Cooperative Aerial Payload Transportation Using Two Quadrotors

A. Rajaeizadeh[*], A. Naghash, and A. Mohamadifard
Amirkabir University of Technology (Tehran Polytechnic), Faculty of Aerospace Engineering, 424 Hafez Ave., Tehran
15875-4413, Iran

Abstract

In this paper, we consider the problem of controlling multiple quadrotors fastened to a payload and cooperatively transport it in 3 dimensions. We model the quadrotors as a group rigidly attached to a payload. Then we develop the equations of motion of this rigid system. We propose a rigid-body formation system controller based on LQR method as well as a Paparazzi-based guidance scheme for the payload mission trajectory. Also a PD controller is developed and its results are compared with the main controller's results. A simulation study with two quadrotors cooperatively stabilizing, and transporting a payload along two different desired three-dimensional trajectories is presented.

1 Introduction

Safe co-operative transportation of possibly large and bulky payloads is extremely important in various missions such as military operations, Search and Rescue and personal assistant. With recent advancements in relevant technologies and commercially available micro aerial vehicles (MAVs), the problem of autonomous grasping, manipulation, and transportation is advancing to the aerial domain in both theory and experiments. The problem is addressed and formulated either in a way that the payload is connected to the aerial vehicles via flexible cables or gripped to the agents in multiple locations, forming a rigid body in total, the latter of which is the subject of the current job. However these results are based on the common and restricting assumption that the Dynamics of the cable and payloads are ignored and they are considered as a bounded disturbance in the transforming vehicle. Therefore it is challenging to incorporate the effects of a payload fixed to a group of quadrotors which prompts a distributed control policy for each agent while the whole system exhibits a rigid body dynamics. We approach the problem by first developing a model for a single quadrotor and a team of quadrotors rigidly attached to a payload 'Sec. 3'. In 'Sec. 4', we propose a LQR control law for an individual quadrotor. Guidance scheme which consists of a reference generator and a PD controller is issued in 'Sec. 5'. A MATLAB-based simulation is conducted for this problem with figures and discussions found in 'Sec. 6'. The final edition of this paper may contain results from an experimental study which is to be performed in the next step.

2 Literature Review

Modeling and control of a quadrotor with a payload which is connected via a flexible cable is addressed in [1]. The work is then extended for multiple quadrotors each carrying a cable which is attached to a single payload [2]. The system modeling is based on Lagrangian mechanics and the flexible cables are modeled as systems of serially-connected links and has been considered in the full dynamic model. We address a different problem as the robots have grasped the payload via rigid connections at multiple locations. The modeling of contact constraints is considerably simpler as issues of form or force closure are not relevant. Additionally, contact conditions do not change in our case. However, the system is statically indeterminate and the coordination of multiple robots is significantly more complex than in the case when the payload is suspended from aerial robots. In particular, as the problem is over-constrained the robots must control to move in directions that are consistent with kinematic constraints. The problem of aerial manipulation using rigid cables (rods and ball joints) is analyzed in [3] with the focus on finding robot configurations that ensure static equilibrium of the payload at a desired pose while respecting constraints on the tension. In the sense that we have access to many rotors to generate the thrust necessary to manipulate payloads, our work is similar to [4], where the authors propose control laws that drive a distributed flight array consisting of many rotors along a desired trajectory. In [5] quadrotors are attached to a flexible net. The fleet is capable of throwing and catching balls with the net. Based on a first-principles model of the net forces, algorithms that generate open-loop trajectories for throwing and catching a ball are introduced. A swarm of quadrotors termed as a flying hand is proposed in [6] which is able to grasp an object where each UAV contributes to the grasping task with a single contact point at the tool-tip and is tele-operated by a human hand whose fingertip motions are tracked. A classification on how the object is carried or grasped is offered in [7] with an elaboration on Aerial Grasp and Manipulation proposing virtual linkage as a new paradigm. According to the offered terminology, our work is termed Aerial Transportation, where

[*] Email address(es): ahmad_rajaiizade@yahoo.com

a static aerial object is grabbed by magnetic or gripper-type mechanism. Then, the object is rigidly attached to the mainframe of the UAV. When two or more UAVs are involved, a new UAV arises because the two original mainframes are now rigidly connected to the object. A novel aerial manipulation system is proposed in [8], where a mechanical structure enjoys a number of thrusters and their geometry will be derived from technical optimization problems. The aforementioned problems are defined by taking into consideration the desired actuation forces and torques applied to the end-effector of the system. But the most inspiring work in which the main results of the current research are mostly utilized from is the one proposed in [9] that makes use of a PD controller to stabilize the system. We employed the main results of that work to formulate our control input (force and moments) distribution algorithm for each quadrotor with slight modifications in case of non-symmetrical payload mass distribution, while differing significantly in terms of autopilot controller and guidance scheme.

