UAV Formation Flight and Collision Warning with Centralized Control of Ground Control Station

Wenxuan Zheng
School of Automation Science and Electriacl Engineering
Science and Technology on Aircraft Control Laboratory
Beihang University
Beijing, China
zwx0130@buaa.edu.cn

Honglun Wang
School of Automation Science and Electriacl Engineering
Science and Technology on Aircraft Control Laboratory
Beihang University
Beijing, China
hl_wang_2002@126.com

Hongxia Ji
School of Automation Science and Electriacl Engineering
Beihang University
Beijing, China
jihongxia76@163.com

Jianfa Wu
School of Automation Science and Electriacl Engineering
Beihang University
Beijing, China
jianfa_wu@163.com

***Abstract*—Unmanned Aerial Vehicles (UAVs) are widely applied in many fields, and the application of UAV formation flights is also developing rapidly. But currently many widely used types of UAVs are difficult to work with distributed formation control theory. Therefore, this paper presents a centralized formation algorithm with a three-degree-of-freedom UAV model, and also builds up two collision detection models. After tested and verified in MATLAB/Simulink, the models and control law are exported to C++ code and embedded in Qt-based simulation platform and Ground Control System (GCS) and pass the test under the condition of manually cutting off the communication. The simulation platform provides UAV state data, and the GCS serves as Human-Machine-Interface (HMI) to show the state data and perform control law to make formation flight. With centralized control of the GCS, any UAVs can do formation flight without modification.**

Keywords—Formation flight, Consensus, Collision detection, Ground control station

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are widely applied in many fields. In the civil field, the applications include monitoring soil erosion [1], 3D mapping [2], forest and agriculture [3]. Since the United States first substantial use of UAVs during the Vietnam War, UAVs play an increasingly important role in the war [4]. From Gulf War to Libyan War, UAVs undertook the missions including radar deception [5], target tracking [6], ground-to-air missile jamming [7], air-to-ground attack [8], etc. But when a single UAV performs a complex mission, it shows significant limitations. When meeting mechanical failure, lack of fuel or some other fatal error, the mission will be affected or even fail and lead to losing in a war. Also in the battlefield, a single UAV is limited by its small operational scope and weak destruction ability, which greatly reduces the battle plan execution efficiency. Thus making several UAVs cooperate on a mission has become a hot topic in recent years [9].

Until now, various control theories have been proposed for UAV formation flight to solve different key point. Ref.[10] addressed the close formation flight control problem considering aerodynamic coupling effects and accomplished the control system with a PID controller. The distributed formation control algorithm was designed with constraint forces in [11]. Model-based Interfered Fluid Dynamical Systems (MIFDS) combined with a virtual leader is aiming at solving formation obstacle avoidance problem in three-dimensional complex obstacle environment [12]. Ref.[13] developed emergent consensus algorithms. Considering the mobility limitations of UAV in flight, Ref.[14] designed a control law with a nonlinear model. Time-varying formation control problem was solved with consensus-based approaches in [15].

However, current research focuses on distributed formation control methods [11-15], which require onboard computers to have strong performance and be installed with formation control programs. But there are many widely used types of UAVs like Global Hawk or DJI's drones that are originally designed for individual combat and are difficult to retrofit. Therefore, it is necessary to develop a centralized control algorithm to make full use of the old types. An efficient solution in this paper where no modifications need to be made to any of the UAVs is to hand over the task to a Ground Control Station (GCS).

Usually, one ground station can only control one or few UAVs, but by gathering UAVs' status data, a GCS can take charge of a whole formation. However, in practical applications, formation flight still faces many key technical challenges, such as communication networks [16], aero-dynamic coupling effects [17], collision avoidance [18] , etc. Therefore, there is an urgent need for developing a system to simulate what problem that UAVs may be encountered in a real environment. Based on the requirements of centralized UAV formation flight control, this paper carries out research on formation algorithm, collision detection and avoidance in case of data link interruption and design of GCS.

UAV model and formation control algorithm are the basis of UAV formation flight. To simplify the problem, it's reasonable to assume that the UAV is equipped with autopilots [19], which make it possible to represent the UAV with a simple first-order three-degree-of-freedom (DOF) system. The formation control algorithm is closely related to network topology structure, and here with a centralized structure, a virtual-leader-based consensus formation control algorithm is adopted. The GCS will take responsibility for doing the formation control algorithm and also monitoring the status of UAVs for collision avoidance. The GCS plays a significant role as Human-Machine-Interface (HMI), whose end goals is providing sufficient information about UAVs with user convenient Graphical User Interface (GUI) and reduce pilot workload.

