



CENTRO DE INVESTIGACIONES
EN OPTICA, A.C.

“TELEOPERATION AND CONTROL OF AN AERIAL MANIPULATOR USING A VIRTUAL REALITY SYSTEM”



Tesis que para obtener el grado de Maestro en Optomecatrónica

Presenta: Carlos Arturo Rivera Quezada

Director de Tesis: Dr. Gerardo Ramón Flores Colunga

León · Guanajuato · México

Junio de 2022

Contents

Acknowledgements	7
Publications	8
Abstract	9
Introduction	11
1 State of the Art	13
2 Theoretical Knowledge	18
2.1 Mathematical Modelling of a Quadrotor	18
2.1.1 Position	18
2.1.2 Attitude	20
2.1.3 Quadrotor modelling equations	21
2.2 Lyapunov Stability: Direct Method	21
2.3 Observers	25
2.4 Robot Forward Kinematics	26
2.4.1 Denavit-Hartenberg	26

2.5	Dynamics	28
2.5.1	Newton-Euler Formulation	28
2.6	PID controller	30
2.7	Exponential Coordinates of Rotations	31
3	Methodology	33
3.1	The control problem	33
3.2	Aerial Vehicle	34
3.2.1	Modelling	34
3.3	Position and Angular Velocity Control equations	34
3.4	Force and Torque Observer	35
3.4.1	Force and Torque Observer equations	35
3.5	Stability demonstration	36
3.6	Manipulator	41
3.6.1	Robot Kinematics	41
3.6.2	Robot Dynamics	43
3.6.3	Multi-variable Joint Control	46
3.6.4	Manipulator PD control stability	47
4	Experiments and results	49
4.1	Position and Manipulator Controller	49
4.2	Attitude	54
4.3	Drone simulation	57
5	Conclusions	59
	Bibliography	60
6	Annex	66

List of Figures

2.1	The position and attitude variables of the quadcopter. Also, the force that each rotor produces.	19
2.2	Angular velocity representation of a coordinate system.	21
2.3	An interpretation of the stability criteria inspired on the geometrical representation. Figure <i>a</i>) shows two regions of stability, region A shows stability and region B shows asymptotic stability. Figure <i>b</i>) is a physical interpretation of this two regions considering the state x_0 as a ball that will not move inside region A because there's any force acting on it. However if the ball is inside region B the gravity force will push the ball to the equilibrium point x_e	23
2.4	Geometrical three dimensional representation of the Lyapunov quadratic function.	24
2.5	Coordinate frames of a 3DOF manipulator.	27
2.6	Graphical representation of the Denavit-Hartenber parameters.	28
2.7	PID controller block diagram representation.	30
2.8	Vector $p(0)$ with a θ rotation over the \hat{w} axis.	31
3.1	Aerial manipulator CAD model.	34

3.2	Control block diagram for the position and attitude controller.	35
3.3	Initial position of the 2DOF manipulator.	42
3.4	Torque and force representations in each joint of the manipulator.	44
3.5	Control diagram for the manipulator PD controller	47
4.1	a) position and desired position over time. b) linear velocity value and the estimated linear velocity.	51
4.2	Force estimation with a load variation at $t = 10s$	51
4.3	Manipulator controller of each joint with a desired joint angle q_d versus the computed q	52
4.4	a) Torque estimation and b) Ω state estimation.	53
4.5	Angular velocity value versus the desired one.	53
4.6	a) actual position in the x,y coordinates and the desired and the actual z_d, z position. b) actual linear velocity and the estimated. The position and veloc- ity are caused by the rotation of the vehicle.	54
4.7	Force estimation	55
4.8	Torque and Ω estimation.	56
4.9	Actual angular velocity versus the desired one.	56
4.10	Trajectory made by the Aerial Manipulator during the simulation time $0 \leq$ $t \leq 10s$	57
4.11	All the rotations of the vehicle during the simulation time $0 \leq t \leq 10s$	58
6.1	a) the operator using the HTC vive device during the simulation. The above figure in column b) shows the Gazebo simulation and the second image shows the aerial vehicle in Unity 3D.	67
6.2	Communication system structure of the virtual reality environment.	68
6.3	Physical aerial manipulator belonging to the Laboratory of Perception and Robotics.	68

List of Tables

3.1 Denavit-Hartenberg parameters.	42
4.1 Simulation parameters.	50

Acknowledgements

I want to thank all my family for supporting me through these years. I especially want to thank my parents, whose advice helped me on this road. I also want to thank all my teachers that guided me with their knowledge. I think every single one of them put a little piece of knowledge that brought me to this moment right now. Finally, I want to thank all my schoolmates and friends. I also want to thank my supervisor Gerardo Flores who tried to guide us when we seemed to be lost.

Nevertheless, I want to thank CONACYT for their scholarship that made it possible to keep studying for my master's degree.

Esta tesis ha sido parcialmente apoyada por el proyecto “Generación de estrategias científico-tecnológicas con un enfoque multidisciplinario e interinstitucional para afrontar la amenaza que representan los complejos ambrosiales en los sectores agrícola y forestal de México” del CONACYT-FORDECYT con número 292399.

Publications

Conference Papers

- R. Verdin, G. Ramirez, **C. Rivera**, and G. Flores. “Teleoperated aerial manipulator and its avatar. Communication, system’s interconnection, and virtual world.” International Conference on Unmanned Aircraft Systems 2019 (PUBLISHED).

Abstract

The present work is about the simulation of an aerial manipulator with a 2DoF robotic arm. First, the Unmanned Aerial Vehicle (UAV) and robotic arm's mathematical modeling is obtained. The UAV model equations consider a force and torque as unknown parameters that need to be compensated by the controllable inputs which are the UAV thrust force and torque. The manipulator's modeling is made through the Newton-Euler algorithm and leads to a Lagrangian expression form. This form is useful to program the manipulators controller and dynamics. The torque and force are estimated and compensated in the UAV model equations. To estimate the force and torque nonlinear observers are employed. A geometric controller on the Special Euclidean group ($SE(3)$) is designed for angular velocity considering angular velocity errors that are proven to converge to zero using the Lyapunov direct method. Also, a position controller is designed to compute the thrust force produced by the rotors so the UAV tracks the desired position. A Proportional-Derivative (PD) joint controller is designed to compute the manipulator joint torques to track a desired angular position. Also, manipulator controller stability has been proven using the Lyapunov method. Finally, the controller simulation results show the controller stability. The position controller tracks the desired position in the z axis direction and the angular velocity error converges to zero like the torque and force estimation errors, even with aggressive rotations. The aerial vehicle is simulated in

MATLAB and the controller is programmed in MATLAB-SIMULINK. A final simulation is made by plotting a representation of the UAV with the real measured parameters to simulate a real-time flight with the obtained values. A future work is presented where a rotation error will be considered so the UAV tracks a desired x, y position from the desired rotation. Also a publication related to this work is annexed as a complementary backup.

Introduction

Unmanned Aerial Vehicles (UAVs) are used nowadays to execute tasks where places are not reachable for humans or where the risk is too high. The advantage of an aerial vehicle is the fact that it can fly and stay suspended in a desired position in the air. Some tasks where UAVs have been used recently are buildings inspection, mapping tasks using Simultaneous Localization and Mapping (SLAM) techniques, transportation, and even for military support. [1,2]. Manipulation devices were added to UAVs to increase the range of possible executable tasks. This is how aerial manipulators become another alternative to different tasks where grasping or manipulation is required. Some examples are valve operation, maintenance, and repair. [3]. The correct operation of these vehicles requires to be analyzed and controlled. Despite what control theory is in a general form, in this specific case, control is in charge to make the vehicle move appropriately without falling, crashing, and capable to execute desired commands.

To understand the behavior of an aerial vehicle system, an interpretation has to be made in a form of differential equations, which is called the mathematical model. Also linear and nonlinear terms can be found in these equations. Being an aerial vehicle a nonlinear system, it is well known in control theory that nonlinear systems have to be analyzed by the use of tools that become a little more complex than the linear system ones. For this reason, tools like

Lyapunov stability analysis are taken to be applied in this work. The mathematical model describes the behavior of the linear and angular position, velocity, and acceleration of the vehicle. In this work, a mathematical model is obtained for an aerial manipulator. The model is obtained in three groups of equations, the position, the attitude, and the manipulator's position.

The manipulator produces a force and a torque that affect the system, these two unknown disturbances are considered in the model to be compensated. To estimate these two disturbances a state estimator (observer) is considered to estimate the manipulator dynamics (force,torque) and to be added in the control law, which have been proven to be stable using the Lyapunov direct method. The position controller is designed using also the Lyapunov method and it will return the desired Thrust force, which is a force produced by the propellers and normal to the UAV plane. For attitude, an angular velocity controller in the Special Euclidean group SE(3) is designed. The fact that this controller is designed in SE(3) makes the control consider the rotation and position of the body-fixed frame instead of the 3 Euler angles (ϕ, θ, ψ) and position individually.

This controller is capable to converge to a desired angular velocity which is obtained from the desired rotation matrix. Finally, a Proportional-Derivative (PD) position controller is designed for the manipulator that returns the desired torque for each joint so each link can move to a desired angular position. The simulation of the whole system is made via MATLAB-Simulink where the entrances are a desired angular velocity and a desired z position divided in two common scenarios where the vehicle rotates in a small time interval and where the vehicle is suspended in the air while the manipulator is doing a routine. Results show the system stability and also the virtual vehicle is simulated.

CHAPTER 1

State of the Art

Aerial manipulators face several problems that have been studied in the last years. One of them is the control problem that leads to some other considerations. The controller must be capable to stabilize the attitude and the position from some desired values. An aerial vehicle, with no arm, has 6 degrees of freedom divided into its coordinates and angular positions. The only variable that can be controlled is a thrust force and torque. Both are directly related to the speed of each of the rotors. If a robotic arm is added the control input is the same but the disturbances increase. Therefore they must be considered in the control law to compensate them. To achieve this compensation, two terms belonging to the manipulator, have to be considered, a force and a torque. Also, from its modeling, it is well known that both are non-linear terms. In dynamic modeling, adding a manipulator changes the property of symmetry in the design of the vehicle. The reason why these vehicles are designed like that is that the center of mass is conveniently placed in the center of the vehicle. If the symmetry is lost, the center of mass, where all the external forces and torques act, change, and the modeling of the whole system changes as well. However, in the next references, it can be seen by the description that the dynamic modeling, control, estimations, and vehicle design are the

research areas that might improve aerial vehicles. In reference [4], the problem is stated as the difficulty for aerial vehicles to access certain places to do manipulating tasks. To face this problem an augmented adaptive controller and an optimal baseline controller are designed. Both controllers are placed in the same architecture. The augmented adaptive controller adds a linear PI controller so it deals with uncertainties that will make the UAV system unstable. The optimal baseline controller is used to control the gripper and deal with disturbances and unwanted constant errors. The error convergence is demonstrated based on the Lyapunov theory. Finally, the controller shows a rapid error convergence, with or without a payload, compared with other controllers. In reference [5], a new control strategy for a 2DoF Aerial Vehicle is presented under the fact that by adding a manipulator the center of mass of the whole vehicle will be displaced. Therefore a backstepping sliding mode controller (BSC) is added considering the new position of the center of mass. The BSC with the center of gravity term shows better results than a PID controller and the BSC with no center of mass term. A control strategy might depend on the perspective of how the Aerial Manipulator is considered. Like the previous papers, the system can be considered as two separated systems coupled, each of them with their properties, or one single system. If the modeling considers some variations as disturbances or if considers those variations as constant terms, that will lead to different results and control strategies. Another way to model aerial manipulators is using quaternions like reference [6, 7]. A simple solution to stabilize a mini aerial manipulator is presented in [6], it has to be considered that the smaller distance from the center of the UAV to the rotor, the less torque can be produced. The controller is based on quaternions, due to the modeling and saturation functions, which are already stable as well. Also, a linear Luenberger observer is added to the controller to estimate the joint angles from the robotic arm. This estimation will provide a solution to estimate the torque produced by the manipulator using its motion equations. Nevertheless in [7], the authors take the torque and force terms, produced by the manipulator, to be linearized. This linearization simplifies the internal model control considered in this reference. Robust control is proposed in [8] but the problem to solve in this reference is the vertical take-off and landing of an aerial manipulator. Also, the vehicle is an air-ducted fan which has some changes in the motion equations than

a quadrotor or a hexacopter. The experimental results show the effect of the manipulator torque and force on vehicle stability. It can be noticed that even when the vehicle seems to be different the modeling results are pretty similar to a quadrotor or a hexacopter. The important terms are the thrust force and the cross product of force and distance that produce a torque. Flight stability is highly demanded on aerial manipulators (AM) and accuracy takes an important role like in [9]. The contribution is an accurate controller that provides a safe flight mode for outdoor tasks. Besides, it allows the manipulator to do more complex tasks during the flight.

Adaptive controllers became a good choice to control AMs due to their adaptability property. Nonlinear adaptive control for the attitude and position is proposed in [10]. This adaptive controller adapts to the change of mass and inertias produced by the manipulators. The adaptability previously mentioned makes the system capable to stabilize under the action of unknown terms. On the other hand, not only stability is required for some tasks but tracking the desired trajectory is also demanded. In [11] an adaptive controller provides the solution for this problem and considers the same unknown disturbances of torque and force (UDTF). The adaptive controller stabilizes the AM during the tracking of the desired trajectory. An \mathcal{L}_1 adaptive loop is what compensates for the UDTF caused by the manipulator.

