerial Vehicle Using Hector SLAM

Aydin Can* Joshua Price** Allahyar Montazeri*

* *Engineering Department, Lancaster University, Lancaster, LA1 4YW, UK*

** *National Nuclear Laboratory, 5th Floor, Chadwick House, Warrington Road, Birchwood Park, Warrington, WA3 6AE, UK*

Abstract: In this paper, a discrete-time sliding mode controller (DTSMC) is designed for full position and attitude control of a quadrotor UAV. The aim of this study is to design a controller suitable for practical implementation on an autonomous quadrotor for remote sensing in the hostile nuclear environments. A nested DTSMC is developed and compared against two continuous-time sliding mode control methods; classical SMC as well as a chattering-free SMC (CFSMC) studied in the previous works. The performance of the controllers are evaluated in combination with the Hector SLAM algorithm for localisation in GPS denied environments. For this purpose, MATLAB in combination with the Robotic Operating System (ROS) are used to develop the controllers. Control signals are sent from MATLAB to the Gazebo simulation environment in ROS, which simulates the quadrotor and runs the Hector SLAM algorithm.

Keywords: Sliding mode control, Quadrotor control, Nonlinear discrete-time control, Hector SLAM, Trajectory tracking.

1. INTRODUCTION

Due to developments in technology, with sensors and computing technology becoming increasingly smaller and more affordable, Unmanned Aerial Vehicles (UAVs) have gained much attention from commercial avenues, including emergency services, building inspection (Congress et al., 2018), and agricultural sectors (Barbedo, 2019). As UAVs are both affordable, and highly mobile, they provide great platforms for sensing tasks. This makes them strong candidates for operations such as sensing in hazardous and remote environments. In particular, they provide a solution to many of the challenges faced by the nuclear industry. So far they have been tested successfully through the RISER project, in which a UAV was remotely operated for the inspection of a plant on Sellafield site. The same UAV was later tested through the remote inspection of the Windscale Pile 1 chimney on the same site to demonstrate the capabilities of this new technology (Taylor et al., 2017). Areas in which the nuclear industry is attempting to develop new technologies to address existing challenges include characterisation of legacy sites, remote sensor deployment, and asset tracking to name a few (Game Changers, n.d.). By introducing these new technologies in the form of cyber-physical systems, the nuclear industry can

be exposed to the fourth industrial revolution, or Industry 4.0 (Montazeri et al., 2021; Lu et al., 2020; Blanchet and Confais, 2016).

One factor that poses a challenge for the adoption of new technologies in the nuclear sector is safety. Hazards such as radiation and contamination can make it difficult to introduce UAVs into the sector. External disturbances such as wind, payload changes (Nemati and Montazeri, 2018b,a), and sensor noise from radiation (Nemati and Montazeri, 2019) are present and it is suggested that robust control algorithms are used to counteract their effects. The coupled non-linear and under-actuated dynamics of quadrotors further adds to this requirement and designing more advanced robust methods are required for use in these safety critical scenarios.

As many of the applications of this technology are required for indoor use, the UAVs must be able to operate without the use of a Global Positioning System (GPS). For this reason, other positioning methods using visual data must be implemented. One example of a system used is Simultaneous Localisation and Mapping (SLAM), in which different visual sensors, such as LiDAR and cameras, are used to position the quadrotor in space while mapping the environment (Sadeghzadeh-Nokhodberiz et al., 2021; Mur-Artal and Tardos, 2017; Hornung et al., 2013). Most notably, Hector SLAM uses a 2D LiDAR and on-board computational technology to position a quadrotor in space (Kamarudin et al., 2014).

With the advent of cyber-physical systems and networked control technologies in design of industrial autonomous systems in general and nuclear industry in particular,

* This work has been supported by the Centre for Innovative Nuclear Decommissioning (CINDe), which is led by the National Nuclear Laboratory, in partnership with Sellafield Ltd. and a network of Universities that includes the University of Manchester, Lancaster University, the University of Liverpool and the University of Cumbria. The authors would also like to acknowledge the Engineering and Physical Sciences Research Council (EPSRC), grant number EP/R02572X/1, and National Centre for Nuclear Robotics (NCRN).