3 System Modeling

3.1 *Generalities*

In modeling the rigid-body system following assumptions are made :

- The system is made of two quadrotors that are rigidly attached to a beam shaped payload.
- The center of mass of the whole system coincides the origin of the rigid-body coordinate system.
- Quadrotors and the payload are rigid.
- Propellers are assumed rigid.
- Thrust and drag forces are proportional to the square of propellers speed.
- The system is symmetric in its X axis.

3.2 *System Configuration and Coordinate*

The NED coordinate system is taken as default coordinate system and it's shown in 'Fig. 1'. The World Frame axes are X, Y and Z pointing downward. We presume the body frame axes are the primary axis of the whole system. Agents have their own coordinate frames and their relative yaw angles which are assumed zero by default.
To derive the rotation matrix from body to the Inertia frame we choose zyx Euler angles rotation. Final matrix is shown as follows:

$$R(\phi,\theta,\psi) = R(x,\phi)R(y,\theta)R(z,\psi) \tag{1}$$

Which will be:

$$R_{IB} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & \psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2}$$

where $c = cos$ and $s = sin$.

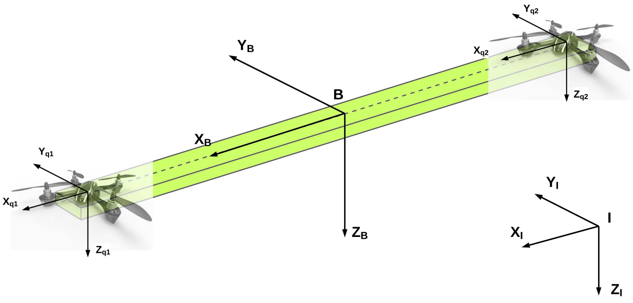

Figure 1: Coordinate Systems.

3.3 *Physical Specifications*

As the system is composed of 3 separate objects with their own specific physical characteristics, we need to calculate system's overall specifications including mass and moments of inertia. Considering that the agents are identical quadrotors with known physical and geometrical specifications the payload is a rectangular-section beam with a mass of m_p, width of w, height of h and length of L, following data are calculated as:

- Payload's moments of inertia:

$$\begin{aligned} I_{X_p} &= \frac{1}{12}m_p(h^2+w^2) \\ I_{Y_p} &= \frac{1}{12}m_p(L^2+h^2) \\ I_{Z_p} &= \frac{1}{12}m_p(L^2+w^2) \end{aligned} \tag{3}$$

- Position of total center of mass:

$$\begin{aligned} \bar{X} &= \frac{\sum_{i=1}^{3} m_i x_i}{\sum_{i=1}^{3} m_i} \\ \bar{Y} &= \frac{\sum_{i=1}^{3} m_i y_i}{\sum_{i=1}^{3} m_i} \\ \bar{Z} &= \frac{\sum_{i=1}^{3} m_i z_i}{\sum_{i=1}^{3} m_i} \end{aligned} \tag{4}$$

- System's mass and moments of inertia:

$$\begin{aligned} M &= m_1 + m_2 + m_p \\ I_X &= \sum_{i=1}^{3}(I_{x_i} + m(\bar{y_i}^2 + \bar{z_i}^2)) \\ I_Y &= \sum_{i=1}^{3}(I_{y_i} + m(\bar{x_i}^2 + \bar{z_i}^2)) \\ I_Z &= \sum_{i=1}^{3}(I_{z_i} + m(\bar{y_i}^2 + \bar{x_i}^2)) \end{aligned} \tag{5}$$

In which $\bar{x_i} = x_i - \bar{X}$, $\bar{y_i} = y_i - \bar{Y}$ and $\bar{z_i} = z_i - \bar{Z}$.

3.4 Equations of Motion

3.4.1 Single Quad:

First we derive the equations for a single quadrotor. Then these equations will be developed for a multi agent system.