Another problem set in [12] is the fact that manipulator force disturbance not only involves its weight but the payload too. Taking into consideration that the center of gravity is displaced by the manipulator, it suffers another displacement once a payload is grabbed. An adaptive fractional-order sliding mode control deals with these mass changes and center of gravity varying terms. Adding an adaptive correcting mass coefficient in the equations is how these mass variations are compensated.

As it was already mentioned state estimators are another option to make a good approach to unknown values. In general, state estimators provide information, that sensors wouldn't be able to give, from not measurable state variables. However, there are some estimators capable to estimate parameters inside the modeling such as disturbances. In [13] an adaptive output control of an AM is presented to stabilize the aerial vehicle. A PD controller is included but requires an error tracking derivative. Therefore a super twisting algorithm is used

to estimate the error derivatives and the adaptive gains are adjusted by differential laws. In reference, [14] an unscented Kalman filter is used as a UDTF estimator. It is explained in this reference that a conventional Kalman filter is not capable to estimate these variables due to the nonlinear behavior and complex dynamics. And more specifically Kalman filters are used as estimators for linear systems. However, the contribution is 2 variants of unscented Kalman filter that are capable to estimate the nonlinear varying terms. Another reference that uses state estimators for AMs is [15] but in this case, the estimated state variables are the linear position and velocity of the vehicle. These variables are included in the control to get better accuracy in the vehicle position. Other variables are measured using sensors but UDTF are not considered in the mathematical model.

Stability has been taken into account as the main problem for AMs but once a safe flight mode is provided the manipulator requires to do specific tasks that might need more than just flight stability. As an example in reference [16] A force control is designed for an AM. The contribution is focused on making the end effector to apply a force in the desired direction with a desired magnitude. A force term is considered in the position controller so the vehicle accelerates in one direction while the end effector is making contact with a surface until the desired magnitude is reached.

Applications for AMs are widely proposed but each leads to specific problems, a new design leads to different dynamics and specific tasks sometimes require more considerations in the controller. The Aerial Manipulator can be designed with a single-arm, dual-arm, or even collaborative [17], [18]. As examples of recent applications in [19] an AM is used for helping Power Line operators with device installing and delivering tools. In [20] a contact-based inspection aerial manipulator is proposed considering forces in the dynamic modeling and [21] propose a Hybrid Force and a Position controller for AM writing. Artificial Intelligence has been considered as well in AM stabilization, not replacing a controller but combining algorithms with convectional control tools like [22] where a PID is combined with Fuzzy Logic and Neural Network to improve the convectional PID controller applied in AMs. Finally but not less important, the vision systems have been recently combined with AMs like in [23] where the end-effector of the manipulator uses stereo vision to reach an object or avoid ob-

stacles at a certain distance. In [23] a PL-SLAM algorithm is proposed specifically for Aerial Manipulators for mapping areas and localizing themselves during the flight.

Theoretical Knowledge

In this chapter, the theoretical knowledge necessary for this work is presented. The most important topics are presented so the reader has an idea of what was made in every step in this work.

2.1 Mathematical Modelling of a Quadrotor

Mathematical modeling is a set of differential equations that describe the behavior of a system in a certain time lapse [24]. The behavior of an aerial vehicle can be represented in two parts, the position and the attitude leading to six involved variables $(x, y, z, \phi, \theta, \psi)$.

2.1.1 Position

It is well known from Newton's Inertia Law that a body will not move until a force is applied or if the sum of all forces interacting equals zero. Also from Newton's Second Law, the sum of all these forces is proportional to the body's acceleration [26].

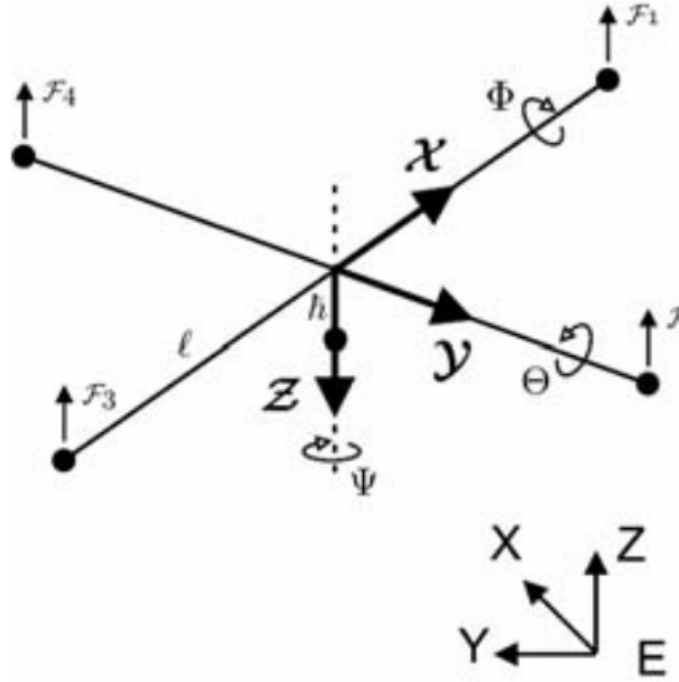


Figure 2.1: The position and attitude variables of the quadcopter. Also, the force that each rotor produces. [25]

$$\sum F = ma \quad (2.1)$$

This acceleration represents the behavior of the velocity which represents the behavior of the position in a time-lapse. From this expression, the position, velocity, and acceleration of an Aerial Manipulator can be described mathematically. An Aerial Vehicle has a symmetric design conveniently so the center of mass is located in the center of the vehicle. This is where all forces and torques involved will be acting. There's a thrust force that can be produced by the rotors shown in fig 2.1. This force combined with its orientation will produce the vehicle to move. Therefore the modeling can be expressed as follows

$$a = ge_3 - \frac{f}{m}Re_3 \quad (2.2)$$

where the vector $e_3 = (0, 0, 1)^T$, the acceleration vector $a = (\ddot{x}, \ddot{y}, \ddot{z})^T$, the gravity acceleration g , the thrust force f and vehicle's mass m . The thrust force is always normal to the rotors

therefore the direction of the force depends on the vehicle's attitude which can be expressed by a rotation matrix R .

$$R = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix} \quad (2.3)$$

Where terms c means the cosine function and s the sine function.

2.1.2 Attitude

The attitude of an aerial vehicle can be modified by the rotors spinning. If one rotor spins faster than the others then a momentum (torque) will be produced by the distance r , which is the rotor's link length, times the force F_r , produced by that rotor. Expressed mathematically $\tau = r \times F_r$. However, the expression that involves the angular acceleration and velocity is obtained from Newton-Euler equations of motion [26] where the angular momentum of a body is expressed as

$$\tau = J\dot{\Omega} + \Omega \times J\Omega \quad (2.4)$$

Where the torque vector $\tau = (\tau_x, \tau_y, \tau_z)^T$, the time derivative of the angular velocity $\dot{\Omega} = (\dot{\Omega}_x, \dot{\Omega}_y, \dot{\Omega}_z)^T$, the angular velocity $\Omega = (\dot{\phi}\hat{w}_x, \dot{\theta}\hat{w}_y, \dot{\psi}\hat{w}_z)^T$, being \hat{w} the unit vector in the Ω direction, and J represents the inertia matrix [27].

In this way, it can be noticed that the angular velocity of a rigid body relates to the angular rate of change that rotates over the angular velocity direction. The angular velocity representation can be appreciated in figure 2.2

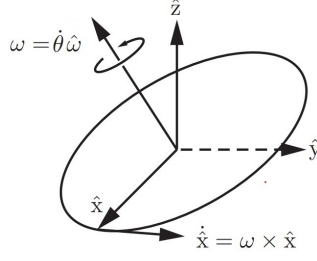


Figure 2.2: Angular velocity representation of a coordinate system. [27]

2.1.3 Quadrotor modelling equations

Finally the mathematical modelling of a quadrotor can be expressed using 2.2 and 2.4 as

$$\dot{x} = v \quad (2.5)$$

$$\dot{v} = ge_3 - \frac{f}{m}Re_3 \quad (2.6)$$

$$\dot{R} = R\hat{\Omega} \quad (2.7)$$

$$\dot{\Omega} = -J^{-1}(\Omega \times J\omega) + J^{-1}\tau \quad (2.8)$$

Where \dot{x} is the derivative of the position, \dot{v} is the derivative of the velocity, \dot{R} is the time derivative of the rotation matrix and $\hat{\Omega}$ is the Ω skew-symmetric matrix denoted by

$$\hat{\Omega} = \begin{pmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{pmatrix}$$

2.2 Lyapunov Stability: Direct Method

In the previous section, it can be appreciated that Quadrotor dynamics contain nonlinear terms, therefore this system is nonlinear. Nonlinear systems cannot be analyzed with linear stability theory tools. To analyze this system it is necessary to use different tools such as Lyapunov stability which is useful for linear and nonlinear systems.

First of all, the form that a nonlinear system has is the following [28]

$$\dot{x}(t) = f(x(t), u(t)), \forall t \geq 0 \quad (2.9)$$

where t denotes the time; $x(t)$ denotes the n -dimensional state variables vector; and $u(t)$ denotes the m -dimensional input vector of the system.

There are three main concepts that describe the stability of a system: stability, asymptotic stability and instability [28].

Let us consider a system in equilibrium that is not being disturbed and consider that the total energy of the system is known. There are three possible scenarios. If the energy of the system, once the disturbance acts, does not increase beyond the magnitude of the energy added by the disturbance, then the system is **stable**. Now if the energy of the system decreases such that eventually, the energy of the system gets to zero, then the system is **asymptotically stable**. Finally, if the energy of the system increases continuously with a bigger energy magnitude than the added by the disturbance, then the system is unstable.

Considering the state variables of a system. An equilibrium point is the value for the state variables that make $\dot{x}(t) = 0$. This equilibrium point represents the solution for x . Lets define x_e as that equilibrium point that makes the system equation $\dot{x}(t) = 0$. Lets define the initial condition x_0 and a region A, B around x_e . If x_0 stays inside the region A but outside region B then the system is stable like in figure 2.3. If x_0 tends to be equal to x_e , if it stays inside the region B the system is asymptotically stable. In other words, asymptotic stability happens if eventually, the state variable gets near to the center of figure 2.3.

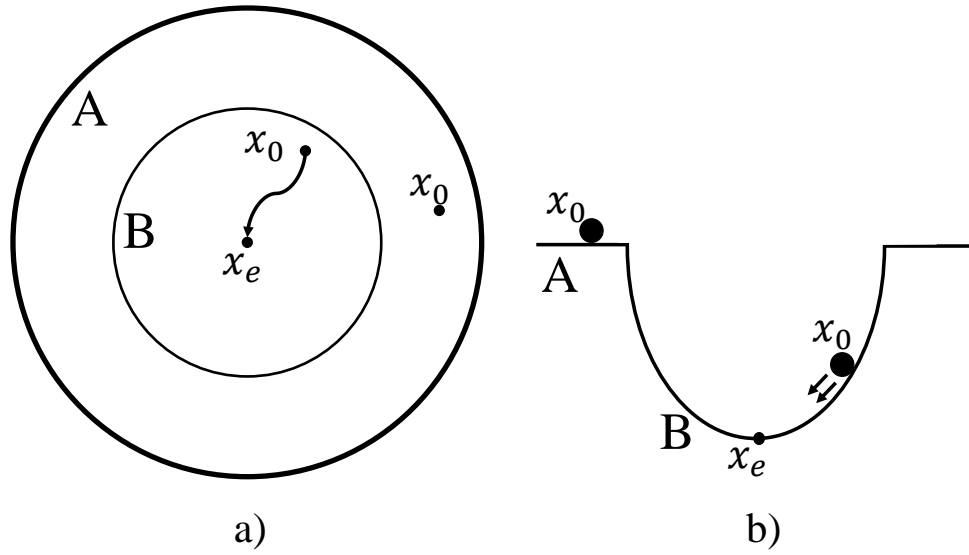


Figure 2.3: An interpretation of the stability criteria inspired on the geometrical representation in [29]. Figure *a*) shows two regions of stability, region A shows stability and region B shows asymptotic stability. Figure *b*) is a physical interpretation of this two regions considering the state x_0 as a ball that will not move inside region A because there's any force acting on it. However if the ball is inside region B the gravity force will push the ball to the equilibrium point x_e .

In the Lyapunov Direct Method, the stability can be proven by selecting a Lyapunov candidate function. This function depends on the state variables of the system and it can be considered as a region where the state variables have certain trajectories. These trajectories define the stability of the system. The Lyapunov candidate function V must have the following properties [30]:

- Scalar function

$$V : \mathbb{R}^n \rightarrow \mathbb{R}$$

- Positive definiteness

$$V(x) > 0$$

- Dissipativity

$$\frac{d}{dt}V(x) = \frac{\partial V}{\partial x} \frac{d(x)}{dt} < 0$$

Once the Lyapunov candidate function is selected, the stability of a system can be proven if \dot{V} is obtained.

