

Network Communication in Quadcopter Formation Flight

Jeff Hindsborg, Ricard Bordalba, Jorge Val.

Department of Electronic Systems, Faculty of Engineering and Science, Aalborg University, Denmark.

E-mail: 14gr730@es.aau.dk

Abstract—This paper studies the behaviour of a network in a quadcopter controller, designed for both single and formation flight applications. A Linear Quadratic Regulator (LQR) was applied, as it yielded optimal results. The performance of the controller is tested in different scenarios of data loss probability and path-loss. The limits of the quadcopter controller caused by loss of data are found. Results shows that the simulated system is stable until package loss probability reaches approximately 50%. The maximum distance between receiver and transmitter, in terms of communication breakdown, is also simulated. Both studies can be applied to any networked systems with this control setup. Finally, flight formation restrictions are found.

Index Terms—LQR, flight formation, quadcopter, real-time communication, wireless network.

I. INTRODUCTION

The increasing rate of quadcopter's developments in the last few years has permitted Unmanned Aerial Vehicles (UAV) to be used in many applications, such as flight formation. Flight formation could be applied in areas that are hardly accessible to search and rescue teams, e.g. due to a natural disaster, in order to locate people in distress in time. In these situations, rapid response is critical and delay can result in the loss of human life. Flight formation allows the quadcopters to cover a larger area in a shorter period of time in order to give a faster location of possible points of interest. To accomplish this, a fast and reliable network communication between the quadcopters and the ground station is crucial.

The department of Computer Science of University of Oxford has already studied the use of quadcopters in rescue operations, with a main focus on different search algorithms [1]. Some control strategies for quadcopter flight formation have been analysed in [2]. Networks have been previously used in Universiti Tun Hussein Onn Malaysia [3] with respect to the design of a wireless quadcopter controlled through a GUI interface but ignoring the study of the possible disturbances in the network. This paper investigates some flight limitations to a quadcopter formation flight, with a specific control and network design.

The behaviour of a networked quadcopter flight formation have been investigated through different simulations with respect to potential network disturbances. During surveillance flights, the quadcopters will be flying in formation with a fixed position between each other as shown in Fig. 1, such that no area remains unscanned. This relative distance vector will be maintained by a continuous inter-communication between the

quadcopters, in a distributed real-time network, such that the quadcopters will not collide into each other. All units will have a short range communication module, while one will have both short and long range in order to receive the global position from a ground station. This will limit the heavy power consumption to one quadcopter, since long range communication can be expensive in terms of battery. These two networks are analysed along the paper. Whenever a disaster strikes, certain

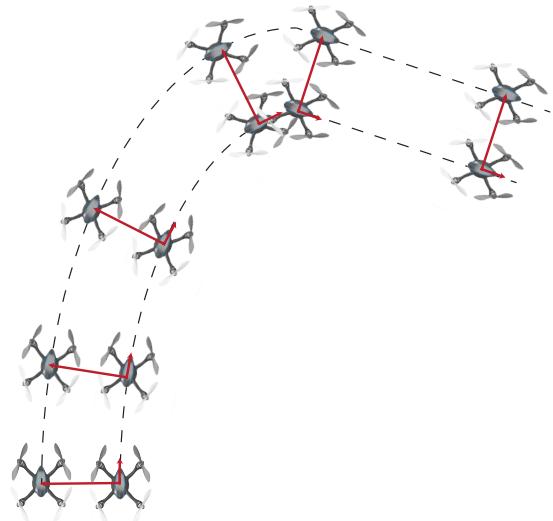


Fig. 1. Flight formation representation.

infrastructures could be damaged causing a noisy environment, which may affect network communications. Thus, data losses and communication failures are very common. The first aim of this project is to design a controller which can handle the effects of these setbacks and still provide a proper and precise autonomous flight. Formation flight scenario is studied in the control where linearisation of the quadcopter model is done in order to simplify the controller design. These assumptions imply some limitations in their possible movements during flight formation. Furthermore, network failures can occur and affect the quadcopter response.

This paper studies the effect of data loss and path-loss in a network between ground station and a quadcopter in a flight. Nevertheless, the conclusions extracted from the results are applicable to any quadcopter flight formation with a similar communication setup. Section II describes the quadcopter

model, as well as the corresponding discrete Linear Quadratic Regulator (LQR) controller design. The results simulated in Section III show the behaviour of the selected model in different scenarios, where data loss probability, path-loss and flight formation performance are studied. Finally, Section IV summarizes and concludes the paper.

