Formation Control in Swarm Systems

Ayça Kula
n20237285@cs.hacettepe.edu.tr
aycakula96@gmail.com

Oğuz Kaan Hancıoğlu
n21145489@cs.hacettepe.edu.tr
oguzkaanhancioglu@hotmail.com

1 Introduction

1.1 Problem Statement

In formation control problem there are three main tasks as: 1) A desired formation topology has to be formed which is called the formation generation task. 2) The desired shape must be kept in specific tasks. 3) When the system is subjected any obstacles, environmental issues etc., the system should re-create the desired shape[6].

1.2 Literature Review

In order to maintain the configuration and the behaviour of the formation and solve problems related with formation control various strategies such as leader-follower, consensus-based etc. methods has been proposed. The proposed methods can be seen more clearly in table 10. In [3], PID controller was used to track the desired trajectory and to keep the formation shape. For the formation of quadrotors in X and Y plane has been reached using lyapunov controller. Also an artificial fish swarm algorithm is used for increasing the performance of controllers. In order to control the leader-follower formation in [15] a sliding mode controller is used and velocity and angular velocity is used as an input to followers in this controller. Moreover, in [14] PID was used for controlling every quadrotor in system and for designing the formation control sliding mode was used.The formation controller generates the desired velocity for the follower. Therefore, a relative distance and orientation can be kept (at same height).

2 Quadrotor Modelling and Control

2.1 Quadrotor Modelling

All equations and parameters are taken from [11]. The author implementation was implemented in Matlab and Simulink and set up to simulate swarm formation. A quadrotor has four independent electric motors that are connected rigid body structure. Each motor has its own propeller in order to generate thrust. Since the angle of the blade is fixed, the angular velocity of the motor is the only control parameter. Because of their simplicity, the quadrotors are one of the most common tools for researchers. In this project, we will use quadrotor systems for formation control.

The quadrotor motor configurations, thrusts, and torques can be seen in Figure 2. The motors only generate positive thrust. Let's define $FW : \{OW; XW; YW; ZW\}$ as an inertial world frame and $FB : \{OB; XB; YB; ZB\}$ as a body frame of Center Of Mass. These coordinate frames can be seen in Figure 3.

There are three main rotations that are named roll, pitch, and yaw in the space. The roll, pitch, and yaw motion can be seen in Figure 4. We can model the quadrotor using these axes to complete dynamic model.

The torque for roll, pitch, and yaw axis can be calculated in Equation 1. We can obtain a dynamic representation of the quadrotor.

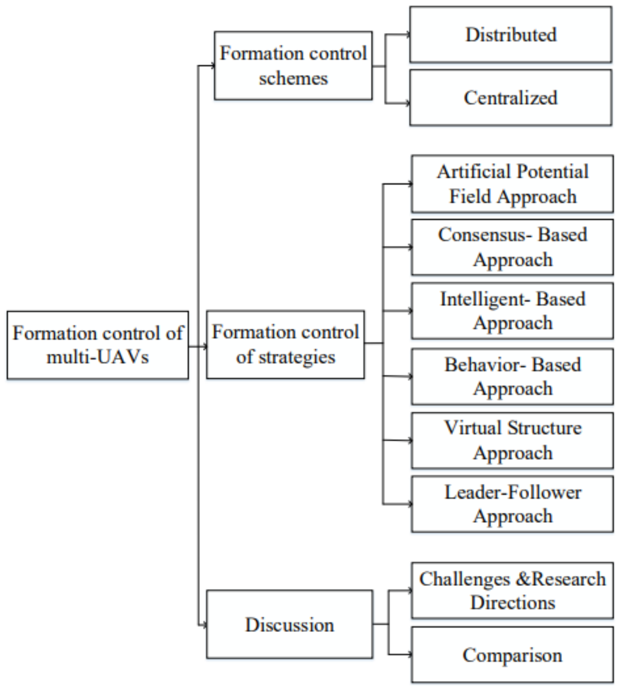

Figure 1: Formation Control of Multil-UAVs system [6]

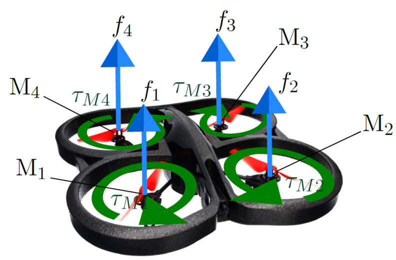

Figure 2: Quadrotor motors, thrusts and torques [11].