- The system is stable if

$$\dot{V}(x) \leq 0$$

- The system is asymptotically stable if

$$\dot{V}(x) < 0$$

In figure 2.4 it is shown an example of a Lyapunov quadratic function where it is interpreted as a region where the bottom is meant to be the equilibrium point. The derivative represents the slope of the function. If this is analyzed as an energy function, a negative derivative means the loss of energy of a particle therefore at one point it will be at zero energy state which is the equilibrium point like was mentioned before. If there's no slope it means the particle is not losing energy but is not earning either as a consequence there's no energy at all. Finally, if the slope is positive it means the energy is growing which means the system is not stable.

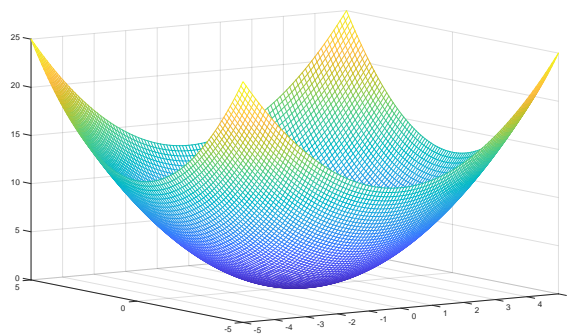


Figure 2.4: Geometrical three dimensional representation of the Lyapunov quadratic function.

2.3 Observers

The systems analyzed in control theory are modeled in an "ideal" scenario. In reality, the control input, such as actuators, does not only depend on one single variable but external disturbances that most of the time cannot be measured. For example, a flying quadrotor can be displaced by a disturbance such as wind. This disturbance might not be considered in mathematical modeling but it is present anyway. If this disturbance has to be measured whether it is going to be difficult, expensive, or even impossible to get a sensor capable to do that. If there's no feedback so the control input compensates for that disturbance then the system will be unstable, if the magnitude is big enough, or will not be stable asymptotically. Observers are used in control theory as a tool to estimate these unknown variables from the system. Observers work by taking information from the system combining it with a measurable feedback signal allowing them to get more precise knowledge of the system behavior [31]. In other words, observers take measurable states and use the model equations to get an approach of the unknown state by comparing the original equations with the estimated.

In the case of nonlinear systems, it is not necessary to consider special observers only for this type of system. If the error behavior (the difference between the real state and the estimated) is proven to have a linear form then an observer for a linear system can be considered as a solution. What's important for an observer is the error dynamics, the observability and if the states variables can be measured. Using the Lyapunov direct method it can be proven that an observer will converge to zero using the error equation. And the observability proves that the variable can be estimated or not.

From [32] a system with the form 2.9 the definition of observability is the following.

$$\dot{x}(t) = f(x(t), u(t), y(t), t) \quad (2.10)$$

$$\hat{x}(t) = h(x(t), u(t), y(t), t)$$

Where $\hat{x}(t)$ is the estimated state and h is the observer equation which is a function that

depends on the input $u(t)$, the state $x(t)$, the output $y(t)$ and the time t . Normally observers have the dynamic equation in the form.

$$\dot{\hat{x}}(t) = f(\hat{x}(t), u(t)) + k(t, h(\hat{x}(t)) - y(t)), \quad \text{with } k(t, 0) = 0. \quad (2.11)$$

such that:

1. $\hat{x}(0) = x(0) \Rightarrow \hat{x}(t) = x(t), \forall t \geq 0$;
2. $\|\hat{x}(t) - x(t)\| \rightarrow 0$ as $t \rightarrow \infty$;
 - If 2 holds for any $x(0), \hat{x}(0)$, the observer is global.
 - If 2 holds with exponential convergence, the observer is exponential.
 - If 2 holds with a convergence rate which can be tuned, the observer is tunable.

2.4 Robot Forward Kinematics

2.4.1 Denavit-Hartenberg

The Denavit-Hartenberg algorithm provides a homogeneous transformation matrix that contains information about the translation and rotation of coordinated systems according to the initial frame. This is a required tool to get the forward kinematics of robotic systems if every joint is considered as an i -th coordinate system. This forward kinematics will return the position of each joint taking the base of the robot as the initial frame, like in figure 2.5 every joint has its coordinate frame.

The homogeneous transformation A_i matrix is obtained considering a rotation in the z axis, a translation in the z, x direction, and a rotation in the x axis in this order [33].

$$A_i = R_{z, \theta_i} T_{z, d_i} T_{x, a_i} R_{x, \alpha_i} \quad (2.12)$$

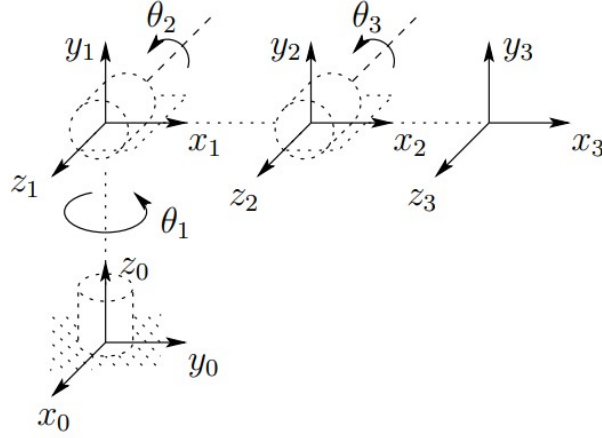


Figure 2.5: Coordinate frames of a 3DOF manipulator [33].

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The valuable information that transformation matrix A_i provides is the rotation matrix R and the translation vector $T = (t_x, t_y, t_z)$ with respect to the initial frame. The matrix R_i is obtained from the first rows and columns of the matrix A_i and the translation is obtained taking the 3 first rows of the fourth column.

The parameters $(\theta_i, d_i, a_i, \alpha_i)$ must to be obtained to compute the transformation matrix A_i . As it is described in the expression 2.12 and shown in figure 2.6, θ_i is the angle around

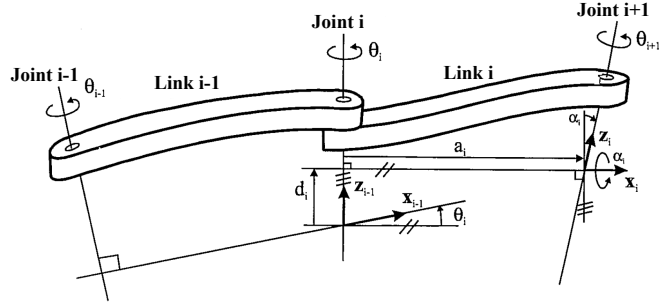


Figure 2.6: Graphical representation of the Denavit-Hartenber parameters [34].

the z axis, d_i is the distance along the z axis, a_i is the distance along the x axis and α_i is the angle around the x axes. Every parameter with respect to the next i frame. In other words this transformation matrix is obtained for each link i and the final transformation matrix is the multiplication of all the transformation matrices of each link.

2.5 Dynamics

2.5.1 Newton-Euler Formulation

The behavior of mechanical systems as a time function can be derived in a set of differential equations called the Euler-Lagrange equations. These equations are derived from Newton's second law of motion for a particle in two dimensions. In the case of a robotic arm, the Euler-Lagrange formulation returns a matrix equation that considers the Coriolis and Centrifugal effects (C), Mass (M), and Gravity (g) terms as well. The sum of all these terms return the Torque in each axis (τ_x, τ_y, τ_z) which represent the dynamic modeling of a manipulator [33].

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (2.13)$$

The physical terms involved in the dynamic modeling of a robotic arm are angular acceleration, velocity, and position. For this reason, kinematic modeling is necessary to get the previous formulation. However, The Euler-Lagrange formulation can be achieved by using

a method called Newton-Euler formulation which is expressed in a set of steps that can be interpreted as an algorithm to be programmed [33].

The difference between the Lagrangian formulation and the Newton-Euler formulation is the computational efficiency especially for many degrees of freedom manipulators. However, the resulting equations of motions must be identical [27].

The torque τ_i and force f_i of each link, using the Newton-Euler formulation, are computed as [33].

$$\tau_i = R_i^{i+1} \tau_{i+1} - f_i \times r_{i,ci} + (R_i^{i+1} f_{i+1}) \times r_{i+1,ci} + \alpha_i + \omega_i \times (I_i \omega_i) \quad (2.14)$$

$$f_i = R_i^{i+1} f_{i+1} + m_i a_{c,i} - m_i g_i \quad (2.15)$$

Where:

- $a_{c,i}$: the acceleration of the center of mass of link i
- $a_{e,i}$: the acceleration at the end of each joint where the next link is connected
- ω_i : the angular velocity of frame in joint i with respect to the initial frame ($i = 0$)
- α_i : the angular acceleration of frame in joint i with respect to the initial frame ($i = 0$)
- g_i : the gravity acceleration in frame i
- f_i : the force exerted by the previous link $i - 1$ on the next one i
- τ_i : the torque exerted by link $i - 1$ on link i
- R_i^{i+1} : the rotation matrix from frame $i + 1$ to frame i

The following terms belong to the manipulator dynamics but they are constant values therefore they do not depend on the manipulator configuration.

- m_i : the mass of link i

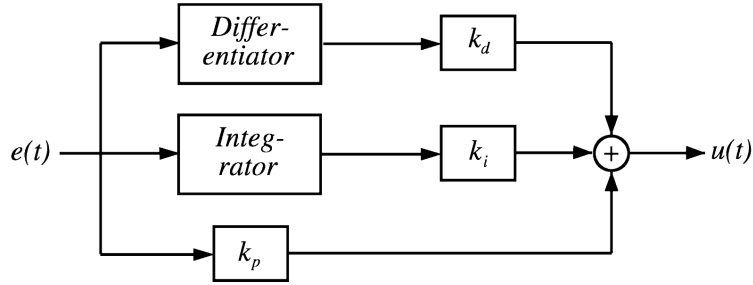


Figure 2.7: PID controller block diagram representation. [35]

- I_i : the inertia matrix of link i about a frame parallel to frame i whose origin is at the center of mass of link i
- $r_{i,ci}$: the vector from joint i to the center of mass of link i .
- $r_{i+1,ci}$: the vector from joint i
- $r_{i,i+1}$: the vector from joint i to joint $i + 1$

2.6 PID controller

A PID controller is a control structure that contains a proportional, integral, and derivative gain multiplied by an error. PID controllers are still powerful tools for physical systems due to the fact that PID controllers don't need information about the modeling (plant). Also, even if they are used in linear systems they are still useful for nonlinear systems if they are combined with terms that belong to the modeling. An input $u(t)$, which is the manipulable variable such as a voltage that moves an actuator, is the term that is also mathematically manipulable. This input will make the system to be stable in the desired state by using an error equation and proposing a control law, which is an equation for the system input $u(t)$. However in figure 2.7 $u(t)$ is shown as an output, this is because the diagram shows how the input $u(t)$ is being modified by the PID controller before it works as the system input.

The PID controller is then defined in [24] as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.16)$$

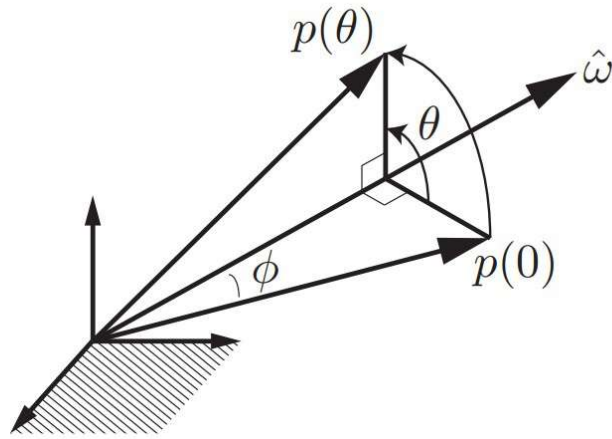


Figure 2.8: Vector $p(0)$ with a θ rotation over the \hat{w} axis. [27]

Where K_p, K_i, K_d are positive constant gains and $e(t)$ is the error that is a differential function that depends on the desired state and the actual state. The structure of this controller is represented in 2.7.

2.7 Exponential Coordinates of Rotations

Suppose that a three-dimensional vector $p(0)$ is rotated by θ about \hat{w} to $p(\theta)$ shown in figure 2.8. Let $p(t)$ denote the trajectory of the vector. The velocity of $p(t)$ is given by [27].

$$\dot{p} = \hat{w} \times p \quad (2.17)$$

This differential equation can be expressed as

$$\dot{p} = [\omega]p \quad (2.18)$$

This expression has a linear differential equation form $\dot{x} = Ax$ which is well known that its solution is given by $x = e^{At}$. Therefore

$$p(t) = e^{\hat{w}t} p(0) \quad (2.19)$$

Taking in consideration the expressions 2.18 and 2.19. A new rotation matrix can be computed since an angular velocity Ω is computed by the skew-symmetric matrix $\hat{\Omega}$. In other words, the solution for the differential equation shown in 2.7 is

$$R = R(0)e^{\hat{\Omega}t} \quad (2.20)$$

Where $R(0)$ represents the initial orientation of the rigid body. This expression can be solved also by using computational tools.

CHAPTER 3

Methodology

In this chapter, the modeling and method to deal with the control problem are presented. Every subsection details each element that compose the final system.