II. SETUP AND METHODS

In the following subsections a quadcopter model is developed which will be used to design a controller. Also, the network setup for the quadcopter is defined.

A. Model

While working with quadcopters or drones in 3D-space, two frames of reference are necessary to fully describe the system. These are shown in Fig. 2. The first coordinate system is referred to as the 'global frame' (often called the earth fixed frame), while the other will be the 'local body frame'. The global frame is a fixed reference frame and the local frame will follow the attitude of the quadcopter rotating. The relationship between both frames is described through the Euler angles [10] (pitch, roll and yaw) and its rotational matrices, which convert from global to local or vice-versa.

Mainly, the four forces and four torques in Fig. 2 describe all the movements of the quadcopter. Throttle is produced by the sum of the 4 motor forces (F_1, F_2, F_3 and F_4), pitch is generated by the front and back motor (F_3 and F_4), roll by the left and right motor (F_1 and F_2) and finally, yaw is produced by the four motor torques (τ_1, τ_2, τ_3 and τ_4).

Moreover, it is known that both forces and torques are directly related to the motor's square angular velocity, Ω_i^2 , by a proportional factor and they will be used to define the input of the system [12]:

$$\begin{aligned} F_i &= K_F \cdot \Omega_i^2 \\ \tau_i &= K_\tau \cdot \Omega_i^2 \end{aligned} \quad (1)$$

Where K_F and K_τ are proportional coefficients, which depends on blade features, such as angle of attack, size of the blade and air density.

Dynamics are defined using Newton-Euler equations [8] which describe both rotational and translational dynamics of a rigid body. For this reason, the quadcopter is hereby considered to be a solid body with its Centre Of Gravity (COG) at its geometrical centre.

Translational dynamics are expressed in the global frame as the quadcopter position is known in the global frame.

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix}_g = \frac{1}{M} \begin{bmatrix} -(s\psi s\phi + c\phi c\psi s\theta)F_M \\ (s\phi c\psi - c\phi s\theta s\psi)F_M \\ Mg - c\theta c\phi F_M \end{bmatrix} + \begin{bmatrix} \omega_\psi v_y - \omega_\theta v_z \\ \omega_\phi v_z - \omega_\psi v_x \\ \omega_\theta v_x - \omega_\phi v_y \end{bmatrix} \quad (2)$$

where M is the total mass of the quadcopter, g is the gravitational acceleration, the total force $F_M = F_1 + F_2 + F_3 + F_4$, and $[v_x, v_y, v_z]^T$ is the velocity in the global frame, $s\alpha = \sin(\alpha)$, $c\alpha = \cos(\alpha)$ and

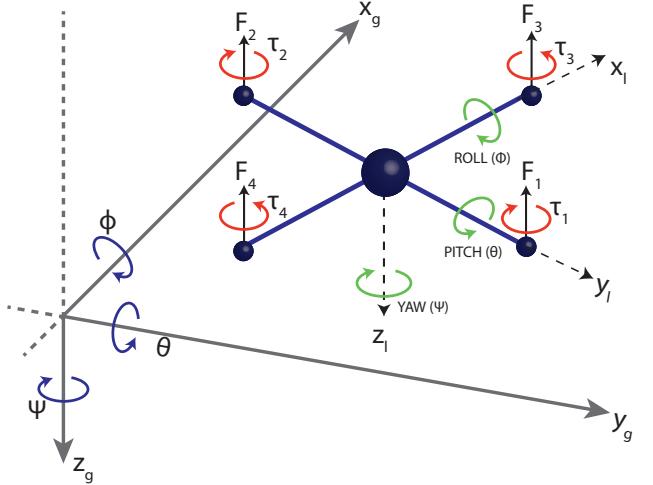


Fig. 2. The global frame is depicted to the left with the euler angles, while the local frame is depicted to the right with its corresponding angles, torques and forces definitions.

$[\omega_\phi, \omega_\theta, \omega_\psi]^T$ are the Euler angular velocities.

On the other hand, rotational dynamics are better expressed in the local frame since [4]:

- The inertia matrix I_{CM} around the centre of masses is constant.
- Body symmetry simplifies the equations.
- Measurements from the gyroscope sensor are in the local frame.