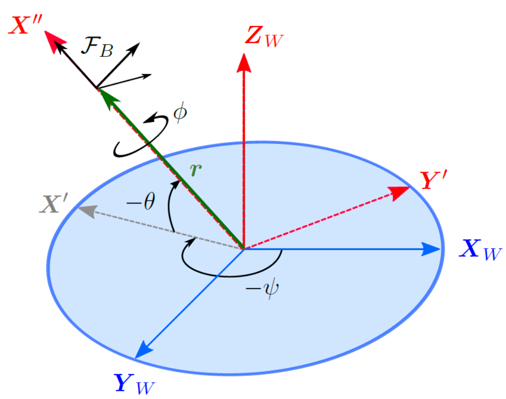

Figure 3: Inertial and body frame of the quadrotor [11].

$$\boldsymbol{\tau} = \begin{bmatrix} (f_2 - f_4)\, l \\ (f_3 - f_1)\, l \\ \sum_{i=1}^{4} \tau_{M_i} \end{bmatrix} = \begin{bmatrix} \tau_{\text{roll}} \\ \tau_{\text{pitch}} \\ \tau_{\text{yaw}} \end{bmatrix} \tag{1}$$

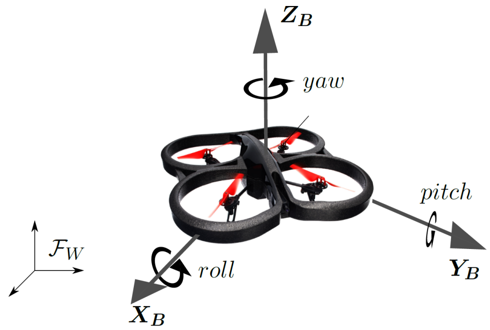

Figure 4: Roll, pitch and yaw axis of the quadrotor [11].

The dynamical equations of the quadrotor are given in Equation 2 - 7.

$$m\ddot{x} = f_{\mathrm{t}}(\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta) - k_{\mathrm{aer,x}}|\dot{x}|\dot{x} \tag{2}$$

$$m\ddot{y} = f_{\mathrm{t}}(\cos\phi\sin\theta\sin\psi - \cos\psi\sin\phi) - k_{\mathrm{aer,y}}|\dot{y}|\dot{y} \tag{3}$$

$$m\ddot{z} = f_{\mathrm{t}}\cos\theta\cos\phi - mg - k_{\mathrm{aer,z}}|\dot{z}|\dot{z} \tag{4}$$

$$\dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}}qr - \frac{I_{\mathrm{M}}}{I_{xx}}q\Omega + \frac{\tau_{\mathrm{roll}}}{I_{xx}} \tag{5}$$

$$\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}}pr + \frac{I_{\mathrm{M}}}{I_{yy}}p\Omega + \frac{\tau_{\mathrm{pitch}}}{I_{yy}} \tag{6}$$

$$\dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}}qr + \frac{\tau_{\mathrm{yaw}}}{I_{zz}} \tag{7}$$

Where the p, q, r are the body frame velocities, $\dot{\phi}, \dot{\theta}, \dot{\phi}$ are the euler angle velocities. We can simplify these dynamical equation by considering small $\dot{\phi}$ and $\dot{\theta}$. Therefore, the body frame velocities would equal to euler velocities. After the simplification the dynamics of quadrotor can be expressed in Equation 8.

$$\begin{aligned} \ddot{\phi} &= \tau_{roll}/I_{xx} \\ \ddot{\theta} &= \tau_{pitch}/I_{yy} \\ \ddot{\psi} &= \tau_{yaw}/I_{zz} \end{aligned} \tag{8}$$

2.2 Quadrotor Controller

The author uses a cascade control structure for both Position and Attitude Control as shown in Figure 5. PID controller is used in the both Position and Attitude Controller. The attitude controller controls the ϕ, θ, and ψ of the quadrotor by changing motor voltage. The position controller controls the x, y, and z of the quadrotor by changing the attitude. Since our aim simulation of the swarm system, we used both controllers and just changed the reference of the position controller.

