Cooperative Load Transportation with Quadrotors

Tiago Bacelar[2], Carlos Cardeira[1,2], Paulo Oliveira[1,2]
[1]IDMEC, [2] Instituto Superior Técnico, Universidade de Lisboa, Portugal
{tiago.bacelar, carlos.cardeira, paulo.j.oliveira}@tecnico.ulisboa.pt

***Abstract*—This paper presents a methodology for cooperative load transportation with two quadrotors, given the state variables estimates, based on measurements from motion sensors installed on-board. The proposed controllers and estimators resort to optimal control techniques, namely, Linear Quadractic Regulators (LQRs) and Kalman filters, respectively. The proposed control system is validated both in simulation and experimentally, resorting to a commercially available quadrotor equipped with an Inertial Measurement Unit (IMU), an ultrasound height, vertical and frontal cameras, among other sensors.**

I. Introduction

In the last decades, the interest in Unmanned Aerial Vehicles (UAV) has increased due to the development and wide commercialization of these type of vehicles. They can operate in highly constrained environments with multiple obstacles, hostiles and impossible to reach by humans. For these reasons they became very useful for a wide range of civil and military applications, such as environment monitoring, surveillance, search and rescue missions and transportation of cargo. One of the major limitations on the available models is in terms of payload carrying capacity, being required the cooperation of several of these vehicles for several applications.

The manipulation of a towed cable system resorting to a aerial vehicle has been studied in [1]. A control method for the transportation of tethered known loads with a single quadrotor is proposed in [2], resorting to a Mixed Integer Quadratic Program focusing on aggressive maneuvering. Methods to tackle applications where the load is unknown are proposed, both for the estimation and the control of the height of a single quadrotor in [3]. Cooperative methods for cooperative manipulation and transportation using multiple aerial vehicles based on quasi-static models are proposed in [4], focusing on the position and orientation control of a payload with six degrees of freedom. The same problem is addressed in [5], studying the dynamics of cooperative manipulation resorting to a complete dynamical model for the cases when the payload is considered to be a point load, and a three-dimensional rigid body.

In this paper, a method for cooperative load transportation is proposed. It is assumed that the length of the load is greater than the distance between both UAVs as shown in Fig. 2, thus a configuration is envisioned where longitudinal forces are small and thus will be neglected. Moreover, the air flow cross disturbance can also be neglected due to the abovementioned geometry.

978-1-7281-3558-8/19/$31.00

A position control loop using LQRs is proposed for the rear UAV, and for estimation a linear Kalman filter is considered. The motion sensors used are a gyroscope, a magnetometer, a ultrasound sensor and a downward pointing camera for optical flow computation proposes, all usually available on-board a quadrotor. The on board Inertial Measurement Unit (IMU) allows the measurement and estimation of the height, attitude angles, accelerations, angular, and ground velocities.

This paper is organized as follows: the problem addressed in this paper is described in Section II. The physical model considered is presented in Section III. The solution for the control problem is proposed in Section IV, and the solution for estimation is presented in Section V. In Section VI simulation results are presented and discussed. In Section VII experimental results are presented and analyzed. Finally, some concluding remarks are presented in Section VIII.

II. Problem Statement

For the control systems design regarding the cooperative transportation of loads with two quadrotors, the controllability property plays a crucial role, such as stabilization of unstable systems, alongside the observability property which measures how well system's internal states can be inferred from knowledge of its own external outputs. The controllability and observability of a system are mathematical duals. A more in depth survey can be found in [6]. For the envisioned solution it is mandatory that both properties are verified. The main goal is to control the quadrotors 3D inertial position and orientation with respect to a reference North-East-Down (NED) frame. The two quadrotors to be considered are denoted as front UAV and rear UAV, which configuration is presented in Fig. 2.

Given the difficulties proving the system controllability and observability when considering the two UAVs as an ensemble with one common state space, a different approach is exploited. In this approach is taken into consideration the following assumptions:

- The front UAV is piloted by an operator
- The relative position and orientation between both UAVs is computed with respect to the rear UAV
- The effect of the load on the UAVs dynamics are neglected

In regard to control the front UAV position and orientation, externally to the rear UAV, a control system for the front UAV must be designed. Under these circumstances the front UAV will be able to maintain its position over a desired location, and therefore stably follow a designed trajectory autonomously, which is required to the solution proposed.