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix}_l = \begin{bmatrix} \frac{L}{I_{xx}}(F_2 - F_1) \\ \frac{L}{I_{yy}}(F_3 - F_4) \\ \frac{1}{I_{zz}}(-\tau_3 - \tau_4 + \tau_1 + \tau_2) \end{bmatrix} + \begin{bmatrix} \omega_y \omega_z \frac{I_{yy} - I_{zz}}{I_{xx}} \\ \omega_x \omega_z \frac{I_{zz} - I_{xx}}{I_{yy}} \\ \omega_y \omega_x \frac{I_{xx} - I_{yy}}{I_{zz}} \end{bmatrix} \quad (3)$$

where $[\omega_x, \omega_y, \omega_z]^T$ are the angular velocities around the local frame axis, L is the length of the quadcopter arm and $[I_{xx}, I_{yy}, I_{zz}]^T$ the inertial moments around each axis of the local frame.

Also, as it is desired to control the Euler angles, another equation is used based on the rotational matrices:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \omega_\phi \\ \omega_\theta \\ \omega_\psi \end{bmatrix} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi s\theta & c\phi s\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (4)$$

where $t\alpha = \tan(\alpha)$, $s\alpha = \sec(\alpha)$.

Finally, a last differential equation is needed to obtain the position in the global frame:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_g = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (5)$$

where $[x, y, z]^T$ is the position in the global frame.

B. Linearised model

The model is simplified through linearisation, in accordance with the following assumptions:

- Both pitch (θ) and roll (ϕ) angles are considered to be very small and close to 0 during the flight in order to approximate $\sin \alpha \approx \alpha$ and $\cos \alpha \approx 1$.
- Yaw angle (ψ) must be controlled and set to 0 as it simplifies the model, thus $\sin(\psi) = 0$ and $\cos(\psi) = 1$.
- The second term from both 2 and 3, which is considered to be the Coriolis effect, is neglected.
- The force F_M from the terms v_x and v_y is assumed to be constant and equal to the minimum force needed to hover: $F_M = F_{\text{hover}} = Mg$.
- Angular velocity is considered to be equal in both local and global frame based on the consideration that small deviations in the Euler angles during steady flight are produced in the quadcopter movement.
- Forces F_i and Torques τ_i will now be expressed as a function of the propellers angular velocity as stated in (1).

These assumptions result in the following linear equations which, altogether with (5), determine the model used to design a controller:

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} \approx \begin{bmatrix} -g\theta \\ g\phi \\ g - \frac{1}{M}K_F(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} \dot{\omega}_x^l \\ \dot{\omega}_y^l \\ \dot{\omega}_z^l \end{bmatrix} = \begin{bmatrix} \frac{LK_F}{I_{xx}}(\Omega_2^2 - \Omega_1^2) \\ \frac{LK_F}{I_{yy}}(\Omega_3^2 - \Omega_4^2) \\ \frac{K_\tau}{I_{zz}}(-\Omega_3^2 - \Omega_4^2 + \Omega_1^2 + \Omega_2^2) \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \omega_x^l \\ \omega_y^l \\ \omega_z^l \end{bmatrix} \quad (8)$$

C. Controller

From (5), (6), (7) and (8) a linear state-space model is developed where $[x, y, z, \phi, \theta, \psi, v_x, v_y, v_z, \omega_x^l, \omega_y^l, \omega_z^l]^T$ are considered the states of the system and $[\Omega_1^2, \Omega_2^2, \Omega_3^2, \Omega_4^2]^T$ the inputs. Gravity effect is not considered in the design and it is assumed that all states are available. A feedback gain K_d is designed and implemented as $u[n] = -K_d(x[n] - x_{\text{ref}}[n])$, which is represented in Fig. 3, for the discrete-time state space model [5]:

$$x[n+1] = \mathbf{A}_d x[n] + \mathbf{B}_d u[n] \quad (9)$$

where \mathbf{K}_d is a matrix with as many rows as inputs (4) and as many columns as states (12). Besides, \mathbf{A}_d and \mathbf{B}_d are the discrete matrices of the plant.