Before implementing formation control, the quadrotor model with the controller is verified. We constructed a Simulink model and simulated using step input for the x and y-axis of the quadrotor. The step response of the position controller is shown in 6. The step response of the attitude controller is shown in 7. It is clear that both controllers works well and are suitable for formation control.

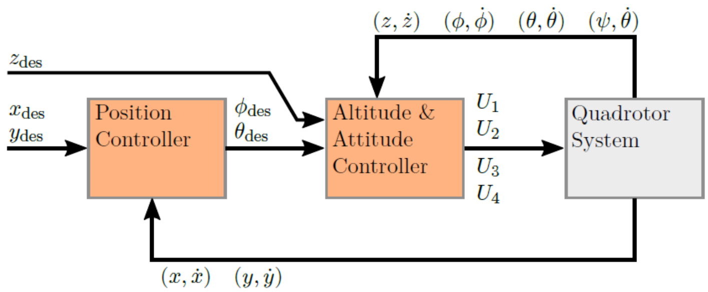

Figure 5: Control structure of the quadrotor [11].

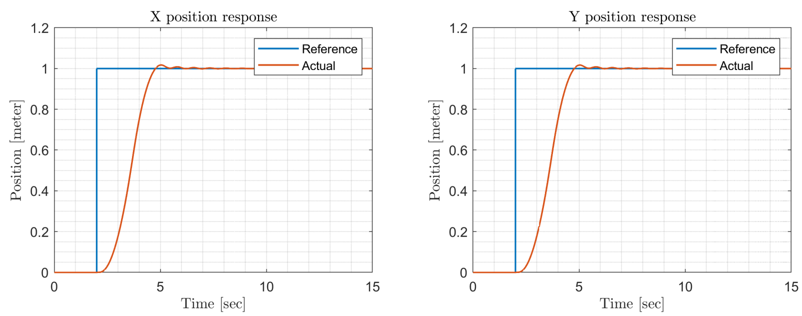

Figure 6: Reference and actual value of X and Y axis

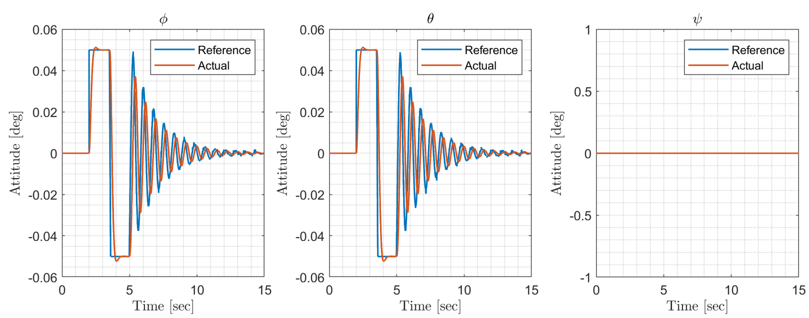

Figure 7: Reference and actual value of attitudes

3 Design

3.1 Leader-follower formation control

The goal of leader-follower formation control is to find the linear velocity and angular velocity for follower vehicle minimize the error value of distance and angle between leader and follower to zero [8].Formation is satisfied by maintaining the desired distance and angle between the leader and each follower. Two types of controllers that deals with: 1) Relative positions between each vehicle in the formation 2) Distance and angle between leader and follower, are implemented into this system [8].

3.1.1 Mathematical Implementation

The mathematical equations are obtained using [12]. The letter L denotes the body frame of leader and F denotes the body frame of the follower. In order to derive the position of leader relative to the follower (${}^{F}\mathbf{p}_L$) as in 10, we first write the Rotation matrix as:

$$\mathcal{R} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \tag{9}$$

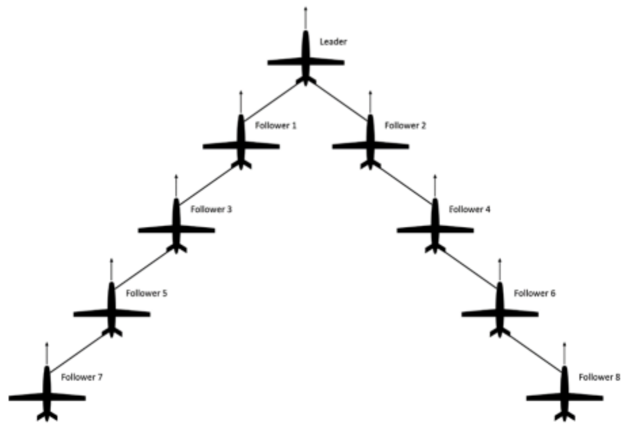

Figure 8: Leader follower configuration [4]

$$ {}^{F}\mathbf{p}_L = \mathcal{R}^T\left(\mathbf{p}_L - \mathbf{p}_F\right) \tag{10} $$

And, our control law design aim is to converge ${}^{F}\mathbf{p}_L$ to desired vector $d = [d_x d_y]$. Using the skew-symmetric matrix $(S(x))$ defined in 11, we have defined the the kinematic equations as in 12 and 13, where $u \in \mathbb{R}$ and $r \in \mathbb{R}$ are the linear and angular speeds.

$$ S(x) = \begin{bmatrix} 0 & -x \\ x & 0 \end{bmatrix}. \tag{11} $$

$$ \dot{\mathcal{R}} = \mathcal{R}S(r) \tag{12} $$

$$ \dot{\mathbf{p}}_F = \mathcal{R}\begin{bmatrix} u & 0 \end{bmatrix}^T \tag{13} $$

Since we are trying to reach the predefined desired distance we have to minimize the position error which is defined with respect to the actual distance between the leader and follower.

$$ \mathbf{e}_1 = {}^{F}\mathbf{p}_L - \mathbf{d} \tag{14} $$

And the time derivative of the error is defined as in 15 where our goal is to make $\dot{\mathbf{e}}_1$ zero.

$$ \dot{\mathbf{e}}_1 = -S(r)\left(\mathbf{e}_1 + \mathbf{d}\right) + R^T\dot{\mathbf{p}}_L - \begin{bmatrix} u \\ 0 \end{bmatrix} \tag{15} $$

A positive Lyapunov function is developed to converge the error term to zero.

$$ V_1 = \frac{1}{2k_1}\mathbf{e}_1^T\mathbf{e}_1 \tag{16} $$

Then the time derivative of this Lyapunov function corresponds to:

$$ \begin{aligned} \dot{V}_1 =& \frac{1}{k_1}\mathbf{e}_1^T\left[-S(r)\mathbf{d} + \mathcal{R}^T\dot{\mathbf{p}}_L - \begin{bmatrix} u & 0 \end{bmatrix}^T\right] \\ =& -\mathbf{e}_1^T\sigma\left(\mathbf{e}_1\right) + \mathbf{e}_1^T\left[\sigma\left(\mathbf{e}_1\right) + \frac{1}{k_1}\left(-S(r)\mathbf{d} + \mathcal{R}^T\dot{\mathbf{p}}_L - \begin{bmatrix} u & 0 \end{bmatrix}^T\right)\right] \end{aligned} \tag{17} $$

where σ is a saturation function defined using σ_K the K-saturation function as in 18.

$$ \boldsymbol{\sigma}(x) = \left[\sigma_K\left(x_1\right)\sigma_K\left(x_2\right)\right]^T \tag{18} $$

Using the definition in[12], σ_K has to satisfy some properties such as **1)** $\sigma_K(0) = 0$, **2)** $s\sigma_K(s) > 0$ for all $s \neq 0$, **3)** $\lim_{s\to\pm\infty}\sigma_K(s) = \pm K$ for some $K > 0$.

Therefore, the σ_K was defined as

$$\dot{\sigma}_K(x) \approx K \frac{\varepsilon}{(|x| + \varepsilon)^2} \dot{x} \tag{19}$$

where $\mathcal{E}$ is the constant defining the steepness of the saturation function.