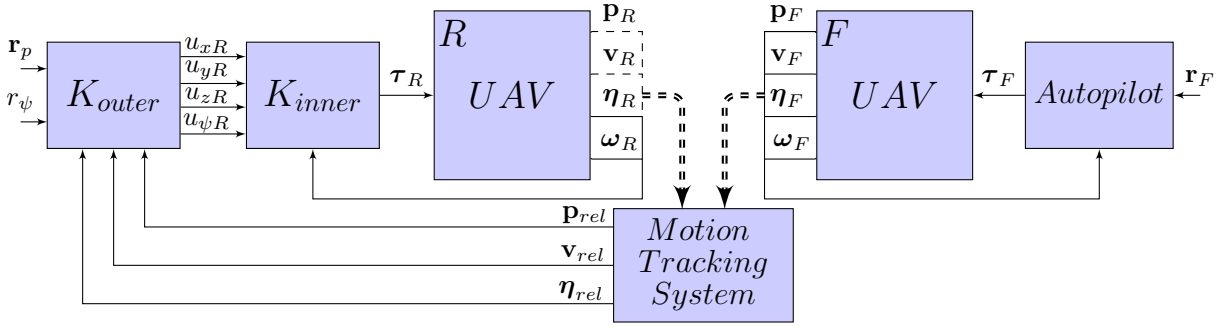

Fig. 1: Control System Architecture

The control systems are designed resorting to optimal control techniques, namely, linear quadratic regulators, which requires full state feedback. In order to reduce the noise impact present in the system measurements, and to ensure the full state feedback, discrete linear Kalman Filters will be designed.

The relative position between both UAVs is the main state to control, since the system's stability relies on this. Poor control of the relative position may lead to undesirable oscillations and twisting forces which can turn the system unstable resulting in the crash of both drones. Ideally, the rear UAV's frontal camera would be used to estimate the relative position, velocity and orientation with respect to the rear UAV. However, in this proposed solution a precision motion capture and 3D positioning tracking system is used to perform the estimation, namely, Qualisys motion capture system. The use of the camera is under work.

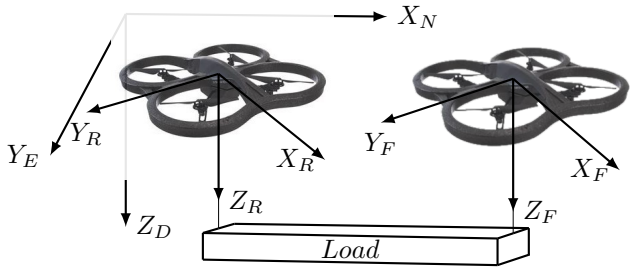

Fig. 2: Rear and Front UAV Configuration

III. Physical Model

The physical model is obtained resorting to the North-East-Down (NED) inertial frame and to the quadrotor body-fixed frame. Let $\mathbf{v} = (u, v, w)^T$ represent the linear velocity vector in the ENU inertial frame, $\boldsymbol{\eta} = (\phi, \theta, \psi)^T$ denote the orientation vector of the body-fixed frame with respect to the inertial frame in terms of Euler angles, and the body-axis angular rate vector be described by $\boldsymbol{\Omega} = (p, q, r)^T$.

The position behaviour where the influence of the Coriolis effects and the provided angular moments are defined in the following equation

$$M\dot{\mathbf{v}} = M\mathbf{g} + \mathbf{R}_{\mathcal{B}}^{\mathcal{I}}\mathbf{F}^{\mathcal{B}} - \mathbf{v} \times \boldsymbol{\omega} \tag{1}$$

where $\mathbf{g} = (0, 0, g)^T$ is the gravity vector with respect to the inertial frame, $\mathbf{R}_{\mathcal{B}}^{\mathcal{I}}$ is the rotation matrix responsible for rotate the quadrotor body-fixed frame into the NED inertial frame is presented in (2). $\mathbf{F}^{\mathcal{B}}$ corresponds to the total force applied in the quadrotor in terms of a vector with respect to the body-fixed frame presented in (3). Here $\mathbf{f}^{\mathcal{B}}$ is the actuation force vector, $\mathbf{R}_{\mathcal{I}}^{\mathcal{B}}$ is the rotation matrix that rotates the NED inertial frame into the body-fixed frame, and $\mathbf{T}$ is the tension force applied to the UAV which is assumed to be approximately null.