The rest of this paper is organized as follows. Section II gives the formulation of three DOF model of the UAV and set up the collision warning model. Section III shows the virtual leader based consensus algorithm and tests the algorithm in Simulink. A GCS and a UAV digital simulation platform are accomplished in Section IV. Finally, in Section V, conclusion and future work are presented.

II. MODELS

A. UAV Model

Firstly, define the coordinate system. Take the airport as the coordinate origin, let the y-axis point to the north, let the x-axis perpendicular to the y-axis in the horizontal plane and point to the east, and let z-axis vertically upwards, thus making the coordinate system XOY an inertial system.

Commonly, a control system for a UAV is consist of two loops where the inner loop controls the attitude dynamics and the outer loop controls the position. In this paper, the outer loop will be realized as formation law, and the inner loop is simplified as a three-degree-of-freedom system which can be considered a first-order autopilot:

$$\begin{cases} v_{ip} = v_i \cos(\mu_i) \\ \dot{v}_i = k_v (v_i - v_{ic}) \\ \dot{\varphi}_i = k_\varphi (\varphi_i - \varphi_{ic}) \\ \dot{\mu}_i = k_\mu (\mu_i - \mu_{ic}) \end{cases} \tag{1}$$

Where the subscript i means the ith UAV; v_i is total ground speed; μ_i is track tilt angle which means the angle between the velocity vector and the xoy plane, and it makes v_{ip} the velocity projection in the xoy plane; φ_i is track deflection angle which is the angle between v_{ip} and x axis and counter-clockwise is the positive direction. Therefore, (v_i, μ_i, φ_i) is the three DOF, and $(v_{ic}, \mu_{ic}, \varphi_{ic})$ is the input control signal to each DOF, and (k_v, k_μ, k_φ) is the factor of autopilot to each DOF. It is easy to use the projection relationship to get the formula of $(\dot{x}_i, \dot{y}_i, \dot{z}_i)$ and calculate position using integrator as in (2).

$$\begin{cases} \dot{x}_i = v_{ip} \cos(\varphi_i) \\ \dot{y}_i = v_{ip} \sin(\varphi_i) \\ \dot{z}_i = v_i \sin(\mu_i) \end{cases} \tag{2}$$

B. Collision Detection Models

The Traffic Alert and Collision Avoidance System (TCAS) [20] and the Automatic Dependent Surveillance Broadcast (ADS-B) [21], which are widely used in the civil aviation field, use the relevant flight interval regulations to define collision avoidance zones, and based on the relative position between the aircraft monitored in real-time, they provide the pilot with warning information, and finally the pilots manipulate the aircraft to avoid conflicts. Although this kind of collision avoidance system cannot be directly used for the collision avoidance decision of the drone, it provides a good research basis for the design of the UAV collision detection and avoidance system.

Considering the difference in maneuverability between the horizontal and vertical directions of the UAV, the space detection model is designed as a cylinder. Centering on the center of mass of the UAV, establish a cylinder detection zone with height H and radius R as shown in Fig. 1. Where $R = \sqrt{(x_0 - x)^2 + (y_0 - y)^2}$ and $H = |z_0 - z|$, which represent horizontal and vertical distance.

A fast-approaching UAV makes it difficult for the other UAVs to ensure sufficient reaction time, thus it's essential to build a time detection model to estimate the remaining time before collision. The remaining time Tau satisfies the equation (3):

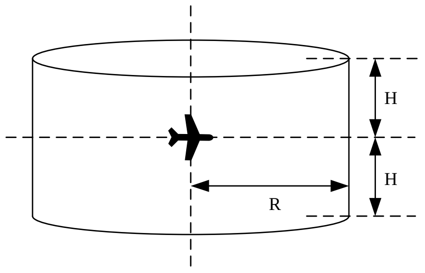


Fig. 1. Cylinder collision detection zone.

$$Tau = \frac{d - D_{Mod}^2 / d}{\max\left(|\dot{d}|, V_{\min}\right)} \tag{3}$$

Where $d = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2}$ is the relative distance between two UAVs; D_{Mod} is a distance correction factor reducing Tau when both d and $\dot{d}$ are small to find collisions in advance. The value of D_{Mod} may vary from 500m to 2000m for different types of UAVs and missions. $V_{\min}$ is a small velocity to avoid singular values.