A new error value is defined using the backstepping procedure,

$$\mathbf{e}_2 = \boldsymbol{\sigma}\left(\mathrm{e}_1\right) + \frac{1}{k_1}\left(-S(r)\mathbf{d} + \mathcal{R}^T \dot{\mathbf{p}}_L - [u]^T\right) \tag{20}$$

Using this error and the fist lyapunov function, we obtain the second lyapunov function as:

$$V_2 = V_1 + \frac{1}{2k_2}\mathbf{e}_2^T \mathbf{e}_2. \tag{21}$$

Now, $\dot{\mathbf{e}}$ can be written as:

$$\dot{\mathbf{e}}_2 = \dot{\boldsymbol{\sigma}}\left(\mathbf{e}_1\right) + \frac{1}{k_1}(\boldsymbol{\delta} - \boldsymbol{\Gamma}\boldsymbol{\mu}) + \mathbf{b} \tag{22}$$

where

$$\boldsymbol{\Gamma} = \begin{bmatrix} 1 & -d_y \\ 0 & d_x \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} T \\ \tau \end{bmatrix}, \quad \delta = -S(r)\mathcal{R}^T \dot{\mathbf{p}}_L + \mathcal{R}^T \ddot{\mathbf{p}}_L,$$

and $\mathbf{b} \in \mathbb{R}^2$ is an unknown constant.

The time derivative of the second lyapunov function is:

$$\dot{V}_2 = -\mathbf{e}_1^T \boldsymbol{\sigma}\left(\mathbf{e}_1\right) + \mathbf{e}_1^T \mathbf{e}_2 + \frac{\mathbf{e}_2^T}{k_1 k_2}\left(k_1 \dot{\boldsymbol{\sigma}}\left(\mathbf{e}_1\right) + \boldsymbol{\delta} - \boldsymbol{\Gamma}\boldsymbol{\mu} + k_1 \mathbf{b}\right) \tag{23}$$

To reject the disturbance $\mathbf{b}$ and provide tracking the state "ξ " is defined.

$$\dot{\xi} = \mathbf{e}_2 \tag{24}$$

and a new Lyapunov function and its derivative is given by

$$V_3 = V_2 + \frac{k_3}{2k_2}\left(\xi - \frac{1}{k_3}\mathbf{b}\right)^T \left(\xi - \frac{1}{k_3}\mathbf{b}\right), \tag{25}$$

$$\dot{V}_3 = \dot{V}_2 + \frac{k_3}{k_2}\left(\xi - \frac{1}{k_3}\mathbf{b}\right)^T \mathbf{e}_2 = -\mathbf{e}_1^T \boldsymbol{\sigma}\left(\mathbf{e}_1\right) + \mathbf{e}_1^T \mathbf{e}_2 + \frac{\mathbf{e}_2^T}{k_1 k_2}\left(k_1 \dot{\sigma}\left(\mathbf{e}_1\right) + \boldsymbol{\delta} + k_1 k_3 \xi - \boldsymbol{\Gamma}\boldsymbol{\mu}\right) \tag{26}$$

In [12], theorem 1 states that the simplified vehicle model described by 12, 13 and

$$\begin{aligned} \dot{u} &= T \\ \dot{r} &= \tau. \end{aligned} \tag{27}$$

and the error system with state given by $\mathbf{e}_1$, $\mathbf{e}_2$, and $\boldsymbol{\xi}' = \boldsymbol{\xi} - \frac{1}{k_3}\mathbf{b}$. The control law is defined as

$$\boldsymbol{\mu} = \Gamma^{-1}\left(\boldsymbol{\delta} + k_1 \dot{\boldsymbol{\sigma}}\left(\mathbf{e}_1\right) + k_1 k_2 \mathbf{e}_2 + k_1 k_3 \xi\right), \tag{28}$$

where k_1, k_2, and k_3 are the positive gains which renders the origin of the error system globally asymptotically stable.