- Rotational Kinematics:

Since the angular rates are related to the body, they should be transformed to Inertia World Frame. Using Euler angles we have:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R_r \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{6}$$

$$R_r = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \tag{7}$$

- Rotational Dynamics:

Using the rotation matrix and Coriolis Effect, rotational dynamic equation takes the form of:

$$\begin{bmatrix} I_{x,j} & 0 & 0 \\ 0 & I_{y,j} & 0 \\ 0 & 0 & I_{z,j} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{x,j} & 0 & 0 \\ 0 & I_{y,j} & 0 \\ 0 & 0 & I_{z,j} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$= M_{q,j} + M_{Gq,j} \tag{8}$$

where:

$$M_{q,j} = \begin{bmatrix} lb(\Omega_{j,4}^2 - \Omega_{j,2}^2) \\ lb(\Omega_{j,3}^2 - \Omega_{j,1}^2) \\ d(-\Omega_{j,1}^2 + \Omega_{j,2}^2 - \Omega_{j,3}^2 + \Omega_{j,4}^2) \end{bmatrix} \tag{9}$$

$$M_{Gq,j} = J_r \Omega_j \begin{bmatrix} \dot{\theta} \\ -\dot{\phi} \\ 0 \end{bmatrix} \tag{10}$$

$$\Omega_j = (\Omega_{j,1} - \Omega_{j,2} + \Omega_{j,3} - \Omega_{j,4}) \tag{11}$$

- Translational Dynamics:

Applying the rotation matrix we have:

$$m_j \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ m_j g \end{bmatrix} + R_{IB} F_{q,j} \tag{12}$$

$$F_{q,j} = \begin{bmatrix} 0 \\ 0 \\ -b(\Omega_{j,1}^2 + \Omega_{j,2}^2 + \Omega_{j,3}^2 + \Omega_{j,4}^2) \end{bmatrix} \tag{13}$$

3.4.2 The Rigid-body System:

As the whole system is supposed rigid its dynamic model is simpler than a system with cable-suspended load. Considering each agent produces forces and moments in its own frame, we need to develop a relationship between the behavior of the system and the agents. Depending on the configuration, the following formulation was developed in which j, x_j, y_j and ψ_j are the number, location and relative heading of each agent in rigid-body coordinate system.

$$\begin{bmatrix} F_B \\ M_B \end{bmatrix} = \sum_{j=1}^{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ y_j & cos\psi_j & -sin\psi_j & 0 \\ -x_j & sin\psi_j & cos\psi_j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_{q,j} \\ M_{q,j} \end{bmatrix} \tag{14}$$

Applying the new formulation for the systems force and moments, dynamic of the system will be achieved as follows:

$$\begin{aligned} \ddot{X} &= (sin\phi sin\psi + cos\phi sin\theta cos\psi)\frac{-U_1}{M} \\ \ddot{Y} &= (-sin\phi cos\psi + cos\phi sin\theta sin\psi)\frac{-U_1}{M} \\ \ddot{Z} &= g + (cos\phi cos\theta)\frac{-U_1}{M} \end{aligned} \tag{15}$$

$$\begin{aligned} \ddot{\phi} &= \frac{I_Y - I_Z}{I_X}\dot{\theta}\dot{\psi} - \frac{J_r \Omega \dot{\theta}}{I_X} + \frac{U_2}{I_X} \\ \ddot{\theta} &= \frac{I_Z - I_X}{I_Y}\dot{\phi}\dot{\psi} + \frac{J_r \Omega \dot{\phi}}{I_Y} + \frac{U_3}{I_Y} \\ \ddot{\psi} &= \frac{I_X - I_Y}{I_Z}\dot{\theta}\dot{\phi} + \frac{U_4}{I_Z} \end{aligned} \tag{16}$$