$$\mathbf{R}_{\mathcal{B}}^{\mathcal{I}}(\eta) = \begin{bmatrix} c\psi c\theta & -c\theta s\psi & s\theta \\ c\phi s\psi - c\psi s\theta s\psi & c\psi c\phi + s\psi s\theta s\phi & c\theta s\phi \\ -s\psi s\phi - c\psi c\phi s\theta & c\theta s\psi s\theta - c\psi s\phi & c\theta c\phi \end{bmatrix} \tag{2}$$

$$\mathbf{F}^{\mathcal{B}} = \mathbf{f}^{\mathcal{B}} + \mathbf{R}_{\mathcal{I}}^{\mathcal{B}}(\mathbf{v} \times \boldsymbol{\omega}) + \mathbf{T} \tag{3}$$

Substituting (3) in (1), yields

$$M\dot{\mathbf{v}} = M\mathbf{g} + \mathbf{f}^{\mathcal{I}} \tag{4}$$

where $\mathbf{f}^{\mathcal{I}} = \mathbf{R}_{\mathcal{B}}^{\mathcal{I}}\mathbf{f}^{\mathcal{B}}$.

The angular behaviour which takes into consideration Coriolis effects influence is presented in the following equation

$$\mathbf{I}\dot{\boldsymbol{\Omega}} = -\boldsymbol{\Omega} \times \mathbf{I}\boldsymbol{\Omega} + \boldsymbol{\tau} \tag{5}$$

where $\mathbf{I}$ is the inertia matrix and $\boldsymbol{\tau}$ is the torque vector that results from a combination of differences between the thrust forces generated by each of the four rotors.

IV. Control

In this Section the control solution is discussed. First the LQR is introduced followed by the controllers design for both UAVs.

A. Linear Quadratic Regulator

Consider the following linear time-invariant (LTI) dynamical system presented in (6). Here the vector $\mathbf{x}$ is the $n \times 1$ state vector, and the $m \times 1$ vector $\mathbf{u}$ is the control input. The control input is defined to be state feedbacks of the form is given in (7), where $\mathbf{K}$ is the state feedback control gain matrix that minimizes the quadratic cost function presented in (8).

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{6}$$

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \tag{7}$$

$$J = \int_0^\infty (\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u})dt \tag{8}$$

In this performance index, the matrices $\mathbf{Q}$ and $\mathbf{R}$ determine the relative importance of the energy associated to the states or to the control action, respectively. The control vector $\mathbf{u}(t)$ is assumed to be unconstrained. The infinite time quadractic optimal control problem relies on the minimization of the cost function J. The optimal gain matrix for the infinite time quadratic optimal control problem is linear and is given in (9). The matrix $\mathbf{P}$ in (9) must satisfy the algebraic Riccati equation presented in (10).

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} \tag{9}$$

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0 \tag{10}$$

The optimal state feedback control gain matrix $\mathbf{K}$ is given by substituting the matrix $\mathbf{P}$ obtained by solving the (10) into (9).

B. Controllers Design

This subsection presents the position and orientation control design for both UAVs. The position control is responsible for maintain the UAV over a desired position.

The position controller for both UAVs is designed and implemented of over an internal attitude control. Since this attitude inner loop is assumed to be significantly faster than the outer loop, the system's planar motion can be interpreted as a dominantly second-order system.

The UAVs are controlled by giving the following set of inputs:

- u_x front/back bending angle, assuming negative values bending forward and zero on the horizontal plane
- u_y left/right bending angle, assuming negative values bending leftward
- u_z vertical speed
- u_ψ angular speed around the yaw axis

In order to allow the choice between smooth and dynamic moves, this input arguments are not directly the control parameter values, but a percentage of the maximum corresponding values as set in the embedded attitude control parameters. Therefore, the input signals must be floating values within $[-1.0, 1.0]$. The internal attitude control loop is responsible for translate the set of inputs mentioned above to individual motor speeds.

In order to distinguish both UAVs, index subscripts $_R$ and $_F$ are used to denote the rear and front UAV, respectively.