Therefore the overall closed-loop system is defined as:

$$\begin{aligned} \dot{\mathbf{e}}_1 &= -S(r)\mathbf{e}_1 - k_1 \sigma\left(\mathbf{e}_1\right) + k_1 \mathbf{e}_2, \\ \dot{\mathbf{e}}_2 &= -k_2 \mathbf{e}_2 - k_3 \xi', \\ \dot{\xi}' &= \mathbf{e}_2. \end{aligned} \tag{29}$$

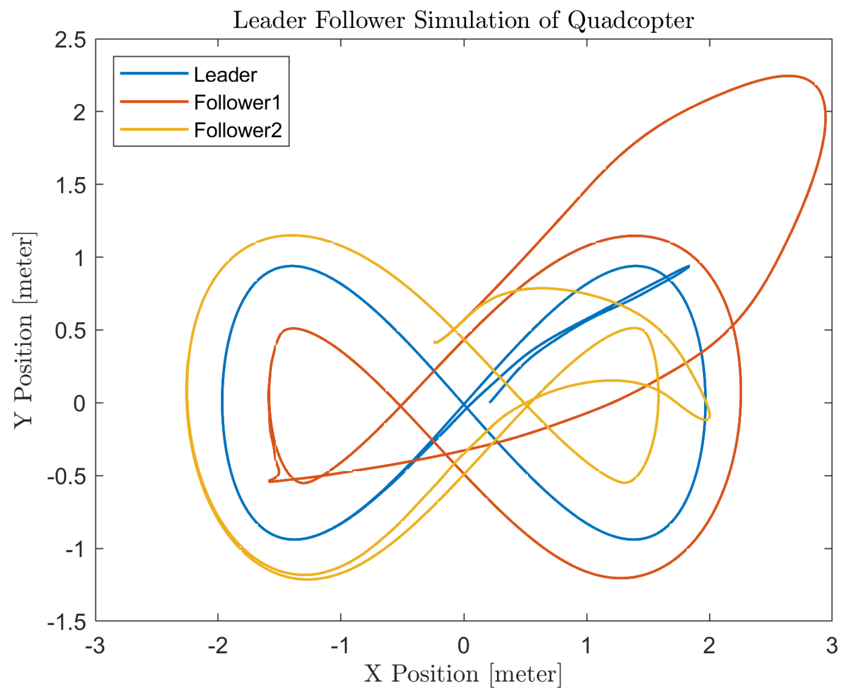

Figure 9: The result of leader follower algorithm

3.1.2 Simulation Results

The result of Leader-Follower is shown in Figure 9. While the blue line represents the leader, the others are the followers. The reference of the leader is taken as an infinity symbol. The Leader follower algorithm generates the references for both leader and followers. The reference is applied to the Simulink model. It is clear that leader and followers track the reference accurately.

3.2 Consensus-Based formation control

In the topic of swarm systems, consensus is a common state achieved for all the agents[16]. When there is restricted and dynamic changing communication topology, using consensus-Based formation control is an advantage [6]. According to [9] [7] [10] [16], a graph theory method can be used to design the information between the N UAVs in the consensus problem. The consensus protocol is designed as in 31 where there is fixed topology and no communication time delay. The values x_i and u_i indicate the states of the i^{th} agent and control input.

$$\dot{x_i} = u_i, i \in N\{1, 2, ..., n\} \tag{30}$$

$$u_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) \tag{31}$$

3.2.1 Mathematical Implementation

The graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ is a set of nodes, and $\mathcal{A} \in \mathcal{V} \times \mathcal{V}$ is a set of edges. Adjacency matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$ is given as:

$$a_{ij} = \begin{cases} 1, & \text{for } (v_j, v_i) \in \mathcal{A} \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

The degree matrix $\mathcal{D} \in \mathbb{R}^{N \times N}$ is,

$$\mathcal{D} = \operatorname{diag}\left(\deg\left(v_1\right), \deg\left(v_2\right), \cdots, \deg\left(v_N\right)\right) \tag{33}$$

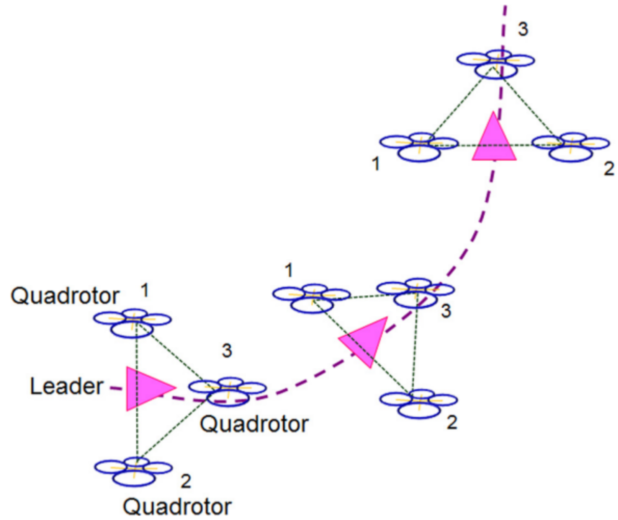

Figure 10: Desired formation for UAVs and leader [9]