4 Controller Design

4.1 Cooperative Control Law:

As the Agents are rigidly attached to the payload, design of the controller will become a challenge. The first step is to assume the rigid-body system as a single quadrotor with specification of the system. Then we calculate the necessary control commands and try to control this quadrotor. For controlling the attitude of the system two techniques were used, PD and $Tracking\ LQR$. In order to distribute controller commands between agents a comprehensive cooperative control law is developed. Defining [9]:

$$u_F = \frac{1}{M}\left[(m_1 + m_{p1}), 0, 0, 0, (m_2 + m_{p2}), 0, 0, 0\right]^T$$

$$u_{M_x} = \frac{1}{\frac{\omega_{Mxy}}{\omega_F}\sum_{j=1}^{2} y_j^2 + 2} \begin{bmatrix} \frac{\omega_{Mxy}}{\omega_F} y_1 \\ c\psi_1 \\ s\psi_1 \\ 0 \\ \frac{\omega_{Mxy}}{\omega_F} y_2 \\ c\psi_2 \\ s\psi_2 \\ 0 \end{bmatrix} \tag{17}$$

$$u_{M_y} = \frac{1}{\frac{\omega_{Mxy}}{\omega_F}\sum_{j=1}^{2} x_j^2 + 2}\begin{bmatrix} -\frac{\omega_{Mxy}}{\omega_F}x_1 \\ -s\psi_1 \\ c\psi_1 \\ 0 \\ -\frac{\omega_{Mxy}}{\omega_F}x_2 \\ -s\psi_2 \\ c\psi_2 \\ 0 \end{bmatrix}$$

$$u_{M_z} = \frac{1}{2}\left[0,0,0,1,0,0,0,1\right]^T$$

where $m_{pj} = m_p(1 - \left|\frac{x_j}{\bar{x_1} - \bar{x_2}}\right|)$, is the share of payload mass applied on each agent that rises when the asymmetric distribution of payload's CG is applied. $\frac{\omega_{Mxy}}{\omega_F}$ indicates the relationship between the moment and the force generated by the agents to produce a system moment. The bigger amount of $\frac{\omega_{Mxy}}{\omega_F}$ means the effect of agents forces is bigger than the effect of their moments in creating systems moment. x and y are the locations of the CG of each agent in body frame. Final control commands applying to each agent will be:

$$U = \left[u_F, u_{M_x}, u_{M_y}, u_{M_z}\right]\left[U_F, U_{M_x}, U_{M_y}, U_{M_z}\right]^T \quad (18)$$

This U is an 8×1 matrix which its first half elements are allocated to the first agent and the second half are allocated to the second agent.

4.2 Control Techniques

In order to control the attitude and altitude of the system we made use of two classic control laws which are PD and LQR methods. The control block of the system is described as:

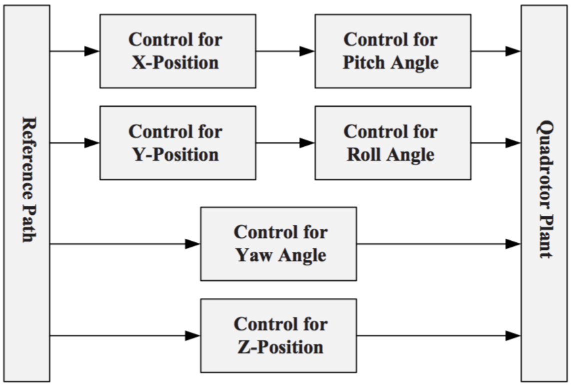

Figure 2: Control Logic

Reference path and controllers of X and Y positions will be described. Since the control laws are linear, the equations of motion of the system need to be linearized. Final linear equations are as follows :

$$\ddot{X} = \frac{-U_1\theta}{M}$$

$$\ddot{Y} = \frac{U_1\phi}{M} \quad (19)$$

$$\ddot{Z} = g - \frac{U_1}{M}$$

$$\ddot{\phi} = \frac{U_2}{I_X}$$

$$\ddot{\theta} = \frac{U_3}{I_Y} \quad (20)$$

$$\ddot{\psi} = \frac{U_4}{I_Z}$$

The control command inputs to each agent are as follows:

$$U_{j,F} = b(\Omega_{j,1}^2 + \Omega_{j,2}^2 + \Omega_{j,3}^2 + \Omega_{j,4}^2)$$

$$U_{j,M_x} = lb(\Omega_{j,4}^2 - \Omega_{j,2}^2) \quad (21)$$

$$U_{j,M_y} = lb(\Omega_{j,3}^2 - \Omega_{j,1}^2)$$

$$U_{j,M_z} = d(-\Omega_{j,1}^2 + \Omega_{j,2}^2 - \Omega_{j,3}^2 + \Omega_{j,4}^2)$$