III. FORMATION CONTROL ALGORITHM

At present, there are many mature precedents for UAV formation control algorithms. In this paper, a multi-agent system-based consensus algorithm is adopted.

A. Consensus Protocol

Information flow graph in consensus formation problem is the basis to build the algorithm. This paper let all the UAVs communicate directly with GCS, thus the formation controller does not have to estimate the status of UAVs and the graph is simplified into only n path where n equals the number of UAVs. In order to achieve both the desired posture and position, here defines consensus variable including the 3DOF and 3 coordinates as $\mathbf{s}_i = (v_i, \mu_i, \varphi_i, x_i, y_i, z_i)^T$, and give the first-order consensus protocol [22] as bellow:

$$\begin{cases} \mathbf{e}_i = \mathbf{k}_i (\mathbf{s}_i - \mathbf{s}_i^d) \\ \mathbf{c}_i = controller(\mathbf{e}_i) \end{cases} \tag{4}$$

Where $\mathbf{s}_i = (v_i, \varphi_i, \mu_i, x_i, y_i, z_i)^T$ is the UAV state and $\mathbf{s}_i^d$ means the desired state. $\mathbf{c}_i = (v_{ic}, \varphi_{ic}, \mu_{ic})^T$ is the input control signal and $\mathbf{e}_i = \left(e_x, e_y, e_z\right)^T$ is state error in 3 DOF. On account of the different dimension of $\mathbf{c}_i$ and $\mathbf{s}_i$, $\mathbf{k}_i$ is a 3×6 factor matrix.

Considering the relationships between each variable, here defines the $\mathbf{k}_i$ in (4) as bellow:

$$\mathbf{k}_i = \begin{bmatrix} k_{ev} & & & k_{ex} & & \\ & k_{e\varphi} & & & k_{ey} & \\ & & k_{e\mu} & & & k_{ez} \end{bmatrix} \tag{5}$$

In (5), the deviation on x-axis is mainly determined by the x axis interval and velocity error, and e_y is determined by y axis and track deflection angle, and e_z is consisted of z axis and track tilt angle. It has to transform the coordinate to ensure the correct relationships. Rotate the ground coordinate system XOY in Section.II.*A* and make its y axis parallel to the heading of the leader UAV, then the leader coordinate system $X^L OY^L$ is shown in Fig. 2, where L and W is leader and wingman, and the superscript L means the coordinate transformed into $X^L OY^L$. The coordinate conversion formula is shown in (6).

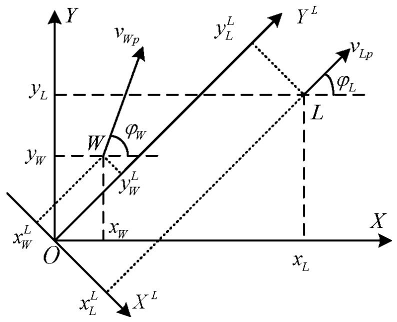


Fig. 2. Coordinate transformation diagram

$$\begin{cases} x^L = x\sin\left(\varphi_L\right) - y\cos\left(\varphi_L\right) \\ y^L = x\cos\left(\varphi_L\right) + y\sin\left(\varphi_L\right) \end{cases} \tag{6}$$

Here record the state variable after coordinate transformation as $\mathbf{s}_i^L$ and replace the $\mathbf{s}_i$ in (4) as bellow:

$$\begin{cases} \mathbf{e}_i = \mathbf{k}_i\left(\mathbf{s}_i^L - \mathbf{s}_i^d\right) \\ \mathbf{c}_i = controller\left(\mathbf{e}_i\right) \end{cases} \tag{7}$$

B. Formation Structure Design

Common formations include triangle, column, etc. Here, with a virtual leader, the formations have to be modified. The virtual leader is set a fixed distance in front of the entire formation which is typically 1000m. For example, a triangle formation of three UAVs is modified as below.