1) Front UAV:

For the front UAV, the objective is the design of an autopilot external control loop by giving as input the desired inertial position and orientation values, denoted as $\mathbf{r}_F$, in order to maintain the UAV over the desired position. This autopilot is responsible for translate the input references into thrust forces to be generated by each rotor, resorting the inertial position, body-fixed frame velocities, Euler angles and rates, measurements based on sensors installed on board.

2) Rear UAV:

The control system for the rear UAV could also be a commercially available autopilot, however controllability would be compromised. Thus, for the rear UAV, the objective is the design of an external control loop by giving the desired relative position and orientation values between both UAVs in respect to the rear UAV body-fixed frame, in order to maintain the UAV over the desired relative position while orientated with respect to the desired relative orientation.

Since it is meant to control the rear UAV relative position and orientation with respect to its body-fixed frame, and assuming that both UAVs share the same dynamics, the dynamical system upon which it is meant to base the control design can be obtained resorting to (1), resulting in the following equations:

$$\begin{aligned} M\dot{\mathbf{v}}_{rel} &= \mathbf{f}_F^{\mathcal{I}} - \mathbf{f}_R^{\mathcal{I}} \\ \dot{\mathbf{p}}_{rel} &= \mathbf{v}_{rel} \\ \mathbf{z} &= \mathbf{C}\mathbf{p}_{rel} \end{aligned} \tag{11}$$

where $\mathbf{v}_{rel} = \mathbf{v}_F - \mathbf{v}_R$, $\mathbf{P}_{rel} = \mathbf{p}_F - \mathbf{p}_R$ and $\mathbf{C}$ is the output matrix. Since $\mathbf{f}_F^{\mathcal{I}}$ is the front UAV actuation force vector, $\mathbf{f}_F^{\mathcal{I}}$ is assumed as an external disturbance. The relative attitude angles are depicted as $\boldsymbol{\eta}_{rel}$.

An external control loop is designed taking into consideration that $\mathbf{p}_{rel}$, $\mathbf{v}_{rel}$ and $\boldsymbol{\eta}_{rel}$, can be measured resorting to the motion tracking system. The external control loop expects a reference relative position and orientation, and combines these references with the sensors measurements in order to obtain the desired roll, pitch, angular velocity around the yaw axis and vertical velocity, which are the attitude inner loop set of inputs. The inner attitude control loop is responsible for prescribing these inputs into individual rotors velocities. The control System architecture is depicted in Fig 1.

Since the relative position control design is based on the rear UAV body-fixed frame, the relative position tracking errors with respect to the NED frame are presented in the following equation:

$$\mathbf{e}_\mathbf{P} = \mathbf{r}_p - \mathbf{R}_{\mathcal{I}}^{\mathcal{B}}\mathbf{p}_{rel} \tag{12}$$

where $\mathbf{r}_p = (x_d, y_d, z_d)^T$ is the desired relative position reference vector between both UAVs in respect to the rear UAV body-fixed frame, and $\mathbf{R}_{\mathcal{I}}^{\mathcal{B}}$ is the rotation matrix that describes the rotation of the NED frame into the rear UAV body-fixed frame.

Therefore, the inner control loop input vector is computed as follows:

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_z \end{bmatrix} \mathbf{e}_\mathbf{P} - \mathbf{K}_{xyz} \begin{bmatrix} \mathbf{v}_{rel} \\ \boldsymbol{\eta}_{rel} \end{bmatrix} \tag{13}$$

where K_x, K_y and K_z are the position optimal gains, and $\mathbf{K}_{xyz}$ is the optimal gain matrix suppressing the gains regarding the position position states.

Resorting to relative position in respect to the NED frame, the relative yaw orientation from the rear to the front UAV can be computed as:

$$\psi_r = \psi_R + tan^{-1}\left(\tfrac{y_F - y_R}{x_F - x_R}\right) \tag{14}$$

Therefore, the orientation input control vector for the rear UAV is presented in (15).

$$u_\psi = K_\psi(r_\psi - \psi_r) \tag{15}$$

V. Estimation

In this Section the discrete linear Kalman filter is introduced, followed by the vertical velocity estimation.