The discrete LQR design determines the matrix \mathbf{K}_d which minimizes the quadratic cost function of the following form:

$$J(u) = \sum_{n=1}^{\infty} (x[n]^T \mathbf{Q} x[n] + u[n]^T \mathbf{R} u[n]) \quad (10)$$

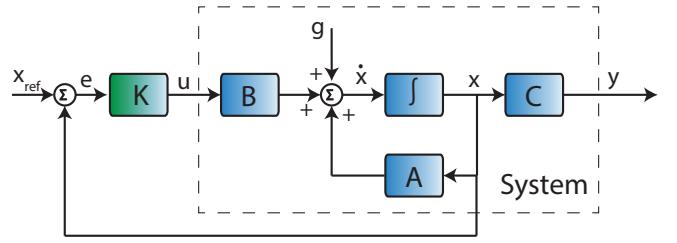


Fig. 3. State Space Feedback Block Diagram.

where \mathbf{K}_d is obtained by:

$$\mathbf{K}_d = (\mathbf{B}_d^T \mathbf{S} \mathbf{B}_d + \mathbf{R})^{-1} (\mathbf{B}_d^T \mathbf{S} \mathbf{A}_d) \quad (11)$$

And S is found by solving the associated Riccati equation [13]:

$$\mathbf{A}_d^T \mathbf{S} \mathbf{A}_d - \mathbf{S} - (\mathbf{A}_d^T \mathbf{S} \mathbf{B}_d)(\mathbf{B}_d^T \mathbf{S} \mathbf{B}_d + \mathbf{R})^{-1} (\mathbf{B}_d^T \mathbf{S} \mathbf{A}_d) + \mathbf{Q} = 0 \quad (12)$$

All these calculation are computed in Matlab using the command $dlqr(\mathbf{A}_d, \mathbf{B}_d, \mathbf{Q}, \mathbf{R})$, which gives \mathbf{K}_d that fulfils the previous statements. Moreover, a first choice for the matrices \mathbf{Q} and \mathbf{R} is done using Bryson's rule [6]:

$$\mathbf{Q} = \begin{bmatrix} \frac{\alpha_1^2}{(x_1)_{\max}^2} & 0 & 0 & \dots \\ 0 & \frac{\alpha_2^2}{(x_2)_{\max}^2} & 0 & \dots \\ 0 & 0 & \ddots & \\ \vdots & \vdots & & \frac{\alpha_n^2}{(x_n)_{\max}^2} \end{bmatrix} \quad (13)$$

$$\mathbf{R} = \begin{bmatrix} \frac{\beta_1^2}{(u_1)_{\max}^2} & 0 & 0 & \dots \\ 0 & \frac{\beta_2^2}{(u_2)_{\max}^2} & 0 & \dots \\ 0 & 0 & \ddots & \\ \vdots & \vdots & & \frac{\beta_n^2}{(u_n)_{\max}^2} \end{bmatrix} \quad (14)$$

where $(x_i)_{\max}$ and $(u_i)_{\max}$ are the maximum desired values of the states and the control inputs, while α and β are used to define the weighting of each state and control input in the controller design.

Two considerations are done in order to find \mathbf{R} and \mathbf{Q} , where \mathbf{R} is a square 4x4 and \mathbf{Q} is a 12x12 matrix.

- Not all states are desired to be controlled, so the weighting α of the states $v_x, v_y, v_z, \omega_\phi, \omega_\theta$ and ω_ψ is set to 0.
- Controlling position $[x, y, z]^T$ is not the main goal, but $[\phi, \theta, \psi]^T$ is because it is essential to assure that pitch, roll and yaw angles are approximately 0 in order to fulfil the linearisation assumptions and to keep the controller working. Thus, the highest α coefficient is set to the states ϕ, θ and ψ .

D. Network

From the mathematical model and the controller design, a simulation of the network communication is performed with the purpose of testing the behaviour of the quadcopter under certain circumstances where there might be communication breakdowns.

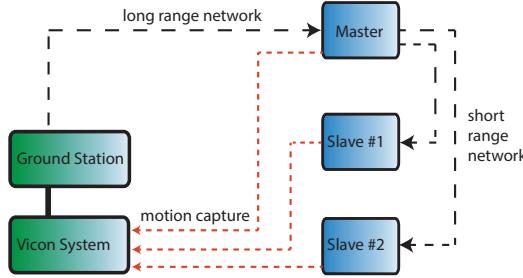


Fig. 4. Networking block diagram.