the use of sampled-data control methods have gained increasing popularity. In the field of robotic control for real-time implementation of such systems, implementing a fixed sampling time is necessary for periodic transmission of sensors' data and execution of the control signals through the actuators. For example, in the SLAM-based control systems, depending on the type of the sensors, the publishing rate of the position estimate can vary greatly. Intel Realsense depth cameras have the ability to operate between 6 Hz and 90 Hz, while various 2D LiDAR scanners used for SLAM can vary between 5 Hz and 50 Hz (Liu et al., 2018; Slamtec, n.d.; Intel, 2020). In these scenarios, when the update rate of the estimated position of a robot is slower, discretisation of the designed continuous-time SMC will not show the same performance as expected due to the long sampling period. Instead, implementing the controller in the discrete domain from the beginning should allow for improved robust stability and performance compared to the continuous-time design. A discrete-time multi-channel SMC is developed in another study for position and attitude control of a quadrotor, for a time-invariant set-point (Xiong and Zhang, 2016). Another study combines a discrete-time SMC with a disturbance observer. This control system provides excellent tracking and robustness in the presence of disturbances and uncertainties (Han et al., 2019). One alternative approach to discrete time methods is an event-triggered approach (Sarkar et al., 2017; Tian et al., 2021; Nokhodberiz et al., 2019). A new strategy for discrete implementation of such control systems, specially due to resource constraint nature of the robotic platforms, is known as the event-triggered control.

This study builds on the previous work by the Engineering Department at Lancaster University on the development of robust control algorithms for quadrotor UAVs in hazardous environments (Can et al., 2020). In this previous research, a nested chattering-free sliding mode controller (CFSMC) was applied for position and attitude control of a quadrotor UAV for use in hazardous environments with parametric uncertainties. In the present paper, a nested discrete-time sliding mode controller is designed and developed for a quadrotor UAV for full trajectory tracking control in indoor hazardous environments. The results are compared with continuous-time control methods including classical SMC as well as chattering-free SMC to highlight practical issues of implementing these controllers along with the SLAM algorithm for low sampling rates in a closed-loop system. The results confirm the improved robustness of the proposed discrete-time method for both inner and outer-loop control with Hector SLAM in loop when compared with the other two methods.

The remainder of this paper is organised as follows: the quadrotor model is derived in section 2, and the derivation of the proposed discrete-time sliding mode controller is presented in section 3; section 4 will discuss some details on the Hector SLAM; and the proposed method compared with the previous ones is studied in section 5; Finally, section 6 will conclude the paper and offer avenues for the future research.

2. QUADROTOR MODEL

The design of a control system for the quadrotor first requires the derivation of the mathematical model of

the quadrotor. The continuous-time quadrotor model is given in equation (1) (Can et al., 2020). Variables $[x, y, z]$ represent the position of the quadrotor, while $[\varphi, \theta, \psi]$ represent the roll, pitch and yaw angles of the quadrotor. The mass of the quadrotor is denoted by m . The moments of inertia are represented by $[I_{xx}, I_{yy}, I_{zz}]$, while control inputs are represented by $[u_1, u_2, u_3, u_4]$.

$$\begin{cases} \ddot{x} = (-c\varphi c\psi s\theta - s\varphi s\psi) \frac{u_1}{m} \\ \ddot{y} = (-c\varphi s\theta s\psi - c\psi s\varphi) \frac{u_1}{m} \\ \ddot{z} = g - c\varphi c\theta \frac{u_1}{m} \\ \ddot{\varphi} = \frac{(I_{yy} - I_{zz}) \times \dot{\theta} \times \dot{\psi}}{I_{xx}} + \frac{u_2}{I_{xx}} \\ \ddot{\theta} = \frac{(I_{zz} - I_{xx}) \times \dot{\varphi} \times \dot{\psi}}{I_{yy}} + \frac{u_3}{I_{yy}} \\ \ddot{\psi} = \frac{(I_{xx} - I_{yy}) \times \dot{\varphi} \times \dot{\theta}}{I_{zz}} + \frac{u_4}{I_{zz}} \end{cases} \quad (1)$$

where $c\varphi = \cos(\varphi)$, $s\varphi = \sin(\varphi)$ and $t\varphi = \tan(\varphi)$.

In order to design a discrete-time control system, the continuous model of the quadrotor must be converted into the discrete domain. Though a number of methods for discretisation exist, this paper will use the forward-Euler method shown in equation (2).

$$\dot{x}_k = \frac{x_{k+1} - x_k}{T} \quad (2)$$

Variable T denotes the sample time of the discrete system. The discrete-time quadrotor model is therefore calculated using equation (2) to give the set of system equations in (3) and (4).