3.1 The control problem

The control problem comes from the dynamics of the system. An aerial vehicle by itself requires an attitude and a position controller considering external disturbances such as wind or drag parameters. However, a robotic arm attached to the center of mass of the UAV can be considered as an external disturbance as well. To simulate the aerial vehicle, the dynamics and control have to be programmed. The simulation is divided in two independent elements, the robotic arm and the aerial vehicle like in figure 3.1. The problem is solved by proposing an observer for the disturbances caused by the robotic arm which makes it possible to design a controller.

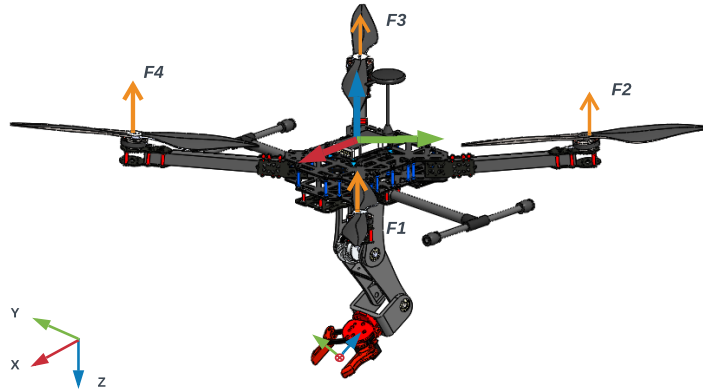


Figure 3.1: Aerial manipulator CAD model. [36]

3.2 Aerial Vehicle

3.2.1 Modelling

An aerial vehicle can be modelled by using 2.5-2.8. As it was previously mentioned, the manipulator is considered as a disturbance or external force/torque terms. To model the aerial manipulator these force and torque terms are included in the aerial vehicle equations. Therefore the modeling of the aerial manipulator is the following

$$\dot{x} = v \quad (3.1)$$

$$\dot{v} = ge_3 - \frac{f}{m}Re_3 + F_a(t) \quad (3.2)$$

$$\dot{R} = R\hat{\omega} \quad (3.3)$$

$$\dot{\Omega} = -J^{-1}\Omega \times J\Omega + J^{-1}\tau + T(t) \quad (3.4)$$

where $F_a(t)$, $T(t)$ are the unknown force and torque terms belonging to the manipulator.

3.3 Position and Angular Velocity Control equations

For position controller the input trust is defined as

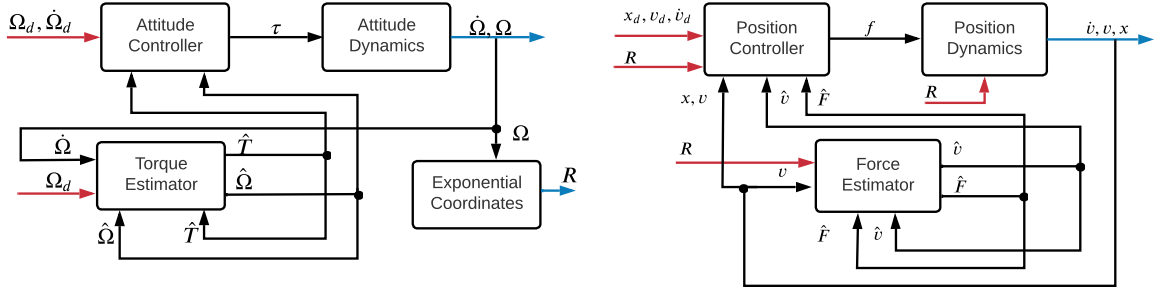


Figure 3.2: Control block diagram for the position and attitude controller.

$$f = m \| g e_3 + k_v e_v + k_p e_p - \ddot{x}_d + \hat{F}_a(t) \| \quad (3.5)$$

Where velocity and position errors e_v, e_p are defined in 3.11, $\hat{F}_a(t)$ is the manipulator estimated force, m is the vehicle's total mass, g is the gravity acceleration and e_3 is the unit vector $[0, 0, 1]^T$. This controller is in charge to produce a thrust force big enough to stabilize the Aerial Manipulator at a certain attitude in a certain rotation.

For attitude controller the torque input τ is defined as

$$\tau = J(-K_1(\hat{\Omega} - \Omega_d) + \dot{\Omega}_d + J^{-1}\Omega \times J\Omega - \hat{T}(t)) \quad (3.6)$$

Where J is the inertia matrix, K_1 is a positive diagonal matrix, $\hat{T}(t)$ is the manipulator estimated torque, and Ω, Ω_d are the actual and desired angular velocity. The input torque τ is in charge to move the aerial vehicle to the desired orientation.

In figure 3.2 the structure of the position and attitude system is represented in a block diagram.

3.4 Force and Torque Observer

3.4.1 Force and Torque Observer equations

To estimate the force an adaptive observer is designed. An adaptive observer is capable to estimate unknown parameters that are not considered as state variables with a nonlinear

behavior, like in [37] where an adaptive observer has been used to estimate an unknown wave model parameter. The proposed force observer is given by

$$\dot{\hat{v}} = ge_3 - \frac{u}{m}Re_3 + \hat{F}_a + k_{v_e}(v - \hat{v}) \quad (3.7)$$

$$\dot{\hat{F}}_a = k_F(v - \hat{v}) \quad (3.8)$$

where: \hat{F}_a is the estimated force and k_{v_e}, k_F are positive diagonal matrices.

This observer is proposed under the assumption that \dot{F}_a is near to 0. In other words this means that the force produced by the robotic arm must change slowly even when its behavior is nonlinear.

The observer for the torque estimation is given by.

$$\dot{\hat{\Omega}} = -K_1(\hat{\Omega} - \Omega_d) + \dot{\Omega}_d - K_2(\hat{\Omega} - \Omega) \quad (3.9)$$

$$\dot{\hat{T}} = -L_1(\hat{\Omega} - \Omega) - L_2\hat{T} \quad (3.10)$$

where $L_1(e_\Omega) = \text{diag} \left[\frac{l_1^p}{|e_\Omega^p| + \varepsilon}, \frac{l_1^q}{|e_\Omega^q| + \varepsilon}, \frac{l_1^r}{|e_\Omega^r| + \varepsilon} \right]$ and $L_2 = \text{diag} [l_2^p, l_2^q, l_2^r]$ with positive real numbers l_1^p, l_1^q, l_1^r and l_2^p, l_2^q, l_2^r . K_1 and K_2 are diagonal positive real matrices.

3.5 Stability demonstration

As it is shown in equation 3.2 the position x depends on the input force f in the direction of the rotation matrix R . For the control design of the system 3.1, 3.2 lets define the error as

$$e_p = x - x_d, e_v = v - v_d \quad (3.11)$$

The error dynamics are given by

$$\dot{e}_p = e_v = \dot{x} - \dot{x}_d \quad (3.12)$$

$$\dot{e}_v = \dot{v} - \dot{v}_d = ge_3 - \frac{f}{m}Re_3 + F_a(t) - \ddot{x}_d \quad (3.13)$$

$$fRe_3 = m(ge_3 + K_v e_v + K_p e_p - \ddot{x} + \hat{F}_a) \quad (3.14)$$

The stability can be proven by considering the following candidate Lyapunov function

$$V(e_p, e_v) = \frac{1}{2}K_p \|e_p\|^2 + \frac{1}{2}\|e_v\|^2 \quad (3.15)$$

The derivative of the Lyapunov function is

$$\dot{V} = K_p e_p e_v + e_v (ge_3 - \frac{f}{m} Re_3 + Fa - \ddot{x}_d) = -K_v e_v^2 \leq 0 \quad (3.16)$$

Substituting the control law 3.14 in 3.16.

$$\dot{V} = -K_v e_v^2 \leq 0 \quad (3.17)$$

It can be appreciated from 3.17 that this expression proves stability but not asymptotic stability. The idea is that only for a point ($e_p = 0, e_v = 0$) the system will be in equilibrium therefore will be no longer asymptotically stable but stable due to $\dot{V} = 0$. However, any desired state $x_d \neq x$ leads to ($e_p \neq 0, e_v \neq 0$) by simply checking the error equations 3.11. In other words, the derivative of $x - x_d$ leads to $e_v^2 > 0$. The only case where $e_p \neq 0$ and $e_v = 0$ is when the desired position is different than the actual but the vehicle is not moving, that makes e_p to be constant which is not possible due to the acceleration of the vehicle provoked by the input force f proportional to e_p .

To prove force observer stability the following Lyapunov candidate function is proposed similar as 3.15

$$V(e_{v_e}, e_F) = \frac{1}{2}k_F \|e_{v_e}\|^2 + \frac{1}{2}\|e_F\|^2 \quad (3.18)$$

Where the errors are defined as

$$e_{v_e} = v - \hat{v} \quad e_F = F_a - \hat{F}_a \quad (3.19)$$

The error dynamics of both errors are obtained, using the observer equations 3.7, 3.8, as the following

$$\dot{e}_{v_e} = \dot{v} - \hat{v} = ge_3 - \frac{f}{m}Re_3 + F_a(t) - ge_3 + \frac{f}{m}Re_3 - \hat{F}_a - k_{v_e}(v - \hat{v}) = e_F - k_{v_e}e_{v_e} \quad (3.20)$$

Considering that $\dot{F} \approx 0$

$$\dot{e}_F = \dot{F} - \hat{F} = -k_F e_{v_e} \quad (3.21)$$

Then the derivative of the Lyapunov function 3.18 is

$$\dot{V} = k_F e_{v_e} e_{v_e} + \dot{e}_F e_F = k_F (e_F - k_{v_e} e_{v_e}) e_{v_e} - k_F e_{v_e} e_F \quad (3.22)$$

$$\dot{V} = -k_F k_{v_e} e_{v_e}^2 \leq 0 \quad (3.23)$$

Similar to 3.17 the system will be asymptotically stable only when $e_{v_e} = 0$ however it is logical that this will happen only when $v = \hat{v}$ otherwise the system will be asymptotically stable. In other words, if there exists a difference between v and \hat{v} the observer will asymptotically converge to zero.

For attitude the input torque τ in equation 3.4 is defined as

$$\tau = J(-K_1(\hat{\Omega} - \Omega_d) + \dot{\Omega}_d + J^{-1}(\Omega \times J\Omega) - \hat{T}(t)) \quad (3.24)$$

Let define the errors

$$e_{\Omega} = \hat{\Omega} - \Omega, \quad e_{T_a} = \hat{T} - T \quad (3.25)$$

And the error dynamics of each error are

$$\dot{e}_{\Omega} = \dot{\hat{\Omega}} - \dot{\Omega} = -K_1(\hat{\Omega} - \Omega_d) + \dot{\Omega}_d - K_2(\hat{\Omega} - \Omega) + J^{-1}(\Omega \times J\Omega) - J^{-1}\tau - T(t) \quad (3.26)$$

$$e_{T_a} \dot{=} \hat{T} - \dot{T} = -L_1(\hat{\Omega} - \Omega) - L_2\hat{T} - \dot{T} \quad (3.27)$$

The Lyapunov candidate function to prove observer stability is

$$W = \bar{e}_{\Omega}^T A^{\Omega} \bar{e}_{\Omega} + \bar{e}_{T_a}^T A^{T_a} \bar{e}_{T_a} \quad (3.28)$$

Where $A^{\Omega}, A^{T_a} \in \mathbb{R}^{3 \times 3}$ are diagonal positive matrices, and

$$\bar{e}_{\Omega} = \left[|e_{\Omega}^p|^{1/2} \text{sgn}(e_{\Omega}^p), |e_{\Omega}^q|^{1/2} \text{sgn}(e_{\Omega}^q), |e_{\Omega}^r|^{1/2} \text{sgn}(e_{\Omega}^r) \right] \quad (3.29)$$

$$\bar{e}_{T_a} = \left[|e_{T_a}^p|^{1/2} \text{sgn}(e_{T_a}^p), |e_{T_a}^q|^{1/2} \text{sgn}(e_{T_a}^q), |e_{T_a}^r|^{1/2} \text{sgn}(e_{T_a}^r) \right] \quad (3.30)$$

A sign function has an indefinite derivative at the origin, for this reason W is only differentiable at any value except for $[e_{\Omega}, e_{T_a}]^T = [0, 0]^T$. Therefore.

$$\dot{W} = [\text{sgn}(e_{\Omega}^p), \text{sgn}(e_{\Omega}^q), \text{sgn}(e_{\Omega}^r)] A^{\Omega} \begin{bmatrix} \dot{e}_{\Omega}^p \\ \dot{e}_{\Omega}^q \\ \dot{e}_{\Omega}^r \end{bmatrix} + [\text{sgn}(e_{T_a}^p), \text{sgn}(e_{T_a}^q), \text{sgn}(e_{T_a}^r)] A^{T_a} \begin{bmatrix} \dot{e}_{T_a}^p \\ \dot{e}_{T_a}^q \\ \dot{e}_{T_a}^r \end{bmatrix} \quad (3.31)$$

Substituting the control input τ from 3.24 and the error dynamics from 3.26, 3.27 in 3.31

$$\dot{W} = \text{sgn}(e_{\Omega}) A^{\Omega} [-K_2 e_{\Omega} + e_{T_a}] + \text{sgn}(e_{T_a}) A^{T_a} [-L_1(e_{\Omega}) - L_2(e_{T_a} + T) - \dot{T}] \quad (3.32)$$