The simulation is performed through TrueTime 2.0 Beta [7]. This represents the real-time communication between the quadcopter and ground-station as represented in Fig. 4 and also includes the discrete system of the microcontroller, where the computation of the input signal is done. A wireless network simulates the connection between a ground station and the quadcopters where the ground station is used to send the desired positions $[x_{ref}, y_{ref}, z_{ref}]^T$ and the current positions $[x, y, z]^T$ to the quadcopters. Current position is measured with a motion capture system, specifically a Vicon system, connected to the ground station as shown in Fig. 4 and the link between them is considered fast enough so it does not affect the overall communication, hence it is omitted from the simulation. Finally, the data received is processed in the quadcopter microcontroller, where a control algorithm is coded. Thus, the feedback control process is completed.

III. RESULTS

This paper intends to analyse the network communication and the performance of the quadcopter in different scenarios, where data loss probability and path-loss are studied and compared. In addition, the restrictions for the flight formation performance are analyzed.

A. Data Loss Probability

The behaviour of the attitude controller in response to a unitary step input, when 0%, 40% and 60% of the packages are lost, is shown in Fig. 5, 6 and 7 respectively. The reference signal in z, y and x is triggered at zero, three and six seconds. Furthermore, the maximum number of times a node will try to re-transmit a message is set to zero. Fig. 6 shows data loss probability around 40%, which slightly disturbs the hover control and increases the overshoot, and the Euler become more noisy than in the ideal case. However the model remains stable. Moreover, the results in Fig. 7 shows that the hover control is no longer stable when there is a data loss between 50% and 60%. After two seconds the hover control drops

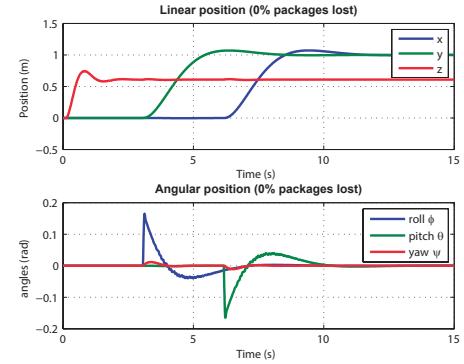


Fig. 5. Attitude control in a 0% loss probability scenario. The behaviour is observed under ideal circumstances in terms of networking communication.

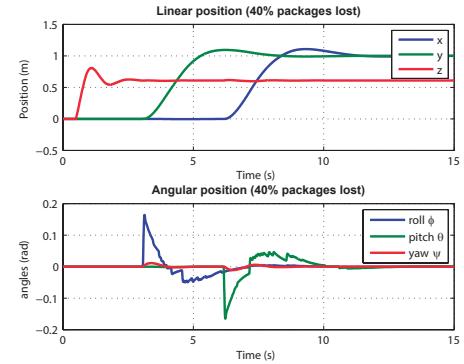


Fig. 6. Attitude control in a 40% loss probability scenario. The controller under these circumstances is slower and the overshoot increases compared with ideal scenario, nevertheless it remains stable.

and the quadcopter crash against the ground, subsequently the incident may damage the quadcopter, and reliability of the flight is no longer maintained. Additionally, the Euler angles remains stable, but are very close to surpass the angular limits, imposed by the linearisation of the model.

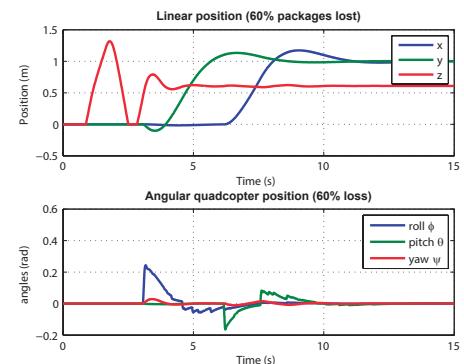


Fig. 7. Attitude control in a 60% loss probability scenario. After two seconds, the hover control drops colliding against the ground.

B. Path-Loss

For this simulation, the path-loss function is defined as:

$$P_{receiver} = \frac{1}{d^a} P_{sender} \quad (15)$$

Where P is the power, d is the relative distance in meters and a the parameter chosen to model the environment. Path-loss exponent is $a = 2$ in an ideal free space and can reach values between 4 and 6 [9] when the transmission is taking place in an environment with many obstacles such as buildings. This paper will depict the worst case scenario in terms of network. Hence the simulations in TrueTime are carried out with $a = 4$, the protocol of transmission, power of the transmitter, the receiver and data rate are modelled according to the specifications of the Xbee Pro S2B, 63mW [11], where the distance 'd' is a variable which depends directly on the current position of the quadcopter $[x, y]^T$.