4.2.1 PD Controller Design:

PD method is carried out for comparison with LQR method to see differences in performance of these two methods. The control command calculated by PD method is obtained as follows:

$$U = K_{(x)}(e_x) + K_{d(\dot{x})}(e_{(\dot{x})}) \quad (22)$$

Attitude control commands take the form of:

$$U_{2,3,4} = K_{p(\phi,\theta,\psi)}(e_{(\phi,\theta,\psi)}) + K_{d(\dot{\phi},\dot{\theta},\dot{\psi})}(e_{(\dot{\phi},\dot{\theta},\dot{\psi})}) \quad (23)$$

Altitude control output calculation is as explained above:

$$U_1 = K_{(z)}(e_z) + K_{d(\dot{z})}(e_{(\dot{z})}) \quad (24)$$

The PD gains are determined through Ziegler-Nichols method. The scheme of this controller is as shown in 'Fig. 3':

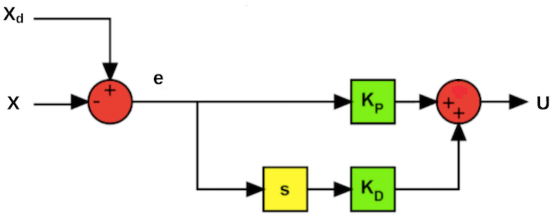

Figure 3: PD Controller Diagram

4.2.2 LQR Controller Design:

- LQR

Attitude Control:
First, we define the attitude state vectors as below:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} \tag{25}$$

Then the attitude system can be written in state space form as:

$$\dot{X} = AX + BU \tag{26}$$

Which will be:

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{I_X} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{I_Y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_Z} \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{27}$$

The goal is to find the stabilizing feedback control law:

$$U = -KX \tag{28}$$

Which minimizes the cost function:

$$J = \int_0^\infty (X^T Q\, X + U^T R\, U) dt \tag{29}$$

where $Q \geq 0$ and $R > 0$ are weighting matrices of a appropriate dimensions. Q is related to the energy of the controlled output, while R is related to the energy of the control signal. So the choice of Q and R is a trade-off between the desired performance and the available capacities. Here we decided Q to be an identity matrix of appropriate dimension and R will be accomplished by a trial-and-error process until the **best** answer is obtained. Afterward, K is defined as:

$$K = R^{-1} B^T P \tag{30}$$

Where P is the solution of linear algebraic Riccati equation:

$$A^T P + PA - PBR^{-1} B^T P + Q = 0 \tag{31}$$

After calculating K controlling of the states will be accomplished.

Altitude Control:
For altitude the state vector is written as:

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} z \\ \dot{z} \end{bmatrix} \tag{32}$$

Then the state space form of the system is as:

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{M} \end{bmatrix} U_1 \tag{33}$$

All discussed about attitude control is applicable here too.

- Tracking LQR

Since the desired states of the system are not supposed to be always zero and the LQR control law is only applicable to the linear time-invariant systems, we needed to make use of $Tracking\ LQR$ controller. All discussed about LQR is applicable to $TLQR$ too but the control law will be different as follows [10]:

$$U = -K(X - X_d) \tag{34}$$

X_d is the desired state and can be either time-variant or time-invariant none-zero. Defining:

$$X_{d\ _{att}} = \left[\phi_d, \dot{\phi}_d, \theta_d, \dot{\theta}_d, \psi_d, \dot{\psi}_d\right]^T \ ,\ X_{d\ _{alt}} = \left[z_d, \dot{z}_d\right]^T$$

Calculation of desired rates will be discussed in 'Sec. 5'. For attitude and altitude control, after applying the iterative trial-and-error design procedure the following form for the Q and R matrices is reached:

Attitude:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{35}$$

$$R = \begin{bmatrix} 1 \times 10^{-3} & 0 & 0 \\ 0 & 1 \times 10^{-4} & 0 \\ 0 & 0 & 1 \times 10^{-3} \end{bmatrix} \tag{36}$$

Altitude:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{37}$$

$$R = 6 \times 10^{-5} \tag{38}$$

5 Trajectory Generation

In addition to set point trajectory, a circular path was implemented and the stability and the performance of the system were analyzed. For both trajectories we made use of a PD controller which transforms the desired position to desired angle. This process is carried out by using Paparazzi algorithm which consists of a reference generator and a PD controller and an attitude generator in order to convert the desired position to desired angle and desired angle to desired states which are the inputs of the control system. These algorithms are available at Paparazzi's website. In both paths its assumed that $\psi(t) = \psi(0) = 0$.