$$\begin{cases} x_{k+1} = x_k + T\dot{x}_k \\ \dot{x}_{k+1} = \dot{x}_k + T(-c\varphi_k c\psi_k s\theta_k - s\varphi_k s\psi_k) \frac{u_{1,k}}{m} \\ y_{k+1} = y_k + T\dot{y}_k \\ \dot{y}_{k+1} = \dot{y}_k + T(-c\varphi_k s\theta_k s\psi_k - c\psi_k s\varphi_k) \frac{u_{1,k}}{m} \\ z_{k+1} = z_k + T\dot{z}_k \\ \dot{z}_{k+1} = \dot{z}_k + T(g - (c\varphi_k c\theta_k) \frac{u_{1,k}}{m}) \end{cases} \quad (3)$$

$$\begin{cases} \varphi_{k+1} = \varphi_k + T\dot{\varphi}_k \\ \dot{\varphi}_{k+1} = \dot{\varphi}_k + T\left(\frac{(I_{yy} - I_{zz})\dot{\theta}_k\dot{\psi}_k + u_{2,k}}{I_{xx}}\right) \\ \varphi_{k+1} = \varphi_k + T\dot{\varphi}_k \\ \dot{\theta}_{k+1} = \dot{\theta}_k + T\left(\frac{(I_{yy} - I_{zz})\dot{\varphi}_k\dot{\psi}_k + u_{3,k}}{I_{xx}}\right) \\ \psi_{k+1} = \psi_k + T\dot{\psi}_k \\ \dot{\psi}_{k+1} = \dot{\psi}_k + T\left(\frac{(I_{yy} - I_{zz})\dot{\varphi}_k\dot{\theta}_k + u_{4,k}}{I_{xx}}\right) \end{cases} \quad (4)$$

where x_k represents the variable x at time step k and x_{k+1} represents the variable x one time step in the future. Using this set of discrete-time equations, a discrete-time sliding mode controller can be developed.

3. DISCRETE-TIME SLIDING MODE CONTROL

The task of control system design for quadrotor UAVs is challenging due to the non-linear and under-actuated quadrotor dynamics. In this section, a discrete-time sliding mode controller (DTSMC) is designed.

Control systems used in modern robotics rely on sensor data for feedback of the state variables. These sensors are inherently discrete-time systems with sampling periods. The natural progression for these control systems is, therefore, for them to be developed and implemented entirely in the discrete domain. The control system developed in this section uses a nested, multi-channel structure to allow for full position and attitude control of the quadrotor. An inner-loop attitude control subsystem controls the roll, pitch and yaw of the quadrotor. An outer-loop control system controls the position of the quadrotor by generating desired angles for the attitude controller based on the position error. A separate subsystem controls the altitude of the quadrotor.

3.1 Attitude and Altitude Subsystems

The attitude subsystem controls the roll, pitch and yaw of the quadrotor, while the altitude control system controls the height of the quadrotor. To derive these, first, the sliding surface for each channel σ_s is defined at time step k in the discrete domain.

$$\sigma_{s,k} = a_s(s_k^d - s_k) + (\dot{s}_k^d - \dot{s}_k) \quad (5)$$

where s_k and \dot{s}_k represent any variable and the rate of change of that variable at time step k . s_k^d and \dot{s}_k^d represent the desired value and rate of change of variables s and \dot{s} at time step k . a_s is a tuning parameter. The sliding surface is then defined at time step $k+1$.

$$\sigma_{s,k+1} = a_s(s_{k+1}^d - s_{k+1}) - (\dot{s}_{k+1}^d - \dot{s}_{k+1}) \quad (6)$$

Next, a discrete-time reaching law proposed by Gao et al. (1995) is implemented to force the system to slide along sliding surface σ .

$$\begin{aligned} \sigma_{s,k+1} - \sigma_{s,k} &= -\eta_s T \sigma_{s,k} - \epsilon_s T \text{sgn}(\sigma_{s,k}) \\ \eta_s &> 0, \sigma_s > 0, 1 - \eta_s T > 0, \end{aligned} \quad (7)$$

where η_s and ϵ_s are tuning parameters. sgn is the signum function. By substituting equations (5) and (6) into (7), while using the discrete model in (3) and (4), the control laws for inputs u_1 to u_4 at time step k can be derived.

$$\begin{aligned} u_{1,k} &= \left(\frac{m}{c\varphi_k c\theta_k T} \right) (Tg + a_z z_{k+1}^d - a_z T \dot{z}_k + \dot{z}_{k+1}^d \\ &\quad - a_z \dot{z}_k^d - \dot{z}_k^d + (\eta_z \sigma_{z,k} T) + (\epsilon_z T \text{sgn}(\sigma_{z,k}))) \end{aligned} \quad (8)$$

$$\begin{aligned} u_{2,k} &= \left(\frac{I_{xx}}{T} \right) \left(-T \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) \dot{\theta}_k \dot{\psi}_k + a_\varphi \varphi_{k+1}^d \right. \\ &\quad \left. - a_\varphi T \dot{\varphi}_k - \dot{\varphi}_{k+1}^d - a_\varphi \varphi_k^d - \dot{\varphi}_k^d + (\eta_\varphi \sigma_{\varphi,k} T) \right. \\ &\quad \left. + (\eta_\varphi \sigma_{\varphi,k} T) + (\epsilon_\varphi T \text{sgn}(\sigma_{\varphi,k})) \right) \end{aligned} \quad (9)$$