A. Linear Kalman Filter

This Section presents the discrete Kalman filter derived using parameter optimization without making any gaussian assumptions, resorting to [7] for the continuous-time case. Consider the following discrete LTI dynamical system:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k + \mathbf{w}_k \\ \mathbf{z}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{16}$$

where the vector $\mathbf{x}_k$ is the $n \times 1$ state vector, the $m \times 1$ $\mathbf{u}_k$ is the control input and the $r \times 1$ $\mathbf{z}_k$ is the output vector. $\mathbf{w}_k$ is the zero-mean white Gaussian process noise vector that conveys the system error sources, and $\mathbf{v}_k$ is the zero-mean white Gaussian process noise vector that represents the measurement error sources. Both noise vectors are mutually independent sequences of zero mean white Gaussian noise, which covariance matrix is presented in (17). Here $\mathbf{Q}$ is a positive-definite (or positive-semidefinite) Hermitian or real symmetric matrix and $\mathbf{R}$ is a positive-definite Hermitian or real symmetric matrix.

$$E\left(\begin{bmatrix}\mathbf{w}_k \\ \mathbf{v}_k\end{bmatrix}\begin{bmatrix}\mathbf{w}_k^T & \mathbf{v}_k^T\end{bmatrix}\right) = \begin{bmatrix}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}\end{bmatrix} \tag{17}$$

1) Predict Phase:

The *a priori* state estimate and the *a priori* error covariance are computed in (18) and (19), respectively.

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}\hat{\mathbf{x}}_{k|k} + \mathbf{G}\mathbf{u}_k \tag{18}$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^T + \mathbf{Q} \tag{19}$$

2) Update Phase:

The Kalman gain matrix is computed in (20). The *a posteriori* estimation and the *a posteriori* estimate covariance are presented in (21) and (22).

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1} \tag{20}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \tag{21}$$

$$\mathbf{P}_{k|k} = (\mathbf{I}_n - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1} \tag{22}$$

B. Vertical Velocity Estimation

In order to estimate the vertical velocity, a complementary filter is designed regarding to the Kalman filter theory. For this purpose is considered the discrete filter's state-space representation presented in (23).

$$\mathbf{F} = \begin{bmatrix}1 & \Delta \\ 0 & 1\end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix}1 & 0\end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix}0 \\ 0\end{bmatrix} \tag{23}$$

VI. Simulation Results

In this section, the simulation results are presented and analyzed.

A. Simulation environment

In order to validate the feasibility of the control system designed a simulation environment was prepared. For this propose is used the dynamical models provided in [8]. The use of this dynamical models is justified by the commercially available quadrotor in use. The selected sample time for this environment is 0.065 seconds.

B. Results

The LQR gains regarding the X and Y position were calculated using a $\mathbf{Q} = \mathit{diag}(3, 1, 1, 0)$, and for Z position using a $\mathbf{Q} = \mathit{diag}(3, 1)$. For the front UAV, the weighting matrices $\mathbf{R}$ regarding XYZ are 270, 100, and 30, respectively. Analogously, the weighting matrices $\mathbf{R}$ regarding the rear UAV controller design are 270, 30, and 10. For the vertical velocity estimations, the Kalman filter gains were calculated using a $\mathbf{Q} = \mathit{diag}(0.01, 1)$ and $\mathbf{R} = 0.0011$

The simulation results regarding the consequent unit step responses are presented in Fig. 3. Here, the unit step responses for the XYZ directions regarding the front UAV present a settling time (5%) is at 7.9, 8.4 and 9.5 seconds, respectively, and it is duly noted an absence of overshoot. The relative position between both UAVs along the XYZ directions settles at 8.8, 7.2 and 5.7 seconds, respectively. The maximum relative position errors between both UAVs along the XYZ directions is 31, 20, 26 centimeters, respectively.

VII. Experimental Results

In this section, the analysis experimental results is presented alongside the implementation details.

A. Implementation

The implementation will be based on Parrot Ar.Drone 2.0. These quadrotors are controlled via WIFI in a range of 50 meters, having its own network defined as an access point. Parrot also provides a dedicated free app for the control of the quadrotor, live video streaming, making films and taking pictures.