According to the framework selected for the wireless network, transmit power is 18.00 dBm, receiver sensitivity is -100.00 dBm, then the maximum signal reach is calculated to 890.25 m. In the simulation, the quadcopter (receiver) is trying to follow a sinusoidal reference at the time it is flying away from the ground station (transmitter). The starting point is set at 875 meters in order to illustrate the communication failures with a proper resolution. The failure occurs due to the dissipation of the signal (path-loss).

The first graph in Fig. 8 shows a breakdown in the communication is shown when the quadcopter surpass the maximum operating distance, which is approximately 890 meters. On the second graph in Fig. 8 the behaviour of the quadcopter is shown with respect to the desired reference. It can be observed that once it reaches areas where the controller does not receive any data, the controller keeps processing control values according to the last data received. These data values remain constant in the controller after the breakdown in the communication. This makes the quadcopter keep flying in the same direction it had when the communication was lost, since the feedback error (reference received - position received) remains constant. Thus, the ground station loses all the control over the device.

C. Flight Formation Analysis

Communication between three quadcopters in flight formation (one master and two slaves) and the ground station is simulated. Through a short range network, the master distributes current locations and reference signals to the slaves, which will be applied for the formation configuration. The formation is an alignment over the y axis, with a relative distance of 2 meters between each quadcopter. Finally, the velocity in the y axis is studied in order to find the flight formation restrictions, that allow the quadcopters to stay in formation without crashing into each other.

The reference signal in the y position is sent to the quadcopters, once the hover control, z , has reached its settling time. The method used consists of giving a sinusoidal reference of the form $y_{ref}(t) = A \cdot \sin(\omega t)$, where A is the amplitude

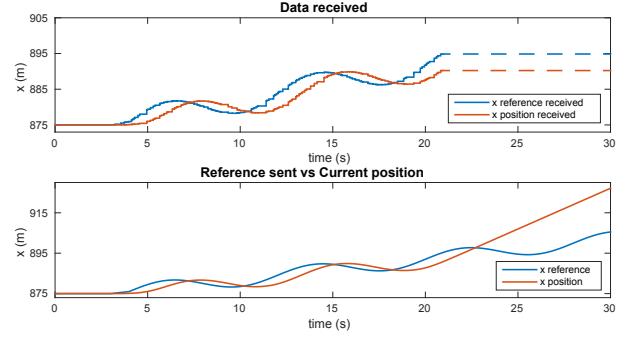


Fig. 8. The upper plot shows data received by the quadcopter from the ground station comparing the reference signal and the current position measured. The point where the communication is lost is shown with a dashed line. The lower plot shows the reference signal sent from ground station compared with the current position of the quadcopter.

and ω is the frequency of the input signal, to the master during a certain period of time in order to see the behaviour of the formation when there is a change in the y direction. The velocity is found by its derivative:

$$\frac{dy_{ref}(t)}{dt} = A\omega \cdot \cos(\omega t) \approx v_y(t) \quad (16)$$

The critical point of the flight, where they may collide, is when the relative distance between the quadcopters is minimal. This minimum is roughly reached when the quadcopters are flying at the maximum velocity which is given in (16) when $\cos(\omega t) = 1$. Results show how the slaves are not able to keep

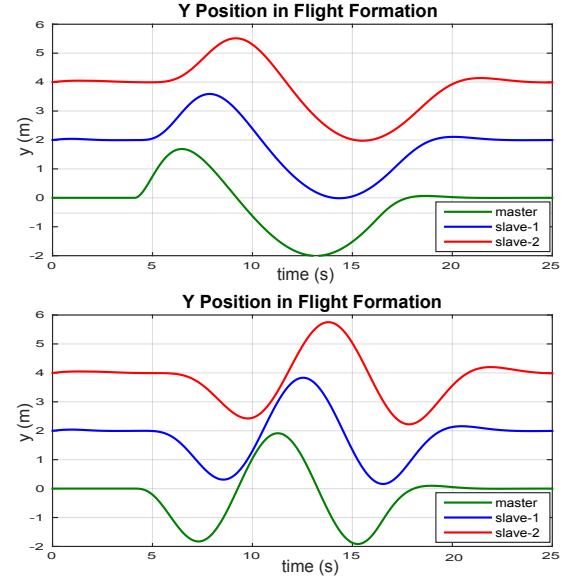


Fig. 9. The upper picture, shows the y position of the 3 quadcopters in flight formation reaching a maximum velocity of roughly 0.8 m/s. Safety distance between them is still maintained during all flight. The lower picture shows y position of 3 quadcopters in flight formation with a maximum velocity of roughly 1.6 m/s.

the safety conditions when the velocity in the y axis is higher than $v_{max} = 1.4$ (m/s). On the upper graph, in Fig. 9, flight

formation with a safety distance is maintained, while in lower graph the quadcopters are about to collide into each other.