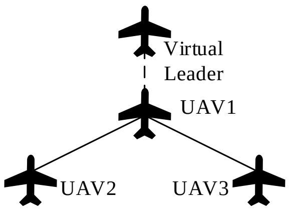


Fig. 3. Modified triangle formation diagram

C. Controller Design

Actually, the desired state $\mathbf{s}_i^d$ in (4) is obtained by subtracting the formation interval $\mathbf{s}_i^I$ from the virtual leader state $\mathbf{s}_L$ as $\mathbf{s}_i^d = \mathbf{s}_L - \mathbf{s}_i^I$. After processed by error mixer, state error $\mathbf{e}_i$ is sent to the controller to get the control signal $\mathbf{c}_i$. This paper chooses PID as the controller. The controller transfer function is as follows:

$$PID(s) = k_P + k_I\frac{1}{s} + k_D\frac{k_N}{1 + k_N/s} \tag{8}$$

It's a typical PID controller with a differential filter used by MATLAB/Simulink [23] where uses $k_D k_N / \left(1 + k_N/s\right)$ to take the place of pure differential $k_D s$. k_P, k_I, k_D, k_N are the factors of proportion, integral, differential and filter terms. Finally, the block diagram of the formation control system with a virtual leader and a real wing plane is shown in Fig. 4.

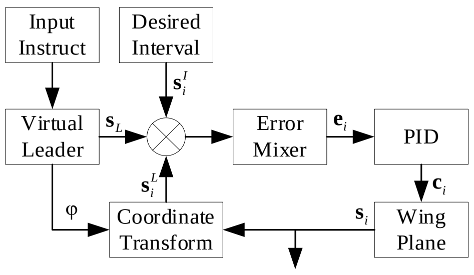


Fig. 4. Formation system block diagram

A problem to be aware of is that, in $\Delta = \left(\mathbf{s}_i^L - \mathbf{s}_i^d\right)$, the value of $\left(\Delta x, \Delta y, \Delta z\right)$ may be much larger than $\left(\Delta v, \Delta\mu, \Delta\varphi\right)$, and in order to quickly converge when the error is small, it is necessary to limit $\left(\Delta x, \Delta y, \Delta z\right)$ and adjust $\mathbf{k}_i$. Here apply saturation function to them and modify (7) as follows:

$$\begin{cases} \mathbf{e}_i = \mathbf{k}_i saturation\left(\mathbf{s}_i^L - \mathbf{s}_i^d\right) \\ \mathbf{c}_i = controller\left(\mathbf{e}_i^{sat}\right) \end{cases} \tag{9}$$

$\left(\Delta x, \Delta y, \Delta z\right)$ in $\Delta = \left(\mathbf{s}_i^L - \mathbf{s}_i^d\right)$ are limited by the formula below and $\left(\Delta v, \Delta\mu, \Delta\varphi\right)$ are keep unchanged:

$$\begin{cases} \Delta x^{sat} = sat\left(\Delta x, X_{\max}, X_{\min}\right) \\ \Delta y^{sat} = sat\left(\Delta y, Y_{\max}, Y_{\min}\right) \\ \Delta z^{sat} = sat\left(\Delta z, Z_{\max}, Z_{\min}\right) \end{cases} \tag{10}$$

Where sat is saturation function as bellow:

$$sat\left(x, high, low\right) = \begin{cases} low & x < low \\ x & low \le x \le high \\ high & x > high \end{cases} \tag{11}$$

Finally, here choose the parameters of the error mixer PID controllers and in TABLE I. and TABLE II. Besides, the factors in (1) are set up as $k_v = 0.01$, $k_\varphi = 0.1$ and $k_\mu = 0.1$, and the upper and lower bound of *x*, *y* and *z* in (10) are set to ± 50.

TABLE I. PARAMETERS OF ERROR MIXER

Axes	Parameters	
x	k_{ex}=1.5	k_{ev}=10
y	k_{ey}=0.1	$k_{e\varphi}$=10
z	k_{ez}=0.04	$k_{e\mu}$=5

TABLE II. PARAMETERS OF PID CONTROLLERS

DOF	Parameters			
	K_P	K_I	K_D	K_N
v	3.0	0.1	1.0	1.0
φ	5.0	1.0	1.0	1.0
μ	5.0	0.1	1.0	1.0

D. Simulation Test

Set up a simulation in SIMULINK to test the formation algorithm. It's reasonable to set only one leader and one wing plane for a centralized algorithm. Here set the leader initial state as $\mathbf{s}_L = (50,0,0,0,5000,1000)^T$ and that of wing plane is set to $\mathbf{s}_W = (60,0,0,1000,3000,800)^T$ and desired formation interval is $\mathbf{s}^d = (0,0,0,0,1000,0)^T$, which means the two UAVs are during the formation assembly process. The simulation result curves are shown in Fig. 6. During this process, the wing plane will first speed up to catch up with the leader in about 100 seconds and slow down and fine-tune the position as it approaches the formation position. The y-axis and z-axis errors can converge to zero in 50 seconds, while the x-axis and velocity errors converge to zero near 150 seconds. It shows that the design of the controller is successful.