In order to implement the control systems designed for the two UAVs, the AR Drone Simulink Development-Kit V1.1 (DevKit) provided in [8] is used. This Devkit provides simulink based models for the WIFI communication between both UAVs and a terminal, for instance, a personal computer. Here the control system is implemented in external mode. These simulink models allows sending combination of attitude (desired angles) and vertical speed commands as input control commands to the Ar.Drone embedded attitude inner control loop, and reading the states available upon the state reconstruction from the sensor data also built in the embedded electronics simultaneously. The state reconstruction provides estimations of the altitude, attitude angles, ground velocities.

For experimentally implement the control systems proposed, it is required to work in discrete time. The selected sample time for the controllers and filters meant to be implemented is 0.065 seconds, taking into consideration the slowest sensor built in the UAV, namely, the ultrasound which operates at a minimum sample time of 0.04 seconds. Therefore, the Qualisys system used as ground truth to the relative position

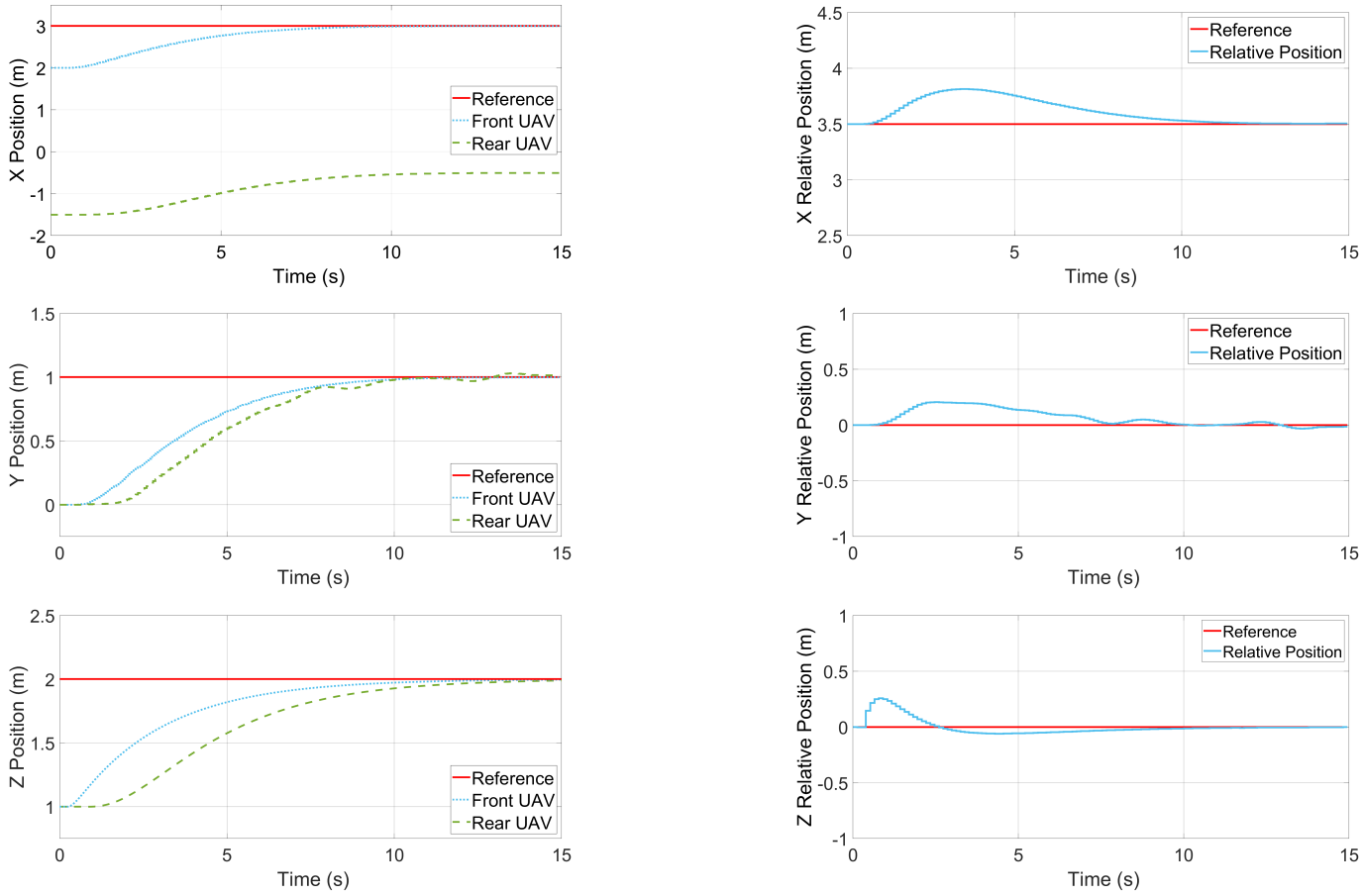

Fig. 3: Simulation results. From left to right, top to bottom: (a) X Position, (B) X Relative Position, (C) Y Position, (D) Y Relative Position, (E) Z Position, (F) Z Relative Position