The graph Laplacian $\mathcal{L} \in \mathbb{R}^{N \times N}$ is

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \tag{34}$$

In order to implement this graph, [5] suggests using Erdös-Renyi algorithm. In order to implement this algorithm into MATLAB [1] is used.

According to [13], the formula 35 has to be implemented for each agent.

$$u_i = -\sum_{j=1}^{N} a_{ij}(x_j - x_i) \tag{35}$$

And, for every quadcopter a reference consensus module is used for keeping the each vehicles position in space.

Each agent experience change of reference state through formation and each of the agents has its own state $\boldsymbol{\xi}_i$.

$$\boldsymbol{\xi}_i = \begin{bmatrix} x_{c,i} & y_{c,i} & z_{c,i} & \alpha_i & \beta_i & \gamma_i \end{bmatrix} \tag{36}$$

Our goal is to drive $\boldsymbol{\xi}_i$ to desired value $\boldsymbol{\xi}^r_{\text{contr}}$. Therefore the final consensus law is defined as:

$$\dot{\xi}_i = \underbrace{\frac{1}{\eta_i} \sum_{j=1}^{n} a_{ij} \left[\dot{\boldsymbol{\xi}}_j - \kappa \left(\boldsymbol{\xi}_i - \boldsymbol{\xi}_j \right) \right]}_{\text{Neighbor's influence}} + \underbrace{\frac{1}{\eta_i} a_{i(n+1)} \left[\dot{\xi}^r_{\text{contr}} - \kappa \left(\boldsymbol{\xi}_i - \boldsymbol{\xi}^r_{\text{contr}} \right) \right]}_{\text{Reference influence}} \tag{37}$$

For further studies the thesis [2] was used.

3.2.2 Simulation Results

The result of Consensus is shown in Figure 9. While the blue line represents the leader, the others are the followers. The reference of the leader is taken as an infinity symbol. The Consensus algorithm generates the references for both leader and followers. The reference is applied to the Simulink model. It is clear that leader and followers track the reference accurately.

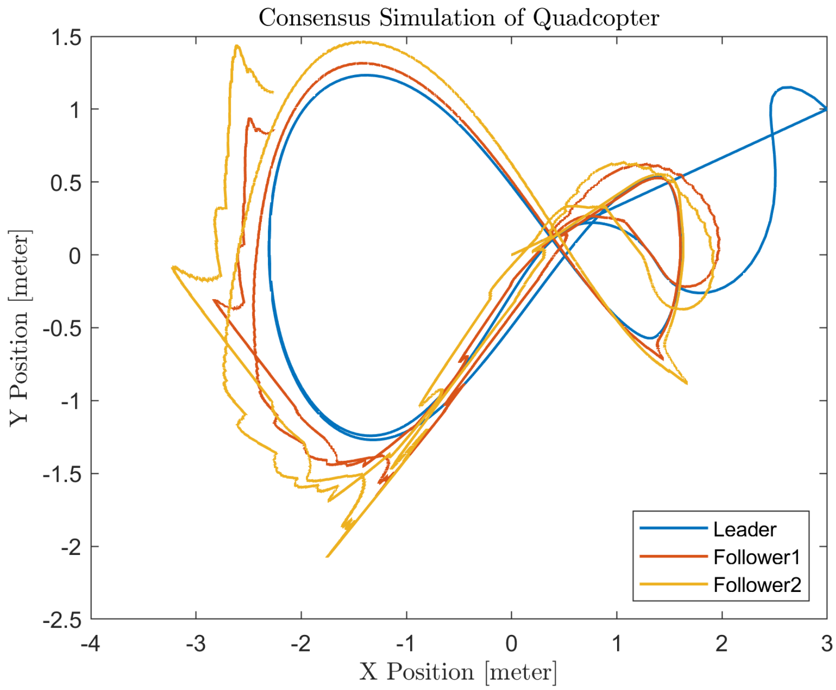

Figure 11: The result of consensus algorithm

4 Conclusions

This study provides a comparison between the leader-follower and consensus control problem. Leader-follower algorithm is more simple to implement and design compared to consensus problem.Two methods has good tracking performance as seen from the simulation results. However, in the leader follower problem there is a lot of dependence on the leader because no feedback is taken from the followers. This comprehensive evaluation may help future work and give insight about the control problems.