$$\begin{aligned} u_{3,k} &= \left(\frac{I_{yy}}{T} \right) \left(-T \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) \dot{\varphi}_k \dot{\psi}_k + a_\theta \theta_{k+1}^d \right. \\ &\quad \left. - a_\theta T \dot{\theta}_k - \dot{\theta}_{k+1}^d - a_\theta \theta_k^d - \dot{\theta}_k^d \right. \\ &\quad \left. + (\eta_\theta \sigma_{\theta,k} T) + (\epsilon_\theta T \text{sgn}(\sigma_{\theta,k})) \right) \end{aligned} \quad (10)$$

$$\begin{aligned} u_{4,k} &= \left(\frac{I_{zz}}{T} \right) \left(-T \left(\frac{I_{xx} - I_{yy}}{I_{zz}} \right) \dot{\varphi}_k \dot{\theta}_k + a_\psi \psi_{k+1}^d \right. \\ &\quad \left. - a_\psi T \dot{\psi}_k - \dot{\psi}_{k+1}^d - a_\psi \psi_k^d - \dot{\psi}_k^d \right. \\ &\quad \left. + (\eta_\psi \sigma_{\psi,k} T) + (\epsilon_\psi T \text{sgn}(\sigma_{\psi,k})) \right) \end{aligned} \quad (11)$$

3.2 Position Control Subsystem

The inner-loop subsystem allows control over the attitude and the altitude of the quadrotor. In order to control the position of the quadrotor to allow control over all six degrees of freedom, an outer-loop controller is developed that generates desired angles based on the position error.

In the discrete-time position quadrotor model in (3), two new controller gains, u_x and u_y , are substituted in.

$$u_{x,k} = s\varphi_k s\psi_k + c\varphi_k s\theta_k c\psi_k \quad (12)$$

$$u_{y,k} = -s\varphi_k c\psi_k + c\varphi_k s\theta_k s\psi_k \quad (13)$$

This gives two new equations for the discrete time quadrotor model.

$$\begin{cases} x_{k+1} = x_k + T\dot{x}_k \\ \dot{x}_{k+1} = \dot{x}_k + u_{x,k} \frac{u_{1,k} T}{m} \\ y_{k+1} = y_k + T\dot{y}_k \\ \dot{y}_{k+1} = \dot{y}_k + u_{y,k} \frac{u_{1,k} T}{m} \end{cases} \quad (14)$$

By substituting equations (5) and (6) into (7), while using the new model shown in (14), two new control laws for u_x and u_y can be derived.

$$\begin{aligned} u_{x,k} &= \left(\frac{m}{T} \right) (a_x x_{k+1}^d - a_x T \dot{x}_k + \dot{x}_{k+1}^d - a_x x_k^d \\ &\quad - \dot{x}_k^d + (\eta_x \sigma_{x,k} T) + (\epsilon_x T \text{sgn}(\sigma_{x,k}))) \end{aligned} \quad (15)$$

$$\begin{aligned} u_{y,k} &= \left(\frac{m}{T} \right) (a_y y_{k+1}^d - a_y T \dot{y}_k + \dot{y}_{k+1}^d - a_y y_k^d \\ &\quad - \dot{y}_k^d + (\eta_y \sigma_{y,k} T) + (\epsilon_y T \text{sgn}(\sigma_{y,k}))) \end{aligned} \quad (16)$$

Values $u_{x,k}$ and $u_{y,k}$ can then be converted into φ_k^d and θ_k^d using equations (17) and (18). This can then be fed into the attitude control subsystem to allow full control over the quadrotor's position and attitude in 3D space.

$$\varphi_k^d = \arcsin(u_{x,k} \sin(\psi_k) - u_{y,k} \cos(\psi_k)) \quad (17)$$

$$\theta_k^d = \arcsin\left(\frac{u_{x,k} \cos(\psi_k) + u_{y,k} \sin(\psi_k)}{\cos(\varphi_k)}\right) \quad (18)$$

4. HECTOR SLAM

Hector SLAM is a 2D SLAM method that relies on laser scan data from a LiDAR sensor mounted to a robot (Kamarudin et al., 2014). Hector SLAM is often used in

aerial robotic applications, as it does not rely on odometry data from the turning of wheels for localisation. Instead, it relies on LiDAR data and scan matching techniques to create a map and localise itself within that map. Other benefits of Hector SLAM are its 360° capabilities due to its implementation of LiDAR data over visual data, its increased computational efficiency, and its increased range when compared to 3D visual methods.