As a centralized algorithm, the control progress is same for each plane in the formation, and a proper initial and target position can make no collision during the movement.

IV. FORMATION FLIGHT CONTROL SYSTEM

After the design of the model and controller is completed, a UAV simulation platform and a GCS can be built. The UAV simulation platform simulates the flight states of each UAV and receives instructions from GCS, that is, it simulates original ground stations receiving data from real UAVs. Here assumes that a ground station can control several UAVs and has high-speed communication links between stations. The GCS serves as the main Human-Machine Interface (HMI) to control and monitor the flight status of the UAVs. At the same time, the formation control thread and the collision detection thread in the background monitor the state of the UAVs and issue appropriate flight instructions. Structure diagram of the whole system shows in Fig. 5.

A. UAV Simulation Platform

The UAV simulation model is built in Simulink. Considering computational efficiency, control accuracy, communication time delay, etc., the fourth-order Runge-Kutta method is selected here and the simulation step size is set to 50 milliseconds. After tested in Simulink, the model is exported as C++ code which makes it possible to create multi-instance in GUI program to simulate several UAVs at the same time in a single application.

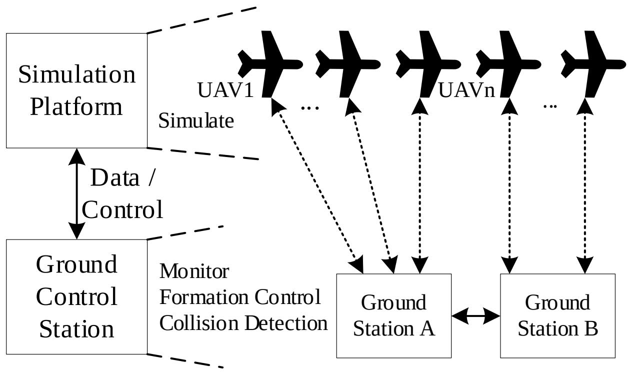


Fig. 5. System structure diagram

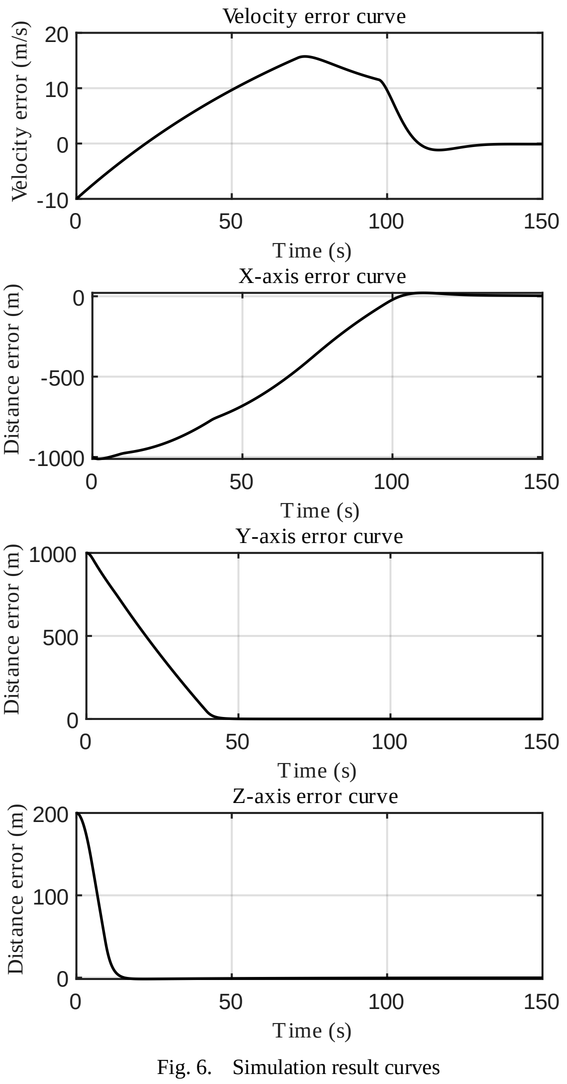


Fig. 6. Simulation result curves

The Qt framework is selected here to do GUI design work. Qt is open-source and cross-platform and can benefit from modern C++. The model is embedded in a separate thread with a timer waking it every 50ms. There is also a communication thread taking charge of exchanging data with GCS. And a GUI is equipped to show some state info of UAVs.