References

[1] Sept. 2011. URL: https://www.cs.purdue.edu/homes/dgleich/demos/erdos_renyi/generate.html.

[2] URL: https://www.jenspetit.de/2019/03/Petit_Jens_BA.pdf.

[3] Rabah Abbas and Qinghe Wu. "Improved Leader Follower Formation controller for multiple Quadrotors based AFSA". In: *TELKOMNIKA Telecommunication Computing Electronics and Control* 13 (2015), pp. 85–92.

[4] Zain Ali and Zhangang Han. "Multi-unmanned aerial vehicle swarm formation control using hybrid strategy". In: *Transactions of the Institute of Measurement and Control* 43 (Apr. 2021), p. 014233122110038. DOI: 10.1177/01423312211003807.

[5] Frederik Deroo and Sandra Hirche. "A MATLAB toolbox for large-scale networked systems". In: *at - Automatisierungstechnik* 61 (July 2013), pp. 506–514. DOI: 10.1524/auto.2013.0041.

[6] Hai Do et al. "Formation Control Algorithms for Multiple-UAVs: A Comprehensive Survey". In: *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems* 8 (June 2021), p. 170230. DOI: 10.4108/eai.10-6-2021.170230.

[7] Yasuhiro Kuriki and Toru Namerikawa. "Consensus-based cooperative formation control with collision avoidance for a multi-UAV system". In: June 2014, pp. 2077–2082. ISBN: 978-1-4799-3274-0. DOI: 10.1109/ACC.2014.6858777.

[8] Yuanchang Liu and Richard W. G. Bucknall. "A survey of formation control and motion planning of multiple unmanned vehicles". In: *Robotica* 36 (2018), pp. 1019–1047.

[9] Toru Namerikawa, Yasuhiro Kuriki, and Ahmed Khalifa. "Consensus-Based Cooperative Formation Control for Multiquadcopter System With Unidirectional Network Connections". In: *Journal of Dynamic Systems, Measurement, and Control* 140 (Nov. 2017). DOI: 10.1115/1.4038375.

[10] Reza Olfati-Saber and Richard Murray. "Consensus problems in networks of agents with switching topology and time-delays". In: *IEEE Transactions on Automatic Control* 49 (Sept. 2004).

[11] Jens Petit. "Distributed Consensus-based Formation Control of Quadrotors with Formation Feedback". PhD thesis. University of Stuttgart, 2016.

[12] V. Roldão et al. "A leader-following trajectory generator with application to quadrotor formation flight". In: *Robotics and Autonomous Systems* 62 (Oct. 2014). DOI: 10.1016/j.robot.2014.05.002.

[13] R.O. Saber and R.M. Murray. "Consensus protocols for networks of dynamic agents". In: *Proceedings of the 2003 American Control Conference, 2003.* Vol. 2. 2003, pp. 951–956. DOI: 10.1109/ACC.2003.1239709.

[14] Falin Wu, Jiemin Chen, and Liang Yuan. "Leader-Follower Formation Control for Quadrotors". In: vol. 187. Dec. 2016, p. 012016. DOI: 10.1088/1757-899X/187/1/012016.

[15] Zhihui Zhao and Jing Wang. "Leader-Follower Formation Control of Multiple Quadrotors". In: *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*. 2018, pp. 1–6. DOI: 10.1109/GNCC42960.2018.9019167.

[16] Bing Zhu, Lihua Xie, and Duo Han. "Recent developments in control and optimization of swarm systems: A brief survey". In: *2016 12th IEEE International Conference on Control and Automation (ICCA)*. 2016, pp. 19–24. DOI: 10.1109/ICCA.2016.7505246.