Assuming bounded torques, i.e.

$$\|T\| \leq c \|e_{T_a}\|, \text{ and } \|\dot{T}\| \leq d \|e_{T_a}\| \quad (3.33)$$

where for a vector x the norm is defined as $\|x\| := \sum_{i=1}^n |x_i|$ is the $L1$ -norm and for a matrix

A the induced matrix norm is $\|A\| := \max_{i \leq j \leq n} \sum_{i=1}^m |a_{ij}|$. Then

$$\dot{W} \leq -\text{sgn}(e_\Omega) A^\Omega K_2 e_\Omega + \|A^{T_a} L_1 e_\Omega\| - \text{sgn}(e_{T_a}) A^{T_a} L_2 e_{T_a} + \|A^\Omega e_{T_a}\| + c \|A^{T_a} e_{T_a}\| + d \|A^{T_a} e_{T_a}\| \quad (3.34)$$

it follows that

$$\dot{W} \leq -\bar{e}_\Omega^T A^\Omega K_2 \bar{e}_\Omega + \|A^{T_a}\| \bar{e}_\Omega^T L_1 \bar{e}_\Omega - \bar{e}_{T_a}^T A^{T_a} L_2 \bar{e}_{T_a} + (\|A^\Omega\| + c \|A^{T_a}\| + d \|A^{T_a}\|) \bar{e}_{T_a}^T I_3 \bar{e}_{T_a} \quad (3.35)$$

$$\dot{W} \leq -\bar{e}_\Omega^T (A^\Omega K_2 - \|A^{T_a}\| L_1) \bar{e}_\Omega - \bar{e}_{T_a}^T (A^{T_a} L_2 - \|\bar{A}\|) \bar{e}_{T_a} \quad (3.36)$$

where $\bar{A} = A^\Omega + (c+d)A^{T_a}$. The previous expression holds as long as the matrix inequality $A^{T_a} L_2 > \alpha I_3$ holds. This is possible since A^{T_a}, L_2 are diagonal matrices with free positive parameters. Matrix inequality $A^\Omega K_2 > \|A^{T_a}\| L_1(e_\Omega)$ holds as long as $A^\Omega K_2 > \|A^{T_a}\| \bar{L}_1$ where the constant matrix $\bar{L}_1 = \text{diag}[\frac{l_1^p}{\varepsilon}, \frac{l_1^q}{\varepsilon}, \frac{l_1^r}{\varepsilon}]$. Since A^Ω, K_2 and \bar{L}_1 are constant matrices with free parameters, $A^\Omega K_2 > \|A^{T_a}\| L_1(e_\Omega)$ holds. Thus

$$\dot{W} \leq -\bar{e}_\Omega^T (A^\Omega K_2) \bar{e}_\Omega + \bar{e}_\Omega^T (\|A^{T_a}\| \bar{L}_1) \bar{e}_\Omega - \bar{e}_{T_a}^T (A^{T_a} L_2) \bar{e}_{T_a} + \|\bar{A}\| \bar{e}_{T_a}^T \bar{e}_{T_a} \quad (3.37)$$

Since

$$\lambda_{\min}\{A^\Omega\} \|\bar{e}_\Omega\|_2^2 \leq W_\Omega(\bar{e}_\Omega) \leq \lambda_{\max}\{A^\Omega\} \|\bar{e}_\Omega\|_2^2 \quad (3.38)$$

$$\lambda_{\min}\{A^{T_a}\} \|\bar{e}_{T_a}\|_2^2 \leq W_{T_a}(\bar{e}_{T_a}) \leq \lambda_{\max}\{A^{T_a}\} \|\bar{e}_{T_a}\|_2^2 \quad (3.39)$$

where $\|\bar{e}_\Omega\|_2$ is the Euclidean norm, and since

$$-\bar{e}_\Omega^T A^\Omega K_2 \bar{e}_\Omega \leq -\lambda_{\min}\{A^\Omega K_2\} \|\bar{e}_\Omega\|_2^2 \leq -\frac{\lambda_{\min}\{A^\Omega K_2\}}{\lambda_{\max}\{A^\Omega\}} W_\Omega \quad (3.40)$$

$$-\bar{e}_{T_a}^T A^{T_a} K_2 \bar{e}_{T_a} \leq -\lambda_{\min}\{A^{T_a} L_2\} \|\bar{e}_{T_a}\|_2^2 \leq -\frac{\lambda_{\min}\{A^{T_a} L_2\}}{\lambda_{\max}\{A^{T_a}\}} W_{T_a} \quad (3.41)$$

and also since $\|C\| = \lambda_{\max}(C)$ for every diagonal matrix $C^{n \times n}$, and the fact that $V \leq \|Cx\| \|x\|$ for every Lyapunov function $V \leq x^T Cx$ it follows

$$\dot{W} \leq -\frac{\lambda_{\min}\{A^\Omega K_2\}}{\lambda_{\max}\{A^\Omega\}} W_\Omega + \frac{\sqrt{\lambda_{\max}(A^{T_a} \bar{L}_1)}}{\sqrt{\lambda_{\min}(A^{T_a} \bar{L}_1)}} W_\Omega - \frac{\lambda_{\min}\{A^{T_a} L_2\}}{\lambda_{\max}\{A^{T_a}\}} W_{T_a} + \frac{\sqrt{\lambda_{\max}(\bar{A})}}{\sqrt{\lambda_{\min}(\bar{A})}} W_{T_a} \quad (3.42)$$

$$\dot{W} = -(\alpha_1 - \alpha_2) W_\Omega - (\beta_1 - \beta_2) W_{T_a} < 0 \quad (3.43)$$

Choosing $\alpha_1 > \alpha_2$, and $\beta_1 > \beta_2$. And since W is radially unbounded $e_\Omega(t)$ and $e_{T_a}(t)$ globally converge to zero exponentially.

3.6 Manipulator

A 2 degrees of freedom robotic arm is chosen as the manipulator.

3.6.1 Robot Kinematics

To start with the robot dynamic analysis the kinematics is necessary to get to a Lagrangian formulation through the iterative Newton-Euler method. The information about translations and rotations in each joint frame will be needed for further dynamic analysis. The transformation matrices provided by the Denavit-Hartenber (D-H) algorithm contain this information.

To obtain the transformation matrices The initial configuration of the robotic arm is the following. An extra degree of freedom is considered in the initial frame as a static rotational joint so the final coordinates consider the link a_0 .

From figure 3.3 the Denavit-Hartenberg parameters are shown in table 3.1. The configuration as been chosen in that form to simplify the dynamics expression. It is not necessary to know the joints position for this work, therefore the UAV frame and the manipulator's initial

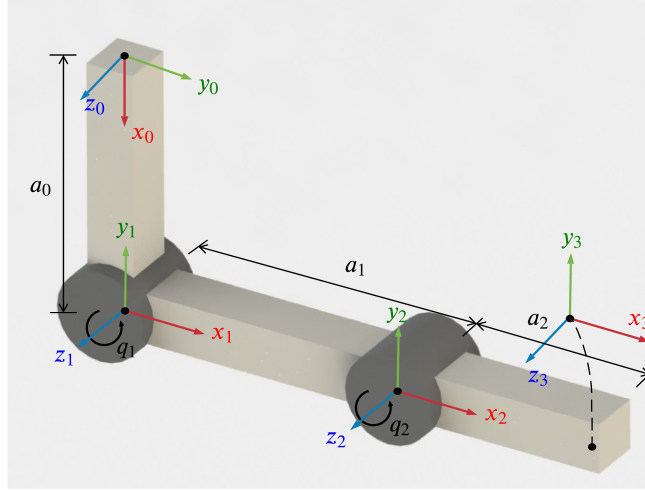


Figure 3.3: Initial position of the 2DOF manipulator.

frame are chosen to not be aligned. However the position can be known by just considering the transformation between the UAV frame and the manipulator frame.

q_i	d_i	a_i	α_i
0	0	a_0	0
$q_1 + 90^\circ$	0	a_1	0
q_2	0	a_2	0

Table 3.1: Denavit-Hartenberg parameters.

Once the parameters are obtained the transformation matrices are obtained as

$$A_{01} = \begin{pmatrix} 1 & 0 & 0 & a_0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_{12} = \begin{pmatrix} \cos(q_2 + 90) & -\sin(q_2 + 90) & 0 & a_1 \cos(q_2 + 90) \\ \sin(q_2 + 90) & \cos(q_2 + 90) & 0 & a_1 \sin(q_2 + 90) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_{23} = \begin{pmatrix} \cos(q_3) & -\sin(q_3) & 0 & a_2\cos(q_3) \\ \sin(q_3) & \cos(q_3) & 0 & a_2\sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The final transformation from the initial frame to the end-effector frame is expressed by

$$T_{03} = A_{01}A_{12}A_{23} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{pmatrix} \quad (3.44)$$

The end effector coordinates (x, y, z) are the values t_x, t_y, t_z that belong to the positions r_{14}, r_{24}, r_{34} from the matrix T_{03} . The rotation matrix from the initial to the end effector frame is obtained from T_{03} as

$$R_{03} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

Now the orientation and position of the end effector is a known value that depends on the input joint angles q_1, q_2 .

3.6.2 Robot Dynamics

Once the forward kinematics is solved the Newton-Euler method, to find the torques and forces exerted in each joint, can be used. The torques, in this case, have to be computed for 3 joints because a third static joint is considered.

For the controller it is required to get the torques τ_1, τ_2, τ_3 and for the UAV controller it is only required τ_1 3.4. This is because the Newton-Euler algorithm returns a function that depends on all the joints located after that joint. Therefore the joint 1 has a torque function

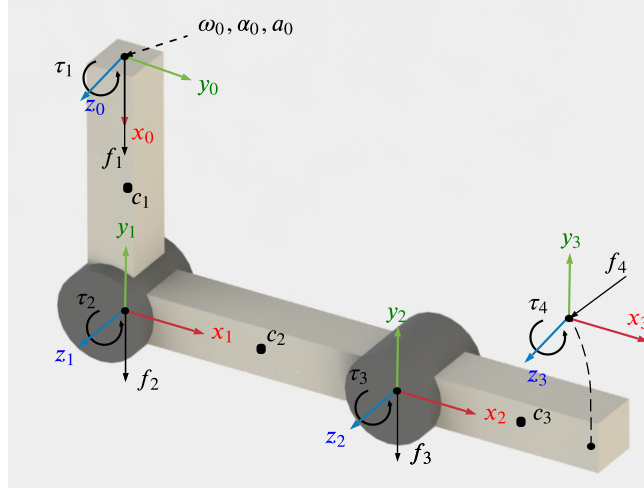


Figure 3.4: Torque and force representations in each joint of the manipulator.

that depends on τ_2, τ_3 . The main step is to find the angular acceleration and angular rate of each joint (α_i, ω_i) starting from the first joint to the last one. Once the angular rates and accelerations are obtained the torques and forces can be obtained for each joint starting from the last joint to the first one.

τ_1, τ_2, τ_3 are computed using the following equations.

$$\tau_3 = R_{34}\tau_4 - f_3 \times r_{3,c3} + (R_{34}f_4) \times r_{4,c3} + \alpha_3 + \omega_3 \times (I_3\omega_3) \quad (3.45)$$

$$\tau_2 = R_{23}\tau_3 - f_2 \times r_{2,c2} + (R_{23}f_3) \times r_{3,c2} + \alpha_2 + \omega_2 \times (I_2\omega_2) \quad (3.46)$$

$$\tau_1 = R_{12}\tau_2 - f_1 \times r_{1,c1} + (R_{12}f_2) \times r_{2,c1} + \alpha_1 + \omega_1 \times (I_1\omega_1) \quad (3.47)$$

f_1, f_2, f_3 are computed using the following equations

$$f_3 = R_{34}f_4 + m_3a_{c,3} - m_3g_3 \quad (3.48)$$

$$f_2 = R_{23}f_3 + m_2a_{c,2} - m_2g_2 \quad (3.49)$$

$$f_1 = R_{12}f_2 + m_1a_{c,1} - m_1g_1 \quad (3.50)$$

where:

R is the rotation matrix from one joint coordinate system to the other (corresponding to a ZYX rotation).

r is the position vector from the joint to the next joint or the center of gravity of the link.

$a_{c,i}$ is the linear acceleration of each link in the center of gravity.

g_i is the gravity vector located in the center of mass of each link.

f_4, τ_4 are equal to 0 if there's no an external force/torque applied on the end effector.