For positioning and localisation of the quadrotor, a 2D LiDAR was mounted to the quadrotor frame in ROS and Hector SLAM was used. Rather than using ground truth values for control, SLAM provides estimated values for the quadrotor's position in space (Kamarudin et al., 2014). A scan matching technique is used to match LiDAR points to a previously generated map. This is estimated using (19), where $\xi = (\hat{x}, \hat{y}, \hat{\psi})^T$, and $\Delta\xi$ is calculated once per revolution of the 2D LiDAR when the map is updated.

$$\xi_k = \xi_{k-1} + \Delta\xi \quad (19)$$

where $\Delta\xi$ is calculated through optimizing the error between the current LiDAR points and the generated map. The variables \hat{x} , \hat{y} , and $\hat{\psi}$ represent the estimated values of x , y , and ψ respectively. For the purpose of this research it is assumed that $(x, y, \psi)^T = (\hat{x}, \hat{y}, \hat{\psi})^T$. Both Simulink and MATLAB were used for implementation of each control system, while the Robot Operating System (ROS) was used to implement the Hector SLAM algorithm. Simulink was also used to read the estimated states of the quadrotor, as well as communicate control signals to ROS. A flow diagram demonstrating the communication between ROS and Simulink is shown in Figure 1.

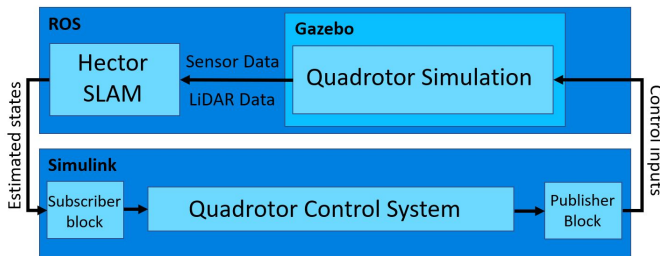


Fig. 1. Flow chart showing communication between ROS and Simulink.

5. SIMULATION RESULTS

To ensure testing of the control systems is as accurate-to-life as possible, ROS was used to simulate the quadrotor system. A quadrotor frame, equipped with various sensors was simulated inside of an indoor world environment. Figure 2 shows the simulation of the quadrotor, as well as the visualised LiDAR data and the map in the ROS software RViz. In order to evaluate the performance of the discrete time controller, it will be compared against classical SMC as well as CFSMC in three scenarios. Firstly it will be tested with a smaller sampling period of $T = 0.01$ seconds, then a larger sampling time of $T = 0.05$ seconds, and finally a sampling time of $T = 0.1$ seconds. In each scenario, the quadrotor was tasked to track a helical trajectory, with $x^d = 0.5 \sin(t)$, $y^d = 0.5 \cos(t)$, and $z^d = 0.3 + \frac{t}{25}$, where t represents the current simulation time.

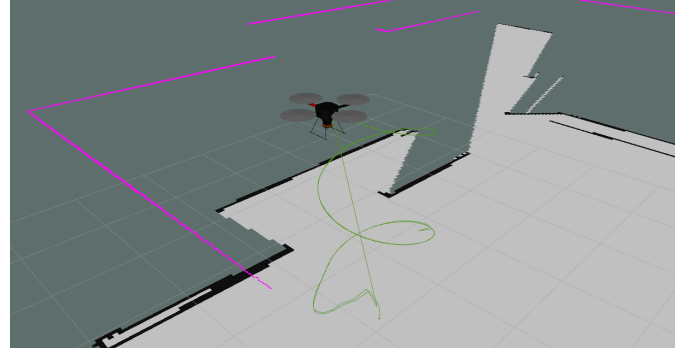


Fig. 2. RViz software displaying quadrotor trajectory, laser scan data, and the developed map.

5.1 Results

The sampling rate of the sensors within ROS, the control system in Simulink, and the rate at which commands were published from Simulink to ROS were each set to the desired sampling rate in order to test each control system. The integral of absolute error (IAE) was measured in order to evaluate and compare the performance of each controller. Figures 3 and 5 demonstrate the performance of the quadrotor following the trajectory, with each control system implemented at a sample time of $T = 0.01$ and $T = 0.05$ seconds respectively. Table 1 displays and compares the integral of absolute error for each control system at each sample time.