Ways to communicate between programs various. Here User Datagram Protocol (UDP) is adopted. UDP provides a way to send data fleetly without establishing a connection. As a network protocol, UDP makes it possible to deploy a GCS and simulation platform as a distributed system.

B. *UAV Ground Control Station*

The GCS is the information center and control center in the ground station system, which is used to obtain real-time flight data and control the UAVs. The GCS can receive and analyze the UAVs' flight data transmitted from the communication link, and display the UAVs' real-time latitude and longitude information, flight attitude data and real-time status to the operator. It also takes charge of uploading waypoint and monitoring the formation of the UAVs and changes the state of the mission load. The GCS software also adopts the Qt framework, and the system composition diagram is as follows in Fig. 7.

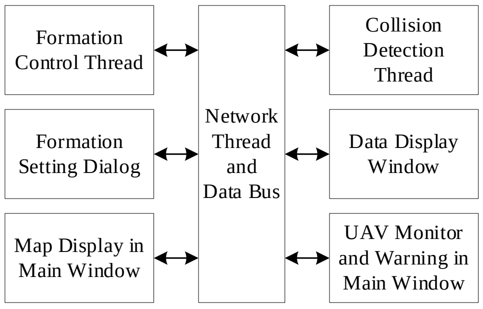

Formation Control Thread
Formation Setting Dialog
Map Display in Main Window
Network Thread and Data Bus
Collision Detection Thread
Data Display Window
UAV Monitor and Warning in Main Window

Fig. 7. GCS composition diagram

1) Formation control thread: The control algorithm is running in a sub-thread waking up when a set of UAVs' state data incoming. It is also responsible for the simulation of the virtual leader, and in each step, it will check the time-stamp of UAVs' and the leader's and keep them in sync. It will also receive collision detection data and do avoidance actions by letting the leader lead the slaves to change the flight altitude when a data link is broken. After finishing the algorithm introduced before, the command is sent to UAVs by communication thread.

2) Collision detection thread: Here, the frequency of collision detection is set to 1hz. Because the calculation frequency is quite different from the control thread, and in order to make the code structure clear, this article will also put the collision detection module in a sub-thread. In this part, the relative distance and velocity are calculated and collision warning data is sent to control thread and main window. This part also monitors the state of the communication link by calculating the difference between two data pack. If the data changes very little over a period of time, it can be considered the communication link has been broken.

3) Main window: As the main HMI, the main window takes charge of display various information and provides a setting interface. In order to display the real-time location and formation status of the UAVs and do path planning, Baidu Map is embedded in the program. Baidu Map provides JavaScrpit API for embedding in a webpage, and after modification and introduction of QWebChannel.js, it can be embedded in Qt and do two-way communication. UAVs' state and collision warning are also shown here. There are also pushbuttons calling for formation setting dialog, data display window and path planning function.

4) Network thread and data bus: Similar to UAV simulation platform, there is also a sub-thread responding for net working. It receives UAVs' data from simulation and sends to other modules. It also plays the role of a data bus, that is, the data of collision detection, formation control, map points, etc. are transmitted over it.

5) Other windows: Formation setting windows contains the parameters of UAV formation flight, and will send them to control thread after the confirmed. Data display window contains several charts to show the state of each UAVs.

The screenshot of the ground station when the system is running normally is shown in Fig. 9. It should be noted that, in the Formation Information box, a LED icon is used to show warning state of a UAV and a chain icon represents data link. The Error column shows formation error, the Distance column shows the shortest distance to other UAVs with the order of index, R distance and H distance, and the Tau column shows the minimum remaining time in (3) with the order of index and time.

In order to verify the algorithm embedded in the qt program, here performs simulation verification on the basis of the GCS and the Simulation Platform. Here set three UAVs as a triangle formation. First, let the UAVs fly autonomously. When the formation tends to be stabilized at around 200 seconds, manually cut off the data link of UAV3 in the Simulation Platform. In 5 seconds, the collision detection thread will determine that UAV3's data link has been interrupted and emit a warning signal. After receiving the signal, the formation control thread will let the virtual-leader plane lead other UAVs to climb 200 meters to avoid collisions. Later at around 235 seconds, the data link is recovered, and flight altitude will return to normal. Flight data curves are shown in Fig. 8, where the solid blue line represents UAV1, the orange dashed line is UAV2, and the yellow dotted line is UAV3. The two red dash-dot lines represent the time when the data link is manually operated. It show that the program can do accurate formation and do some collision avoidance.