5.1 Paparazzi Set Point Reference Generator

The generator receives the desired position in X or Y direction as input signal and converts it to 3 output signals which are position, velocity and acceleration related to the desired point respectively. The scheme of the set point R.G. is shown below:

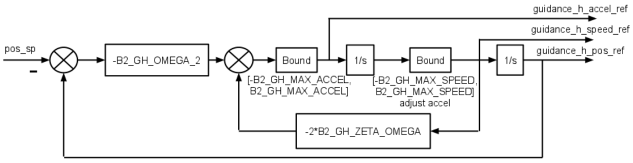

Figure 4: Set Point Reference Generator

5.2 Reference Controller and States Reference Generator

This module takes signals generated by 'Subsec. 5.1' as inputs and produces the desired attitude using a PD controller as follows:

$$(\theta_d) = K_p(X - X_{ref}) + K_d(\dot{X} - \dot{X}_{ref}) \tag{39}$$

$$(\phi_d) = K_p(Y - Y_{ref}) + K_d(\dot{Y} - \dot{Y}_{ref}) \tag{40}$$

As the control laws developed in 'Sec. 4' need desired states and considering that reference PD controller only generates the angle and not its rate, and noting that using the perfect derivation of this angle is not a scientific solution, we need to calculate the real attitude rate. This calculation is carried out by using Paparazzi attitude R.G. This module does the same as 'Subsec. 5.1' but the input is an angle not a position. This module is shown as:

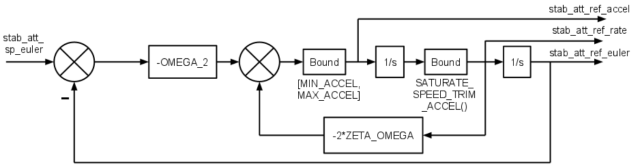

Figure 5: Attitude Reference Generator

It should be mentioned that 'Subsec. 5.1' is used in altitude controller too.

6 Results

In this section the behavior of the system's states in tracking two different trajectories will be analyzed and the performance of the controllers will be compared.

6.1 Set Point Trajectory

To make the simulation similar to what IMAV 2017 outdoor cooperative mission requires, the value of 50 meters is set as desired position. In order to see system's behavior in both lateral and longitudinal axes, the desired position is set in both X and Y directions and the results are as follows (fig.: 6-9):

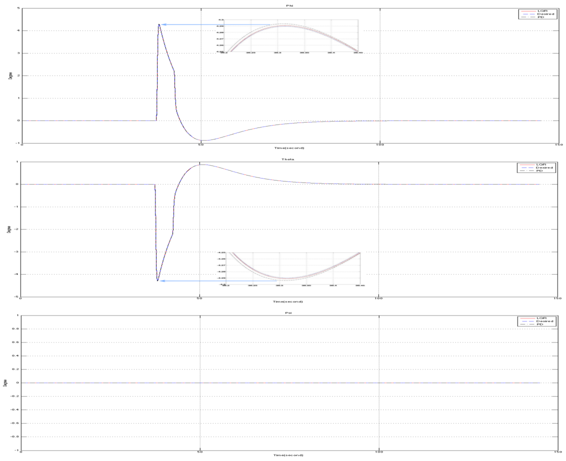

Figure 6: Euler Angles

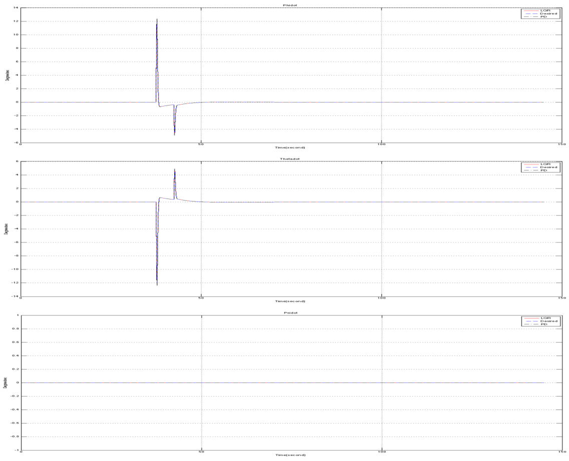

Figure 7: Euler Rates

6.2 Circular Trajectory

A circular path with a radius of 50 meters in an altitude of 3 meters is designed and is given to the system as the desired trajectory. The results are as follows (fig.: 10-13):

7 Conclusion and Future Works

Due to asymmetry in the geometry of the rigid-body system, for maneuvers in the asymmetry axis more control effort is needed to achieve the desired state and the stabilization in this axis is more challenging.