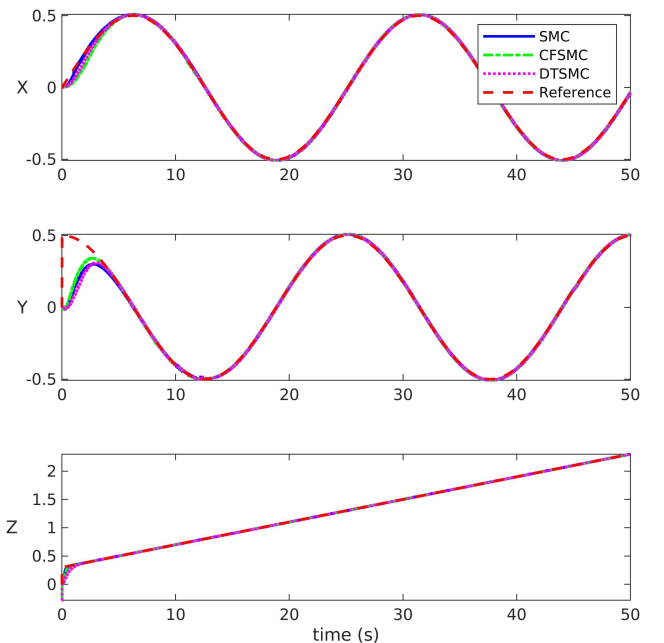


Fig. 3. Position response of all control systems with a sampling time of 0.01s.

Due to the robust nature of sliding mode control, all three control systems display strong performance with a smaller sampling time of 0.01 seconds. DTSMC offers comparable performance to classical SMC for control of the X and Y position of the quadrotor. DTSMC also displays slightly worse performance in the control of the Z position of the quadrotor. CFSMC displays the best control over the Y

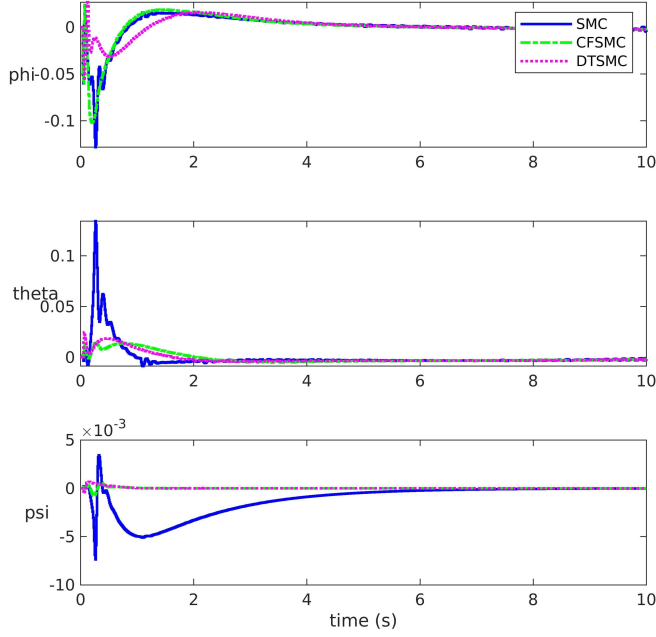


Fig. 4. Attitude response of all control systems with a sampling time of 0.01s.

position of the quadrotor, while performing slightly worse than SMC and DTSMC in the control over the X position.

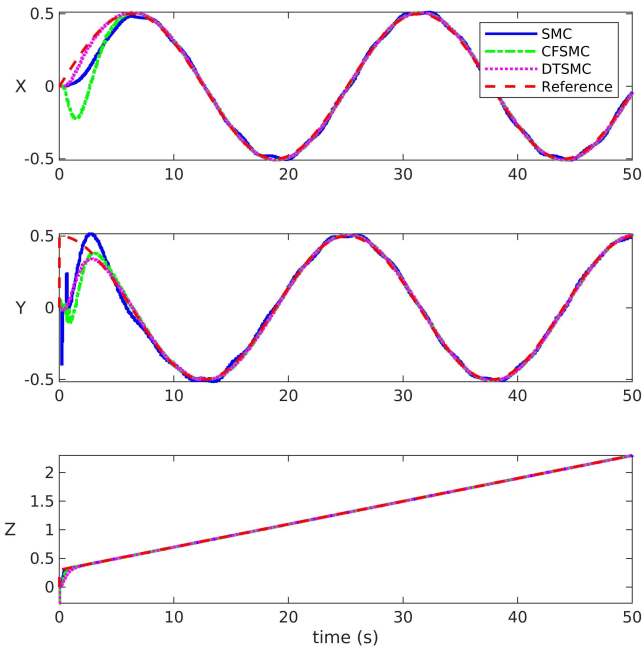


Fig. 5. Position response of all control systems with a sampling time of 0.05s.