The equations to compute the linear accelerations in the center of mass of each link $a_{c,1}, a_{c,2}, a_{c,3}$ are the following

$$a_{c,1} = R_{01}^T a_{e,0} + \dot{\omega}_1 \times r_{1,c1} + \omega_1 \times (\omega_1 \times r_{1,c1}) \quad (3.51)$$

$$a_{c,2} = R_{12}^T a_{e,1} + \dot{\omega}_2 \times r_{2,c2} + \omega_2 \times (\omega_2 \times r_{2,c2}) \quad (3.52)$$

$$a_{c,3} = R_{23}^T a_{e,2} + \dot{\omega}_3 \times r_{3,c3} + \omega_3 \times (\omega_3 \times r_{3,c3}) \quad (3.53)$$

The linear accelerations in the end of each link $a_{e,1}, a_{e,2}$ are computed using the equations.

$$a_{e,1} = R_{01}^T a_{e,0} + \dot{\omega}_1 \times r_{1,2} + \omega_1 \times (\omega_1 \times r_{1,2}) \quad (3.54)$$

$$a_{e,2} = R_{12}^T a_{e,1} + \dot{\omega}_2 \times r_{2,3} + \omega_2 \times (\omega_2 \times r_{2,3}) \quad (3.55)$$

The angular accelerations $\alpha_1, \alpha_2, \alpha_3$ are computed using the following equations ($q_0 = 0$ is the static joint).

$$\alpha_1 = \dot{\omega}_1 = R_{01}^T \alpha_0 + R_{01}^T e_3^T \ddot{q}_0 + \omega_1 \times R_{01}^T e_3^T \dot{q}_0 \quad (3.56)$$

$$\alpha_2 = \dot{\omega}_2 = R_{12}^T \alpha_1 + R_{02}^T e_3^T \ddot{q}_1 + \omega_2 \times R_{02}^T e_3^T \dot{q}_1 \quad (3.57)$$

$$\alpha_3 = \dot{\omega}_3 = R_{23}^T \alpha_2 + R_{03}^T e_3^T \ddot{q}_2 + \omega_3 \times R_{03}^T e_3^T \dot{q}_2 \quad (3.58)$$

Finally the angular velocities $\omega_1, \omega_2, \omega_3$ are computed using the following equations.

$$\omega_1 = R_{01}\omega_0 + R_{01}e_3^T \quad (3.59)$$

$$\omega_2 = R_{12}\omega_1 + R_{02}e_3^T \quad (3.60)$$

$$\omega_3 = R_{23}\omega_0 + R_{03}e_3^T \quad (3.61)$$

At this point, it is important to mention that the initial conditions $\omega_0, \alpha_0, a_{e,0}$ are input values coming from the UAV. Supposing the UAV is a moving platform then the accelerations and velocities from the UAV will be transmitted to each link of the robotic arm. If the torque equations are computed symbolically the final expression can be grouped to get to a Lagrangian formulation that will be needed to apply a controller. The Lagrangian formulation has the form.

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (3.62)$$

For the record the term $M(q)$ is the $n \times n$ inertia matrix, $C(q, \dot{q})$ is the Coriolis/Centrifugal matrix and $g(q)$ is the gravitational vector.

3.6.3 Multi-variable Joint Control

In [27, 33] a multi-variable joint control is proposed. The variable that can be controlled in this case is the torque which is related to the voltage applied to each motor. Therefore the torque is chosen as the input u and a simple PD controller is applied. Being $u = \tau$ then the dynamics are expressed by

$$u = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (3.63)$$

By setting u as the input \ddot{q} will be obtained and integrating twice a new angle will be

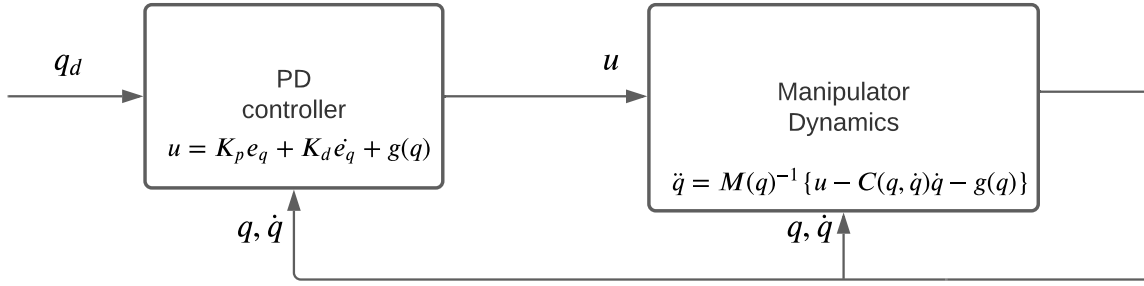


Figure 3.5: Control diagram for the manipulator PD controller

obtained from the torque input. From equation 3.63 \ddot{q} , is computed.

$$\ddot{q}_{new} = M(q)^{-1} \{u - C(q, \dot{q})\dot{q} - g(q)\} \quad (3.64)$$

The control law for u as it was mentioned before it is only a PD controller.

$$u = K_p e_q + K_d \dot{q} + g(q) \quad (3.65)$$

The angular position error $e_q = q_d - q$ and K_p, K_d are diagonal positive gain matrices. The last term $g(q)$ is added to compensate for the gravity effects on each link when the manipulator is not moving.

The control diagram for the manipulator is shown in fig 3.5

3.6.4 Manipulator PD control stability

The stability for 3.63 using the control input 3.65 is proven using a Lyapunov candidate function

$$V = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} e_q^T K_p e_q \quad (3.66)$$

Considering that $M(q) = D(q) + J$ where J is constant [33]. The time derivative of the Lyapunov function is

$$\dot{V} = \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{D}(q) \dot{q} + \dot{e}_q^T K_p e_q \quad (3.67)$$

Considering $\dot{e}_q = -\dot{q}$ due to $\dot{q}_d = 0$ and substituting $M(q)\ddot{q}$ using 3.63

$$\dot{V} = \dot{q}^T (u - C(q, \dot{q})\dot{q} - g(q)) + \frac{1}{2} \dot{q}^T \dot{D}\dot{q} - \dot{q}^T K_p e_q \quad (3.68)$$

Simplifying

$$\dot{V} = \dot{q}^T (u - g(q) - K_p e_q) + \frac{1}{2} \dot{q}^T (\dot{D}(q) - 2C(q, \dot{q}))\dot{q} \quad (3.69)$$

From [33] it is proven that $\dot{D}(q) - 2C(q, \dot{q})$ is a skew-symmetric matrix therefore $\dot{q}^T (\dot{D}(q) - 2C(q, \dot{q}))\dot{q} = 0$ and substituting the control input 3.65

$$\dot{V} = -\dot{q}^T K_D \dot{q} \leq 0 \quad (3.70)$$

A similar situation to the position of aerial manipulator control is presented here. When the angular velocity $\dot{q} = 0$ the system will be stable but not asymptotically stable. This will only be possible if the desired position $q_d \neq q$ and the angular speed $\dot{q} = 0$. This is not possible due to the position error e_q in the control input. The torque change leads to an acceleration change that leads to a velocity change as well. For this reason, the system will be asymptotically stable unless the desired position and the actual are equal.

CHAPTER 4

Experiments and results

The experimental results of this work focus mainly on the simulation of all the previous equations to prove the effectiveness and show the behavior of the state variables. This chapter will be focused on two simulations that demonstrate attitude stability and position stability. The following table contains all the parameters used to simulate the Aerial Manipulator. It is important to mention that these simulations do not consider an x_d, y_d controller since the desired rotation is proposed directly without considering the position in x, y when the vehicle is rotating. This work is not focused on the x, y position controller. However, if the attitude controller is stable then the position x, y will simply become a function that relates the desired rotation with the desired position.

4.1 Position and Manipulator Controller

The simulations were programmed in MATLAB Simulink environment using a fixed-step of 1×10^{-5} . In this simulation the position controller is considered only in the z direction because the x_d and y_d depend on specific desired Euler angles ϕ_d, θ_d, ψ_d or a desired rotation

Parameters	Values
UAV physical parameters	$m = 2.5\text{kg}, J_{xx} = 2.8 \times 10^{-7}, J_{yy} = 5.3 \times 10^{-7}, J_{zz} = 2.9 \times 10^{-7}$
Manipulator physical parameters	$l_1 = 12.5\text{cm}, l_2 = 10\text{cm}, m_{l_1} = m_{l_2} = 100\text{gr}$
Position Controller Gains	$K_p = \text{diag}(100), K_v = \text{diag}(5)$
Force Observer Gains	$K_{v_e} = \text{diag}(2), K_F = \text{diag}(2)$
Omega Controller Gains	$K_1 = \text{diag}(1000)$
Torque Observer Gains	$K_2 = \text{diag}(1000), l_1^p = l_1^q = l_1^r = 500, l_2^p = l_2^q = l_2^r = 2000, \varepsilon = 1 \times 10^{-6}$
Manipulator Controller Gains	$K_p = \text{diag}(100), K_d = \text{diag}(20)$

Table 4.1: Simulation parameters.

matrix R_d . This rotation matrix is computed on the PX4 firmware, which is the physical controller device mounted on the UAV. By setting an x_d, y_d values a R_d is returned as an command input. However, in this work, the controller is only considered for the angular velocity. This simulation is divided into the following steps.

1. For $0 \leq t \leq 3\text{s}$ $\rightarrow z_d = 3\text{m}$
2. For $4 \leq t \leq 5\text{s}$ the manipulator joint will move to $q_{1_d} = 40^\circ$. For $7 \leq t \leq 8\text{s}$ the manipulator joint will move to $q_{2_d} = 30^\circ$. In $t = 10\text{s}$ and $t = 11\text{s}$ q_{2_d} and q_{3_d} will return to home respectively. Simulating the trajectory of a manipulator routine.
3. On $t = 10\text{s}$ the force is increased simulating the manipulator grabbing an object with a weight of 100gr.

During the simulations the actual rotation is computed using the exponential coordinates formula $R = R(0)e^{t\Omega}$ which is solved via MATLAB.

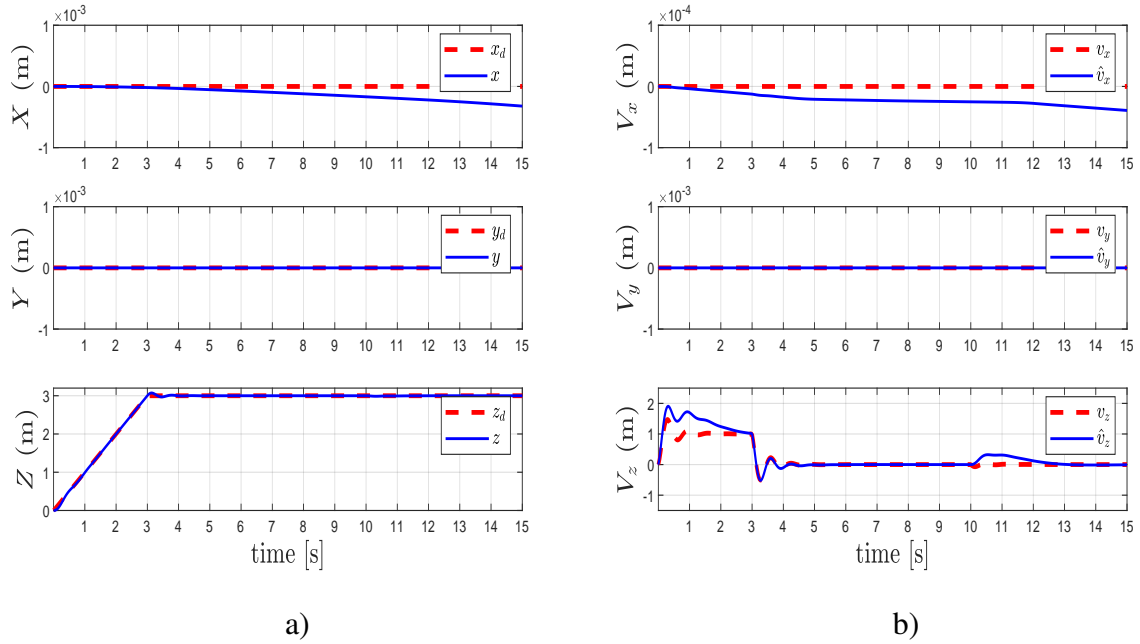


Figure 4.1: a) position and desired position over time. b) linear velocity value and the estimated linear velocity.

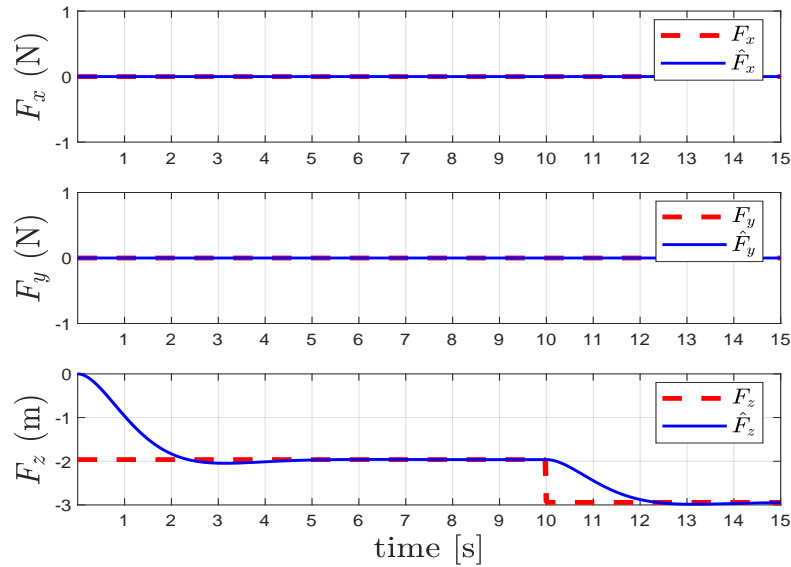


Figure 4.2: Force estimation with a load variation at $t = 10$ s.