IV. CONCLUSION

After studying the path-loss and data loss probability in a network between quadcopter and ground station in section III, the results infer that the most suitable scenario for a working controller is to have rates of data loss below 50% to maintain a fast and stable control output. On the other hand, the path-loss study proves that the power of the signal sent by the ground station is lost when the receiver is at a 890 meters away from the transmitter, when using the aforementioned XBee setup. Hence the ground station loses control of the quadcopter. This issue could be compensated by implementing an algorithm that makes the quadcopter go into hover mode and thus, stay in its current position when the signal is lost. In addition, the use of flight formation implies a limitation to the velocity in the alignment axis of the formation. When the quadcopters reaches a velocity more than 1.4 m/s collisions may occur, with this specific controller and safety distance.

The design of the control implies limitations to euler angles due to linearisation, which restricts the movements of the quadcopter.

Additionally, the path-loss and data loss probability simulations yields bounds to the setup used in this paper which is applicable to other control designs.

Finally, simulation of the flight formation provides a relationship between the minimum (or safety) relative distance between the units and the velocity in the axis of movement. This method can also be applied to similar controllers that does not specifically control the translational velocities.

ACKNOWLEDGMENT

The authors of this research would like to thank Henrik Schiøler for the help and guidance throughout this project and to the Department of Electronic Systems, at Aalborg University for providing the facilities and materials used to make it possible.

REFERENCES

- [1] Waharte, S. ; Trigoni, N., *Supporting Search and Rescue Operations with UAVs*, Emerging Security Technologies (EST), 2010 International Conference on DOI: 10.1109/EST.2010.51, Pages 142-147.
- [2] Weihua Zhaoa, Tiauw Hiong Go, *Quadcopter formation flight control combining MPC and robust feedback linearization*, Journal of the Franklin Institute, Volume 351, Issue 3, March 2014, Pages 13351355.
- [3] Dirman Hanafi, Mongkhun Qetkeaw, Rozaimi Ghazali, Mohd Nor Mohd Than, Wahyu Mulyo Utomo, Rosli Omar, *Simple GUI Wireless Controller of Quadcopter*, Int'l J. of Communications, Network and System Sciences, Vol. 6 No. 1, 2013, Pages 52-59.
- [4] Tristan Perez (2005), *Ship Motion Control: Course Keeping and Roll Stabilisation Using Rudder and Fins*, Springer Science & Business Media B.V.
- [5] Michael S. Triantafyllou, Franz S. Hover, *Maneuvering and control of Marine vehicles*, Department of Ocean Engineering, Massachusetts Institute of Technology
- [6] Department of Aeronautics and Astronautics, *Pole placement approach, Lecture Notes, Feedback Control Systems*, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology

- [7] Anton Cervin, Dan Henriksson, Bo Lincoln, Johan Eker, Karl-Erik rzn: "How Does Control Timing Affect Performance Analysis and Simulation of Timing Using Jitterbug and TrueTime." *IEEE Control Systems Magazine*, 23:3, pp. 1630, June 2003.
- [8] Roy Featherstone (2008). *Rigid Body Dynamics Algorithms*. Springer. ISBN 978-0-387-74314-1.
- [9] Julius Goldhirsh; Wolfhard J. Vogel. *Handbook of Propagation Effects for Vehicular and Personal Mobile Satellite Systems*.
- [10] Gregory G. Slabaugh, 1999, *Computing Euler Angles From a Rotation Matrix*. <http://staff.city.ac.uk/~sbbh653/publications/euler.pdf>
- [11] XBee /XBee-Pro RF Modules, *Data sheet from XBee Pro*, <https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>
- [12] S. Bouabdallah, 2007, *Design and control of quadrotors with application to autonomous flying*, EPFL, Lausanne, Switzerland
- [13] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini, 2010, *Feedback Control of Dynamic Systems* 6.ed, Prentice-Hall