When the sampling time is increased to $T = 0.05$ seconds, both classical SMC and CFMCM display a deterioration in controller performance. The mean increase in IAE between $T = 0.01$ and $T = 0.05$ for position error of the quadrotor using SMC and CFMCM is 75.6% and 86.3% respectively. The mean increase in position error of the quadrotor using DTSMC is 0.33%. These results suggest that continuous sliding mode control methods such as classical SMC and CFMCM are not robust against changes in sample time

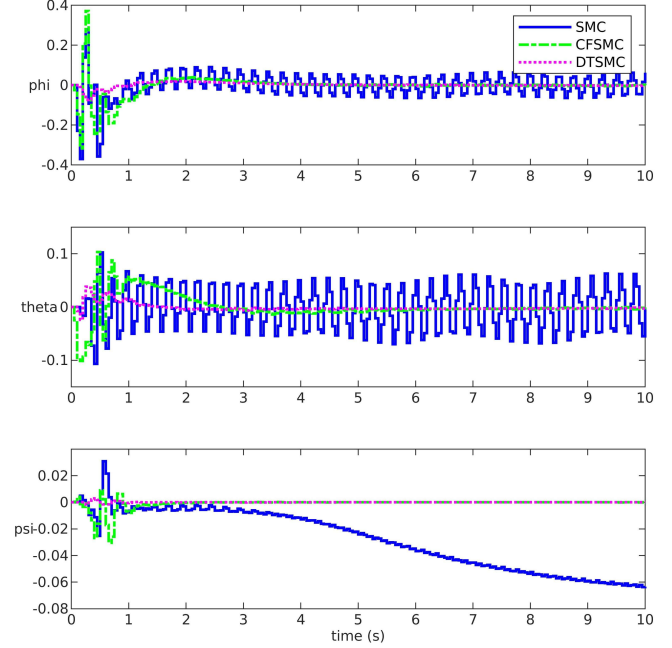


Fig. 6. Attitude response of all control systems with a sampling time of 0.05s.

Table 1. IAE for Quadrotor Positions for Each Sample Time T .

Integral of Absolute Error	Control Systems		
	SMC	CFMCM	DTSMC
T = 0.01 seconds			
X Position	0.3147	0.4382	0.3457
Y Position	1.1046	0.9495	1.1204
Z Position	0.0759	0.0935	0.1630
Total	1.4952	1.4812	1.6291
T = 0.05 seconds			
X Position	1.0590	1.3213	0.3944
Y Position	1.4730	1.3058	1.0573
Z Position	0.0942	0.1328	0.1827
Total	2.6262	2.7599	1.6344
T = 0.1 seconds			
X Position	Fail	Fail	0.8995
Y Position	Fail	Fail	1.1565
Z Position	Fail	Fail	0.0931
Total	-	-	2.1491

on discrete systems. Meanwhile, the discrete-time sliding mode controller developed in this study demonstrates robustness against an increase in sample time, and shows far better performance than both continuous methods at a larger sample time of 0.05 seconds. Both SMC and CFMCM displayed overshoot when the sampling time was increased, while DTSMC did not. Furthermore, increasing the sample time to $T = 0.1$ seconds caused both continuous control methods to fail, while DTSMC remained stable.

In the case of attitude response, SMC and CFMCM both display oscillatory behaviour when the sampling time is increased, while DTSMC has stable transient response. SMC also displays steady state error for the yaw of the quadrotor.

6. CONCLUSIONS AND FUTURE WORK

In this study, a robust control system was developed in the discrete domain for full trajectory tracking control of a quadrotor in 3D space. Final implementation of this controller will take place on an autonomous quadrotor for implementation in hazardous indoor environments in the nuclear industry. All three controllers displayed robust and accurate performance for the task of trajectory tracking in indoor environments when sampling time was small. However, discrete-time sliding mode control provides a far more suitable option for scenarios where the sampling time of a system is larger, such as when autonomous SLAM is used for on-board localisation.