In figure 4.1 it can be appreciated how the position controller makes the aerial vehicle follow the desired z trajectory and asymptotically converges as the state observer as well. On

$t = 10$ in figure 4.1.b) a velocity disturbance in the z axis can be appreciated showing how the manipulator mass variation affects the UAV and how this disturbance is being compensated during a time interval. This compensation is done by considering the estimated force, shown in figure 4.2 in the controller.

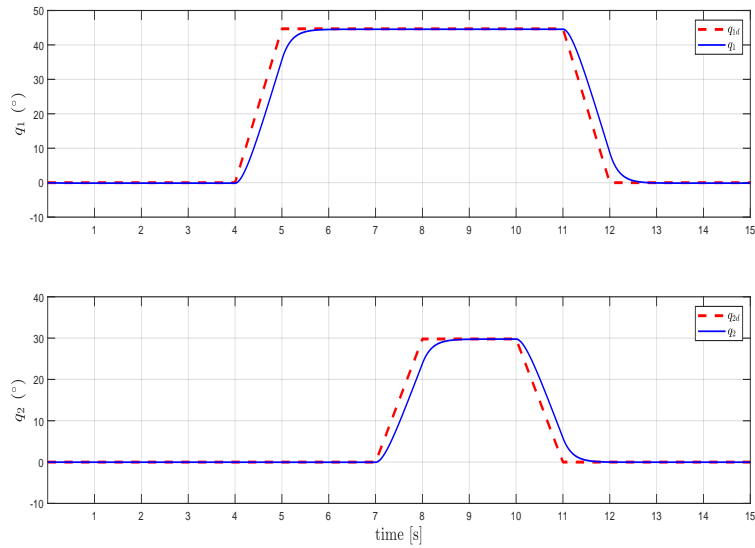


Figure 4.3: Manipulator controller of each joint with a desired joint angle q_d versus the computed q .

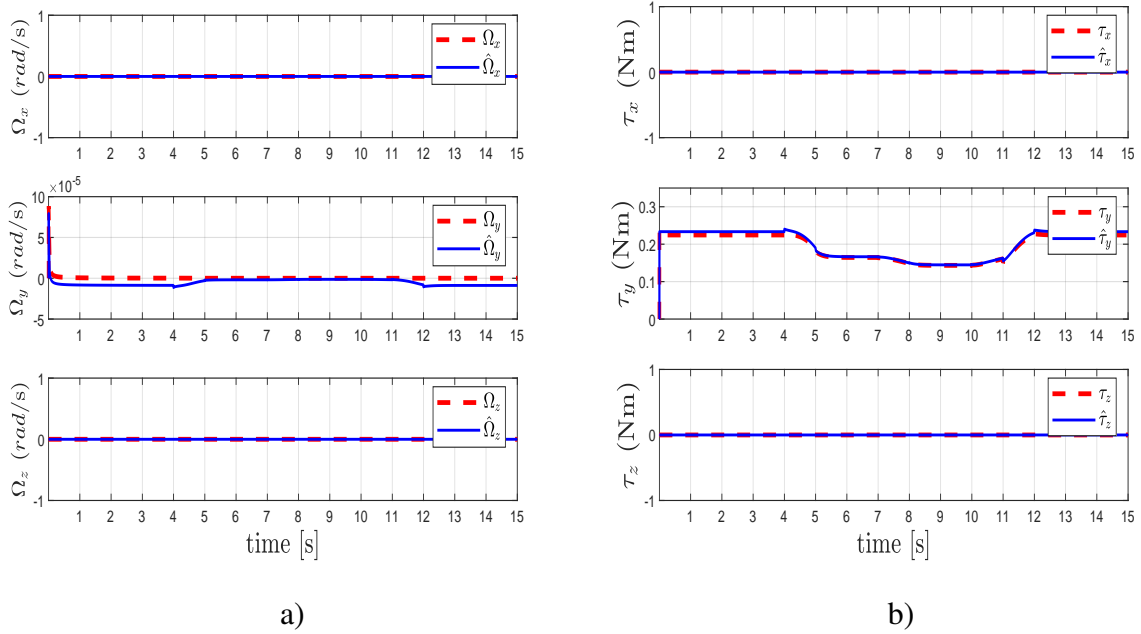


Figure 4.4: a) Torque estimation and b) Ω state estimation.

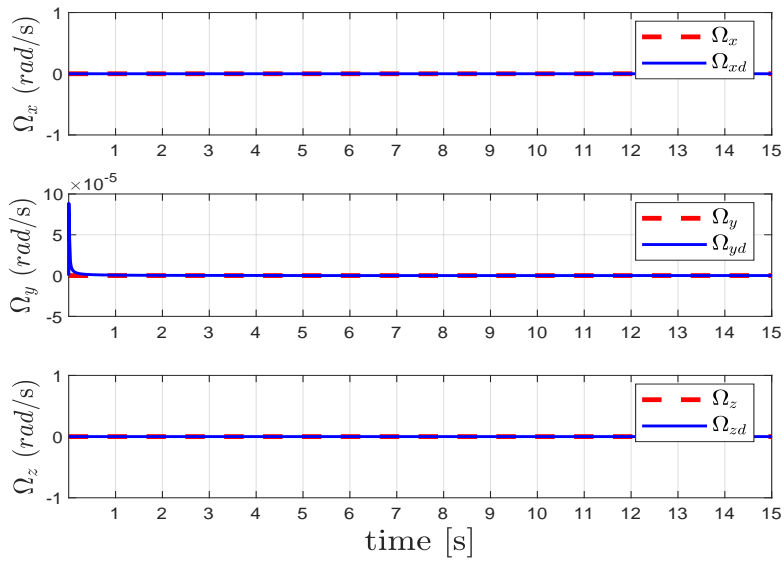


Figure 4.5: Angular velocity value versus the desired one.

In figure 4.3 the PD controller applied makes each joint of the manipulator follow the desired joint angle. The torque disturbance produced by the manipulator can be shown in figure

4.4.a) and 4.4.b) where the estimated ω and the estimated τ are affected while the manipulator is moving. Also the rotation is affected in figure 4.5.b) however using the estimated τ in the attitude controller makes that this disturbance can be compensated. In figure 4.5 it can be shown that the angular velocity converges to the desired value.

4.2 Attitude

For this simulation the desired angular velocity is proposed as a nonlinear varying function to show system stability when "aggressive" rotations have to be made. The desired angular velocities are proposed as the time varying functions $\Omega_x = 0.05\sin(\pi t)$, $\Omega_y = 0.05\cos(\pi t)$, $\Omega_z = 0$

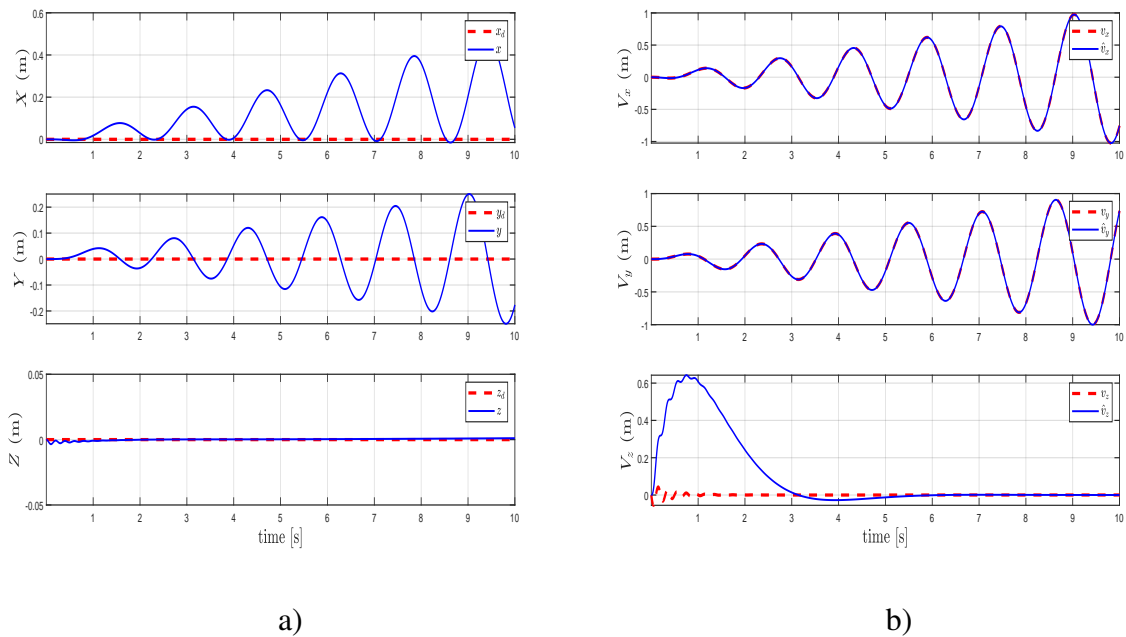


Figure 4.6: a) actual position in the x, y coordinates and the desired and the actual z_d, z position. b) actual linear velocity and the estimated. The position and velocity are caused by the rotation of the vehicle.

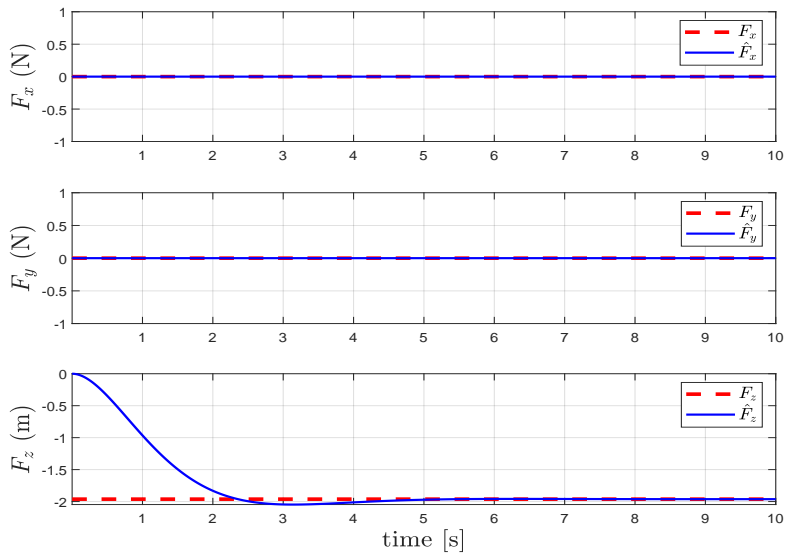
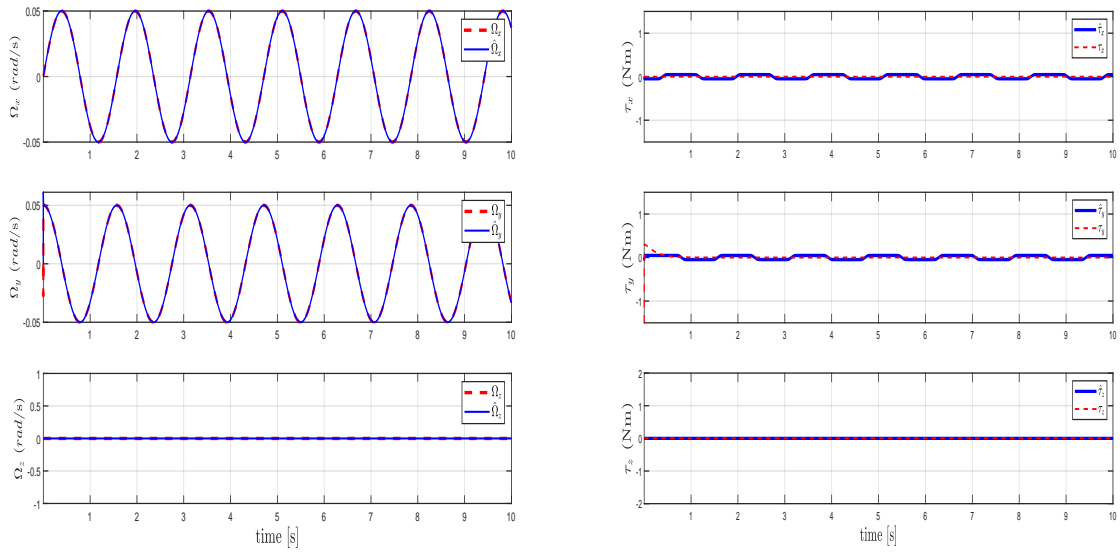


Figure 4.7: Force estimation

In figure 4.6.a) it can be appreciated how the angular velocity, that leads to a rotation, affects the position in the x,y axis. There's a relationship between these two elements that can be taken to get to a desired position in the x,y plane from the desired rotation and angular velocity. In figure 4.6.b) it can be appreciated how the estimated velocity tracks the velocity. Oscillations in the V_Z plot at $t = 1$ show the change in the direction of the thrust force due to the rotation of the vehicle. Force estimation is shown in figure 4.7 where only the force in z is being affected due to the weight of the robotic arm.