and orientation is set to work also at a sample time of 0.065 seconds.

In order to reduce the ultrasound signal interferences between both UAVs, resulting in conflicts on the altitude estimation of both subsystems, one of the UAVs was set to operate at a different signal frequency. The reference distance along the longitudinal direction for the relative position was set to 3.5 meters.

The rear UAV, which is responsible for regulating the relative position, the controller was designed aiming a faster response than the front UAV, in order to compensate more efficiently the delay present in the system.

B. Results

The LQR controller gains for the front and rear UAV, were calculated using the weighting matrices presented in table I and II, and are presented in table III and IV. The Kalman filter regarding the vertical velocity estimation were designed using a $\mathbf{Q} = diag(0.01, 1)$ and $\mathbf{R} = 0.0011$.

In Figure 4 are presented the experimental results regarding the consequent unit step responses. Here, the unit step responses for the XYZ directions regarding the front UAV present a settling time (5%) is at 3, 5.6 and 8.9 seconds, respectively. The maximum overshoot is 2.2, 3.6 and 0 centimeters for the XYZ directions. The relative position between both UAVs along the XYZ directions settles at 6, 6 and 8 seconds, respectively. The maximum relative position errors between both UAVs along the XYZ directions is 35, 22, 43 centimeters, respectively. It is possible to infer that the results were quite satisfactory.

A video showing the cooperative load transportation reported in this paper can be found in (https://youtu.be/QlUf5pb1f1w).

VIII. Conclusion

The control system proposed provided good results, where the maximum relative error relied under 43 centimeters for the XYZ directions. However, the control system performance is evaluated through unit steps, in order to study the control system in a more aggressive way. The maximum relative error decreases for references of reduced frequency.

TABLE I: Front UAV LQR weighting matrices

	X Position	Y Position	Z Position
Q	*diag*(3, 10, 1, 0)	*diag*(3, 1, 1, 0)	*diag*(3, 1)
R	150	220	25