REFERENCES

- Barbedo, J. (2019). A review on the use of unmanned aerial vehicles and imaging sensors for monitoring and assessing plant stresses. *Drones*, 3(2), 40.
- Blanchet, M. and Confais, E. (2016). 4.0 industry: a new industrial challenge and a new economic model. *Revue Generale Nucleaire*, 12–16. URL <http://inis.iaea.org/Search/search.aspx?orig{ }q=RN:48015032>.
- Can, A., Efstathiades, H., and Montazeri, A. (2020). Design of a chattering-free sliding mode control system for robust position control of a quadrotor. In *2020 International Conference Nonlinearity, Information and Robotics, NIR 2020*. Institute of Electrical and Electronics Engineers Inc.
- Congress, S.S., Puppala, A.J., and Lundberg, C.L. (2018). Total system error analysis of UAV-CRP technology for monitoring transportation infrastructure assets. *Engineering Geology*, 247, 104–116.
- Game Changers (n.d.). Challenges. URL <https://www.gamechangers.technology/challenges/>.
- Gao, W., Wang, Y., and Homaifa, A. (1995). Discrete-time variable structure control systems. *IEEE Transactions on Industrial Electronics*, 42(2), 117–122. doi:10.1109/41.370376.
- Han, J.S., Kim, T.I., Oh, T.H., Kim, Y.S., Lee, J.H., Kim, S.O., Lee, S.S., Lee, S.H., and Cho, D.I.D. (2019). Error-dynamics-based performance shaping methodology for discrete-time sliding mode control with disturbance observer. volume 52, 460–464. Elsevier B.V.
- Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3), 189–206. doi:10.1007/s10514-012-9321-0.
- Intel (2020). External synchronization of Intel RealSense™ depth cameras. URL <https://dev.intelrealsense.com/docs/external-synchronization-of-intel-realsense-depth-cameras>.
- Kamarudin, K., Mamduh, S.M., Md Shakaff, A.Y., and Zakaria, A. (2014). Performance analysis of the Microsoft Kinect sensor for 2D simultaneous localization and mapping (SLAM) techniques. *Sensors (Switzerland)*, 14(12), 23365–23387.
- Liu, H., Gao, B., Shen, Y., Hussain, F., Addis, D., and Kai Pan, C. (2018). Comparison of Sick and Hokuyo UTM-30LX laser sensors in canopy detection for variable-rate sprayer. *Information Processing in Agriculture*, 5(4), 504–515.
- Lu, C., Lyu, J., Zhang, L., Gong, A., Fan, Y., Yan, J., and Li, X. (2020). Nuclear power plants with artificial intelligence in industry 4.0 era: Top-level design and current applications—a systemic review. *IEEE Access*, 8, 194315–194332. doi:10.1109/ACCESS.2020.3032529.
- Montazeri, A., Can, A., and Imran, I.H. (2021). Unmanned aerial systems: autonomy, cognition, and control. In A. Koubaa and A.T. Azar (eds.), *Unmanned Aerial Systems*, Advances in Nonlinear Dynamics and Chaos (ANDC), 47–80. Academic Press.
- Mur-Artal, R. and Tardos, J.D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5), 1255–1262. doi:10.1109/TRO.2017.2705103.
- Nemati, H. and Montazeri, A. (2018a). Analysis and design of a multi-channel time-varying sliding mode controller and its application in unmanned aerial vehicles. *IFAC-PapersOnLine*, 51(22), 244–249.
- Nemati, H. and Montazeri, A. (2018b). Design and development of a novel controller for robust attitude stabilisation of an unmanned air vehicle for nuclear environments. In *2018 UKACC 12th International Conference on Control, CONTROL 2018*, 373–378. Institute of Electrical and Electronics Engineers Inc.
- Nemati, H. and Montazeri, A. (2019). Output feedback sliding mode control of quadcopter using imu navigation. In *2019 IEEE International Conference on Mechatronics (ICM)*, volume 1, 634–639. IEEE.
- Nokhodberiz, N.S., Nemati, H., and Montazeri, A. (2019). Event-triggered based state estimation for autonomous operation of an aerial robotic vehicle. *IFAC-PapersOnLine*, 52(13), 2348–2353.
- Sadeghzadeh-Nokhodberiz, N., Can, A., Stolkin, R., and Montazeri, A. (2021). Dynamics-based modified fast simultaneous localization and mapping for unmanned aerial vehicles with joint inertial sensor bias and drift estimation. *IEEE Access*, 9, 120247–120260.
- Sarkar, M.K., Arkdev, and Singh, S.S.K. (2017). Sliding mode control: A higher order and event triggered based approach for nonlinear uncertain systems. In *2017 8th Industrial Automation and Electromechanical Engineering Conference, IEMECON 2017*, 208–211. Institute of Electrical and Electronics Engineers Inc.
- Slamtec (n.d.). RPLIDAR-A1 360° Laser Range Scanner. URL <https://www.slamtec.com/en/Lidar/A1>.
- Taylor, A., Hope, C., Smart, N., Morris, D., Warren, K., and Banford, A. (2017). A programme of technology demonstrations for decommissioning at Sellafield. *Nuclear Future*, 12(2), 30–34.
- Tian, B., Cui, J., Lu, H., Liu, L., and Zong, Q. (2021). Attitude control of UAVs based on event-triggered supertwisting algorithm. *IEEE Transactions on Industrial Informatics*, 17(2), 1029–1038.
- Xiong, J.J. and Zhang, G. (2016). Discrete-time sliding mode control for a quadrotor UAV. *Optik*, 127(8), 3718–3722.