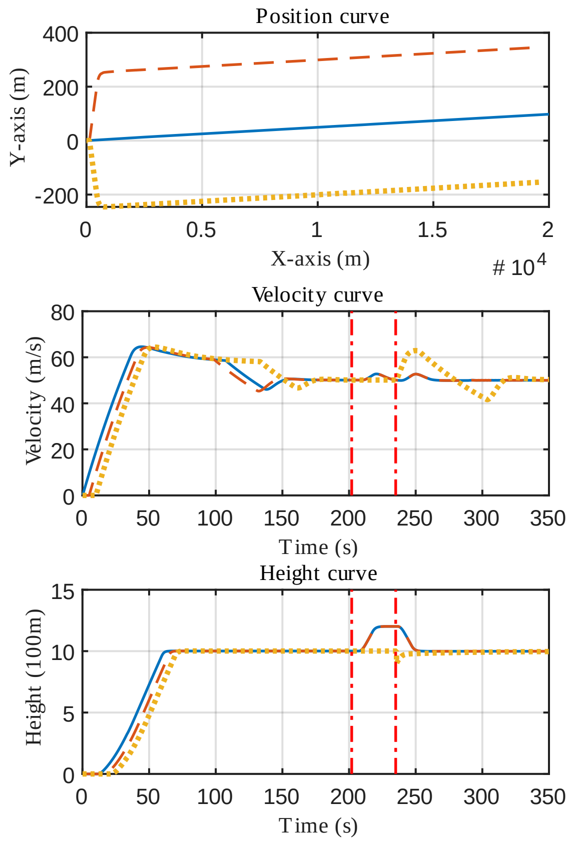
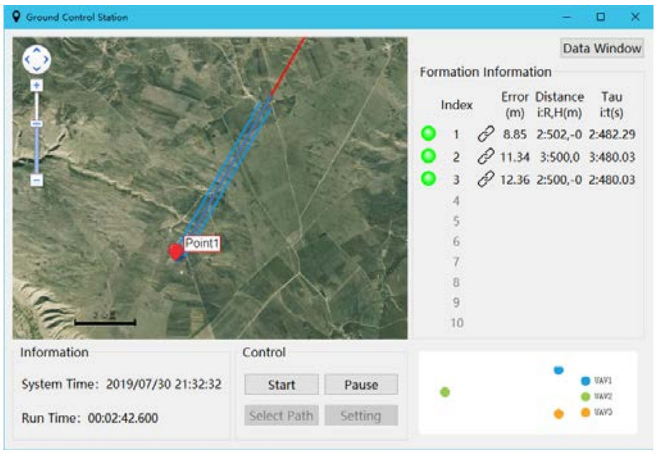


Fig. 8. GCS test flight data curve



Formation Information

	Index		Error (m)	Distance i:R,H(m)	Tau i:t(s)
●	1		8.85	2:502,-0	2:482.29
●	2		11.34	3:500,0	3:480.03
●	3		12.36	2:500,-0	2:480.03
	4				
	5				
	6				
	7				
	8				
	9				
	10				

Information

System Time: 2019/07/30 21:32:32

Run Time: 00:02:42.600

Control

Fig. 9. Screenshot of the GCS

V. Conclusions and Future Work

In this paper, a simple first-order three DOF model of UAV is built, and also build a collision detection model in space dimension and time dimension. After that, a centralized consensus-based UAV formation flight control algorithm is designed. After testing and verification in Matlab/Simulink, the UAV model and control algorithm is exported to C++ code and embedded in Qt-based simulation platform and GCS. The programs are also tested under the condition of manually cutting off the data link and successfully accomplish the formation and collision avoidance. The GCS provides a user-friendly interface while providing effective control for the UAVs. The separation of the GCS from the simulation platform allows it to be applied to semi-physical or physical simulations in the future.

Future work will be focused on the next lines:

1) Extend the model to six degrees of freedom and adjust the formation algorithm.

2) Add more obstacle and collision avoidance algorithm based on the collision detection model.

3) Integrate more sensor information in the GCS, such as camera, LIDAR, etc.