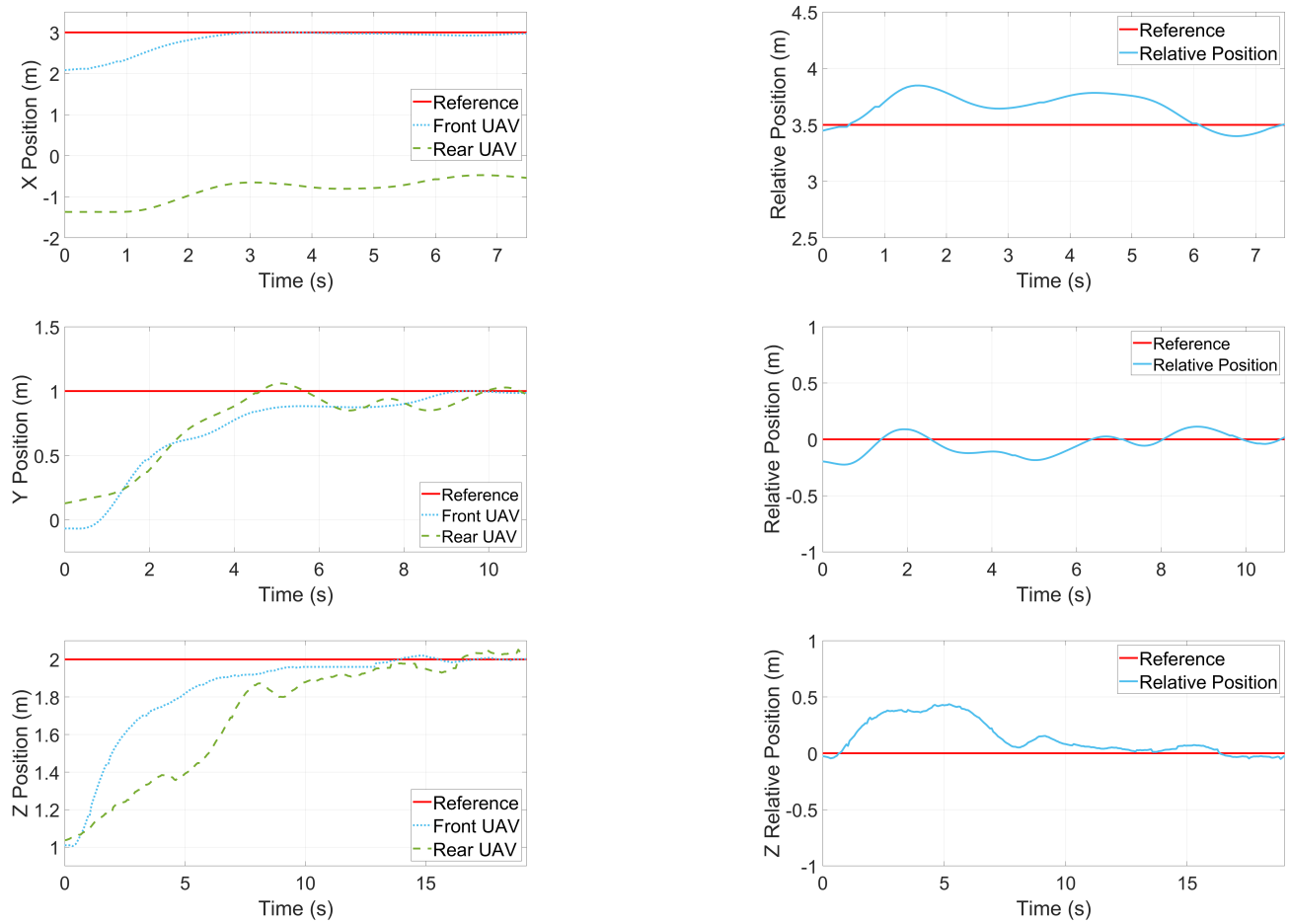

Fig. 4: Experimental results. From left to right, top to bottom: (a) X Position, (B) X Relative Position, (C) Y Position, (D) Y Relative Position, (E) Z Position, (F) Z Relative Position

TABLE II: Rear UAV LQR weighting matrices

	X Position	Y Position	Z Position
Q	*diag*(6, 15, 2, 0)	*diag*(3, 1, 1, 0)	*diag*(3, 1)
R	200	180	20

TABLE III: Front UAV LQR Gains

	Gains			
X Position	$[-0.1383$	-0.2298	0.4781	$0.1001]$
Y Position	$[0.1156$	0.2323	0.0794	$0.0156]$
Z Position		$[0.3425$	$0.0688]$	

TABLE IV: Front UAV LQR Gains

	Gains			
X Position	$[0.1688$	0.2605	-0.5394	$-0.1116]$
Y Position	$[-0.1277$	-0.2526	-0.0870	$-0.0169]$
Z Position		$[-0.3823$	$-0.0781]$	

ACKNOWLEDGMENT

This work was supported by FCT, through IDMEC, under LAETA, project UID/EMS/50022/2019

REFERENCES

[1] R. M. Murray, "Trajectory generation for a towed cable system using differential flatness," in *IFAC world congress*, 1996, pp. 395–400.

[2] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2216–2222.

[3] P. Outeiro, C. Cardeira, and P. Oliveira, "Adaptive/multi-model height control system of a quadrotor constant unknown load transportation," in *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, April 2018, pp. 65–70.

[4] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 324–334, 2011.

[5] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," *rn*, vol. 1, no. r2, p. r3, 2013.

[6] K. Ogata, *Modern Control Engineering*, ser. Instrumentation and controls series. Prentice Hall, 2010.

[7] M. Athans, "Derivation of the kalman-bucy filter using parameter optimization," MIT ISR/IST, Tech. Rep. KF #8A, 2001.

[8] D. Sanabria and P. Mosterman, "Ar drone simulink development-kit v1.1," November 2014, https://www.mathworks.com/matlabcentral/fileexchange/43719.