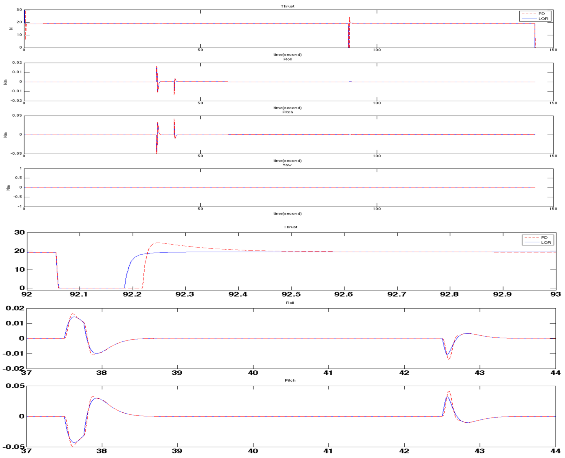

Figure 8: Control Inputs $(U_F, U_{M_{x,y,z}})$

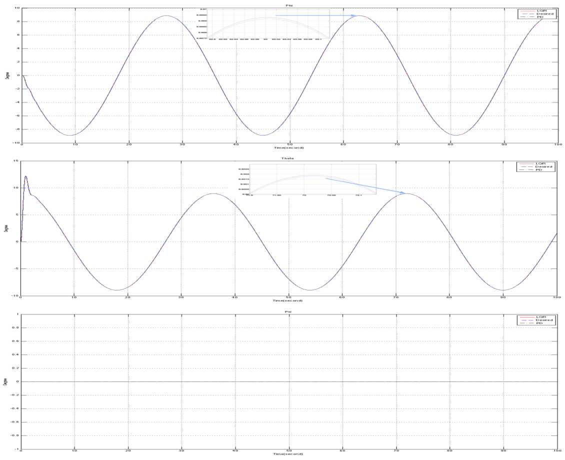

Figure 10: Euler Angles

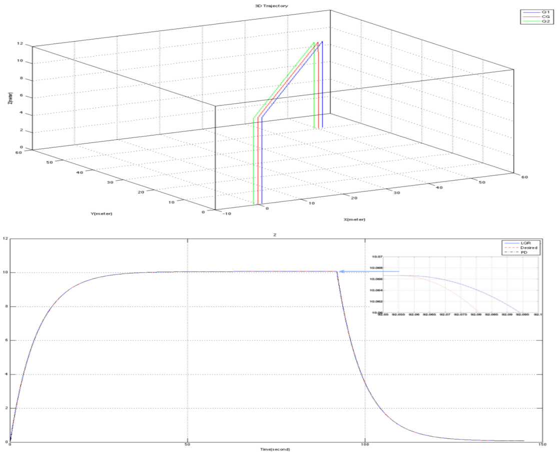

Figure 9: 3D Trajectory

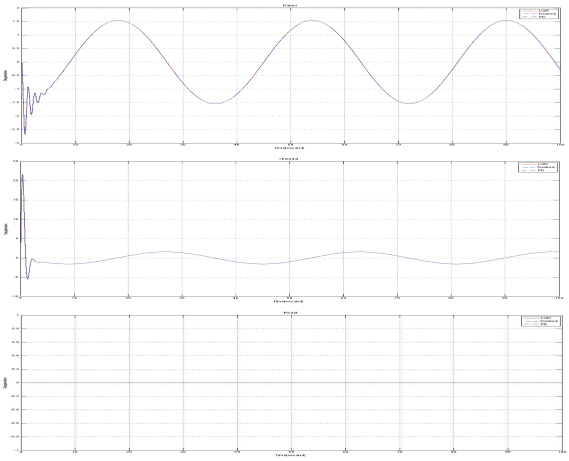

Figure 11: Euler Rates

We defined the subject of transporting a payload by 2 quadrotors rigidly attached to it and made use of two control techniques to reach the desired states of the system. Analyzing the achieved results in defined scenarios we can see that the behaviors of these two controllers are close to each other. We are currently planning to implement this simulation and carry out the experimental phase of this article using Paparazzi autopilot. Experimental results may be available in the final edition of this paper. Also a visualized simulation of this job is in progress.