References

[1] S. d'Oleire-Oltmanns, I. Marzolff, K. Peter, and J. Ries, "Unmanned aerial vehicle (UAV) for monitoring soil erosion in Morocco," *Remote Sensing,* vol. 4, no. 11, pp. 3390-3416, 2012.

[2] F. Nex and F. Remondino, "UAV for 3D mapping applications: a review," *Applied geomatics,* vol. 6, no. 1, pp. 1-15, 2014.

[3] H. Saari *et al.*, "Unmanned Aerial Vehicle (UAV) operated spectral camera system for forest and agriculture applications," in *Remote Sensing for Agriculture, Ecosystems, and Hydrology XIII*, 2011, vol. 8174: International Society for Optics and Photonics, p. 81740H.

[4] K. L. Cook, "The silent force multiplier: the history and role of UAVs in warfare," in *2007 IEEE Aerospace Conference*, 2007: IEEE, pp. 1-7.

[5] D. H. A. Maithripala and S. Jayasuriya, "Radar deception through phantom track generation," in *Proceedings of the 2005, American Control Conference, 2005.*, 2005: IEEE, pp. 4102-4106.

[6] J. Wu *et al.*, "Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by Adaptive Grasshopper Optimization Algorithm," *Aerospace Science and Technology,* vol. 70, pp. 497-510, 2017.

[7] J. Kim and J. P. Hespanha, "Cooperative radar jamming for groups of unmanned air vehicles," in *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, 2004, vol. 1: IEEE, pp. 632-637.

[8] X.-x. Hu, Y. Chen, and H. Luo, "Robust decision making for UAV air-to-ground attack under severe uncertainty," *Journal of Central South University,* vol. 22, no. 11, pp. 4263-4273, 2015.

[9] X. Wang, V. Yadav, and S. Balakrishnan, "Cooperative UAV formation flying with obstacle/collision avoidance," *IEEE Transactions on control systems technology,* vol. 15, no. 4, pp. 672-679, 2007.

[10] A. Proud, M. Pachter, and J. D'Azzo, "Close formation flight control," in *Guidance, navigation, and control conference and exhibit*, 1999, p. 4207.

[11] Y. Zou, P. R. Pagilla, and R. T. Ratliff, "Distributed formation flight control using constraint forces," *Journal of guidance, control, and dynamics,* vol. 32, no. 1, pp. 112-120, 2009.

[12] J. Wu, H. Wang, N. Li, and Z. Su, "Formation Obstacle Avoidance: A Fluid-Based Solution," *IEEE Systems Journal*, 2019.

[13] W. Ren and Y. Chen, "Leaderless formation control for multiple autonomous vehicles," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, p. 6069.

[14] Z. Chao, S.-L. Zhou, L. Ming, and W.-G. Zhang, "UAV formation flight based on nonlinear model predictive control," *Mathematical Problems in Engineering,* vol. 2012, 2012.

[15] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: Theories and applications," *IEEE Transactions on Control Systems Technology,* vol. 23, no. 1, pp. 340-348, 2014.

[16] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 2, pp. 1123-1152, 2015.

[17] M. Pachter, J. J. D', Azzo, and A. W. Proud, "Tight formation flight control," *Journal of Guidance, Control, and Dynamics,* vol. 24, no. 2, pp. 246-254, 2001.

[18] J. Zhang *et al.*, "The collision avoidance control algorithm of the UAV formation flight," in *Proc. ARAA*, 2017, pp. 1-7.

[19] W. Ren and E. Atkins, "Nonlinear trajectory tracking for fixed wing UAVs via backstepping and parameter adaptation," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005, p. 6196.

[20] C. Livadas, J. Lygeros, and N. A. Lynch, "High-level modeling and analysis of the traffic alert and collision avoidance system (TCAS)," *Proceedings of the IEEE,* vol. 88, no. 7, pp. 926-948, 2000.

[21] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *International Journal of Critical Infrastructure Protection,* vol. 4, no. 2, pp. 78-87, 2011.

[22] R. Olfati-Saber and R. M. Murray, "Consensus Problems in Networks of Agents With Switching Topology and Time-Delays," *IEEE Transactions on Automatic Control,* vol. 49, no. 9, pp. 1520-1533, 2004, doi: 10.1109/tac.2004.834113.

[23] C. Bohn and D. Atherton, "A Simulink package for comparative studies of PID Anti-windup Strategies," in *Proceedings of IEEE Symposium on Computer-Aided Control Systems Design (CACSD)*, 1994: IEEE, pp. 447-452.