Acknowledgements

The authors would like to thank Hamidreza Nemati for fruitful discussions and professional advises on $Traking\ LQR$ concept.

References

[1] F.A. Goodarzi, D. Lee, and T. Lee. Geometric control of a quadrotor uav transporting a payload connected via flexible cable. *International Journal of Control, Automation and Systems*, 13(6):1486–1498, 2015.

[2] F.A. Goodarzii and T. Lee. Dynamics and control of quadrotor uavs transporting a rigid body connected via flexible cables. In *IEEE American Control Conference (ACC)*, 2015.

[3] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, 2011.

[4] R. Oung and R. D'Andrea. The distributed flight array. *Mechatronics*, 21(6):908–917, 2011.

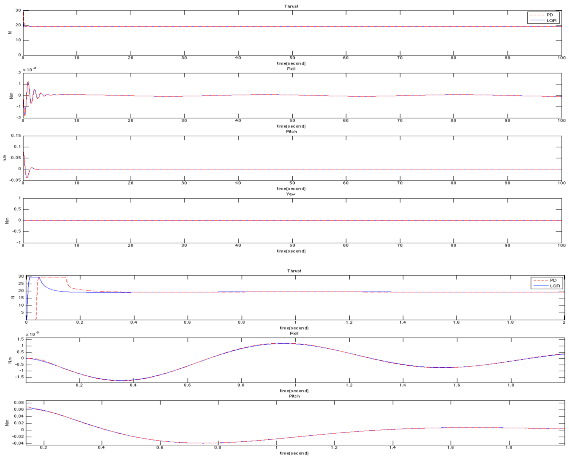

Figure 12: Control Inputs $(U_F, U_{M_{x,y,z}})$

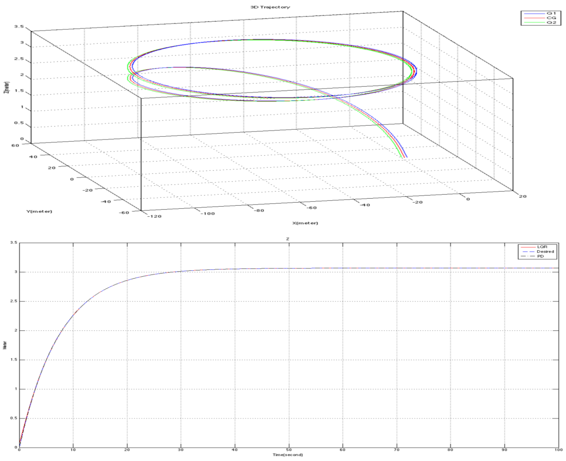

Figure 13: 3D Trajectory

[5] R. Ritz, M.W. Mller, M. Hehn, and R. DAndrea. Cooperative quadrocopter ball throwing and catching. In *IEEE Intelligent Robots and Systems (IROS)*, 2012.

[6] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo. The flying hand: A formation of uavs for cooperative aerial tele-manipulation. In *IEEE Robotics and Automation (ICRA)*, 2014.

[7] V. Parra-Vega, A. Sanchez, C. Izaguirre, O. Garcia, and F. Ruiz-Sanchez. Toward aerial grasping and manipulation with multiple uavs. *Journal of Intelligent and Robotic Systems*, pages 1–19, 2013.

[8] A. Nikou, G.C. Gavridis, K.J. Kyriakopoulos, O. Garcia, , and F. Ruiz-Sanchez. Mechanical design, modelling and control of a novel aerial manipulator. In *IEEE Conference Robotics and Automation (ICRA)*, 2015.

[9] D. Mellinger, M. Shomin, N. Michael, and V. Kumar. *Cooperative grasping and transport using multiple quadrotors*. Springer Distributed autonomous robotic systems, Berlin, Heidelberg, 2013.

[10] V.G. Adir and A.M. Stocia. Integral lqr control of a star-shaped octorotor. *Incas Bulletin*, 4(2):3–18, 2012.