(a)

(b)

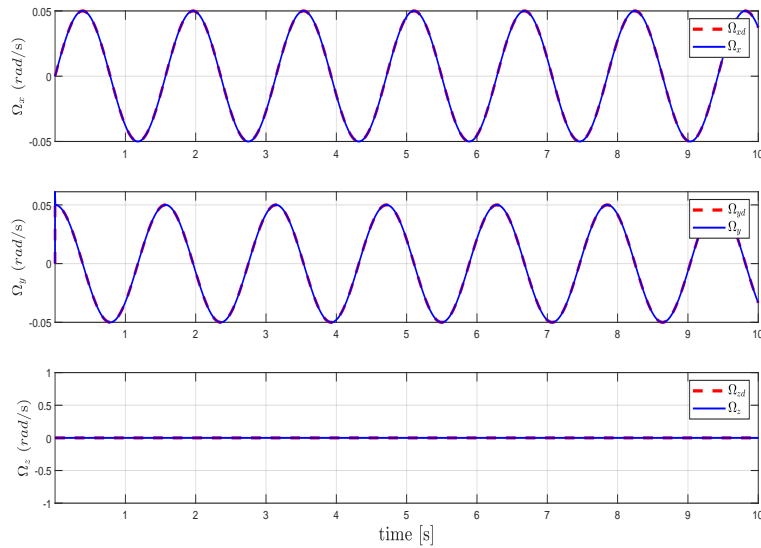
Figure 4.8: Torque and Ω estimation.

Figure 4.9: Actual angular velocity versus the desired one.

Figure 4.8a) shows the estimated angular velocity and the actual where it can be seen that the difference between both is minimum. Therefore in figure 4.8b) it can be appreciated that error in the estimated torque is also minimum however oscillations appear due to the oscillat-

ing behavior of the angular velocity. In figure 4.9 the angular velocity controller makes the input τ to track the desired angular velocity value.

4.3 Drone simulation

To visualize the rotations of this simulation the flight of the vehicle was simulated and a set of screenshots were taken to show the vehicle orientation in different time intervals. To have a better visualization the unit vectors of the body-fixed frame were displayed to visualize the orientation that the vehicle has. The simulation in figures 4.11 is made by plotting circles and vectors for drawing the quadrotor.

Figure 4.10 shows the position x, y, z of the Aerial Manipulator during the simulation time. The robotic arm is plotted using the direct kinematic equations. All the plotted points are rotated into the actual rotation of the vehicle in that time interval.

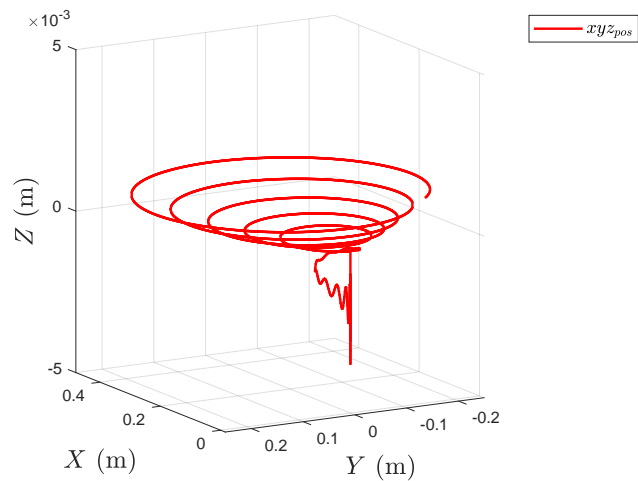


Figure 4.10: Trajectory made by the Aerial Manipulator during the simulation time $0 \leq t \leq 10s$.

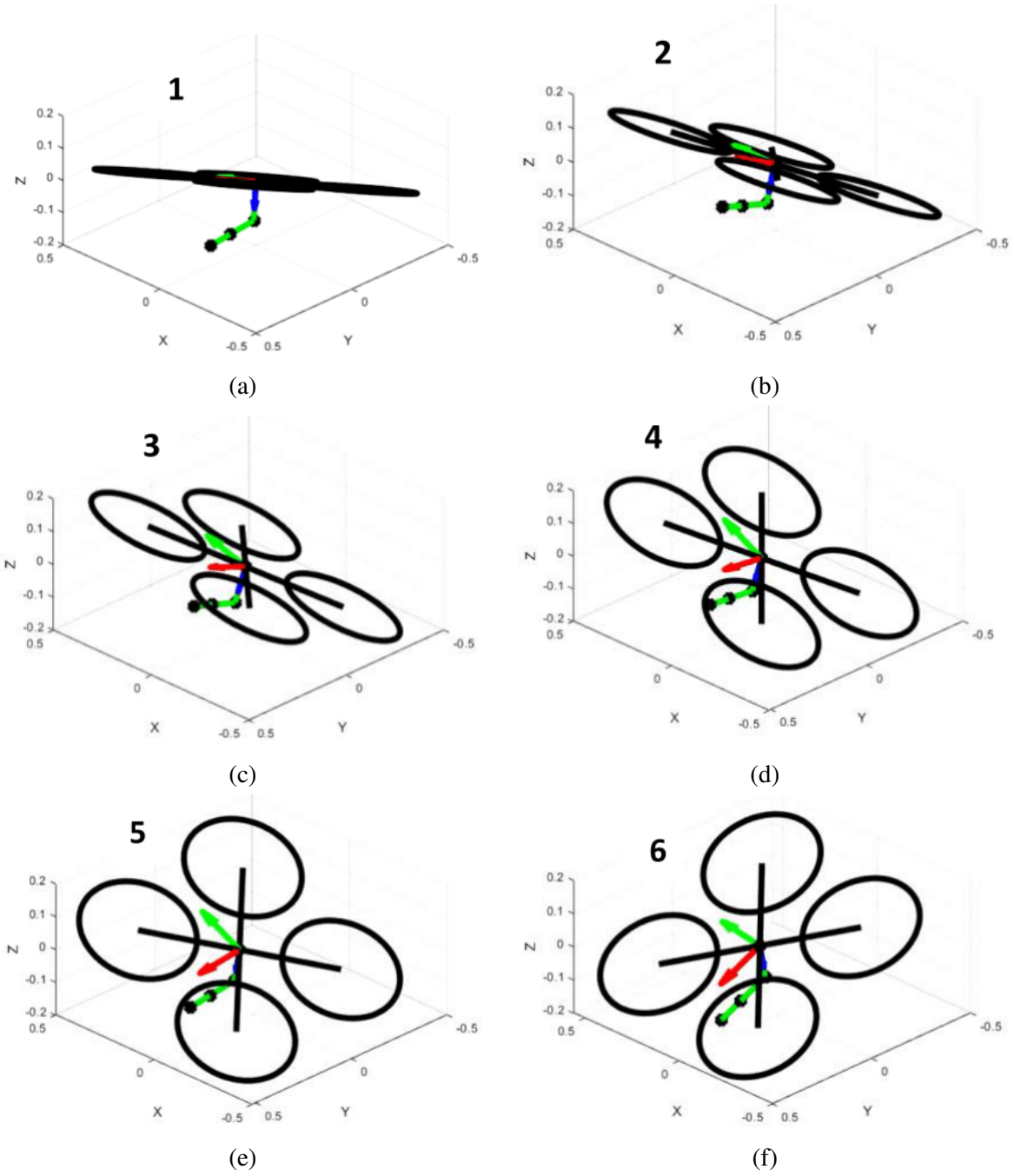


Figure 4.11: All the rotations of the vehicle during the simulation time $0 \leq t \leq 10s$.

CHAPTER 5

Conclusions

In this work, a position and attitude controller for an Aerial Manipulator is proposed using state estimators for the Torque and Force produced by the manipulator. The angular velocity controller is proved to be effective whether in a mathematical form or simulated results. The position controller is also proved to be effective. The estimated disturbances of torque and force were effectively estimated. As in references from [4] to [16] the stability has been proven by making a comparison between the desired input values and the simulation output values and also by a mathematical proof such as using the Lyapunov stability method. In sections 4.1,4.2 each figure shows that the input desired value and the simulation output value are almost equal. Also the stability can be appreciated for the controlled and estimated variables because the output value gets closer to the desired value during a time interval t . Nevertheless in section 3.5 the mathematical proof is shown for the position, attitude and estimations. It has been concluded that, in this case, estimators are not only capable to estimate disturbances caused by the manipulator but any external disturbance such as wind that produce disturbances on the vehicle's center of mass, the observers will help the controller to compensate for these disturbances and therefore stability will not be only in charge of the po-

sition errors or angular velocity errors. Another conclusion related to the position controller is the fact that the Thrust force magnitude depends directly on the z component. Otherwise, the vehicle will not get to the desired altitude therefore it might not be necessary to consider, in further work, the x, y position errors to compute the thrust force magnitude. Further work is to modify the attitude control law so the vehicle tracks the desired rotation. Once this is done and proved the next step would be to propose a function that relates the desired rotation with a desired x, y position.

Bibliography

- [1] M. Sivakumar and N. M. TYJ, “A literature survey of unmanned aerial vehicle usage for civil applications,” *Journal of Aerospace Technology and Management*, vol. 13, 2021.
- [2] C. A. Kozera, “Military use of unmanned aerial vehicles—a historical study,” *Safety & Defense*, vol. 4, no. 1, pp. 17–21, 2018.
- [3] F. Ruggiero, V. Lippiello, and A. Ollero, “Aerial manipulation: A literature review,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.
- [4] Z. A. Ali and X. Li, “Modeling and controlling of quadrotor aerial vehicle equipped with a gripper,” *Measurement and Control*, vol. 52, no. 5-6, pp. 577–587, 2019.
- [5] R. Jiao, W. Chou, R. Ding, and M. Dong, “Adaptive robust control of quadrotor with a 2-degree-of-freedom robotic arm,” *Advances in Mechanical Engineering*, vol. 10, no. 8, p. 1687814018778639, 2018.
- [6] J. Alvarez, N. Marchand, J. Guerrero-Castellanos, J. Tellez-Guzman, J. Escareno, and M. Rakotondrabe, “Rotorcraft with a 3dof rigid manipulator: Quaternion-based modeling and real-time control tolerant to multi-body couplings,” *International Journal of Automation and Computing*, vol. 15, pp. 547–558, 10 2018.

- [7] R. Mo, H. Cai, and S.-L. Dai, “Unit quaternion based attitude control of an aerial manipulator**this work was supported in part by the national natural science foundation of china under grant 61803160, in part by the foshan science and technology innovation team special project under grant 2018it100322, in part by the science and technology program of nansha district under grant 2017gg006. the corresponding author is he cai.” *IFAC-PapersOnLine*, vol. 52, no. 24, pp. 190–194, 2019, 5th IFAC Symposium on Telematics Applications TA 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896319323080>
- [8] R. Naldi, A. Macchelli, N. Mimmo, and L. Marconi, “Robust control of an aerial manipulator interacting with the environment**this work has been partially supported by the european project airborne (ict 780960).” *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 537–542, 2018, 2nd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896318310905>
- [9] J. Á. Acosta, C. R. de Cos, and A. Ollero, “Accurate control of aerial manipulators outdoors. a reliable and self-coordinated nonlinear approach,” *Aerospace Science and Technology*, vol. 99, p. 105731, 2020.
- [10] E. Yilmaz, H. Zaki, and M. Unel, “Nonlinear adaptive control of an aerial manipulation system,” in *2019 18th European Control Conference (ECC)*, 2019, pp. 3916–3921.
- [11] H. Beikzadeh and G. Liu, “Trajectory tracking of quadrotor flying manipulators using 11 adaptive control,” *Journal of the Franklin Institute*, vol. 355, no. 14, pp. 6239–6261, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0016003218304149>
- [12] M. Vahdanipour and M. Khodabandeh, “Adaptive fractional order sliding mode control for a quadrotor with a varying load,” *Aerospace Science and Technology*, vol. 86, 03 2019.

- [13] M. Ballesteros-Escamilla, D. Cruz-Ortiz, I. Chairez, and A. Luviano-Juárez, “Adaptive output control of a mobile manipulator hanging from a quadcopter unmanned vehicle.” *ISA transactions*, 2019.
- [14] H. Bonyan Khamseh, S. Ghorbani, and F. Janabi-Sharifi, “Unscented kalman filter state estimation for manipulating unmanned aerial vehicles,” *Aerospace Science and Technology*, vol. 92, pp. 446–463, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1270963818315931>
- [15] A. Khalifa and M. Fanni, “Experimental implementation of a new non-redundant 6-dof quadrotor manipulation system,” *ISA Transactions*, vol. 104, pp. 345–355, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0019057820301890>
- [16] K. Gkountas, D. Chaikalis, and A. Tzes, “Force control design for a robot manipulator attached to a uav,” *IFAC-PapersOnLine*, vol. 51, no. 30, pp. 548–553, 2018, 18th IFAC Conference on Technology, Culture and International Stability TECIS 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896318329410>
- [17] F. Ruggiero, V. Lippiello, and A. Ollero, “Introduction to the special issue on aerial manipulation,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2734–2737, 2018.
- [18] J. Mendoza-Mendoza, V. J. Gonzalez-Villela, C. Aguilar-Ibañez, M. S. Suarez-Castañon, and L. Fonseca-Ruiz, “Snake aerial manipulators: A review,” *IEEE Access*, vol. 8, pp. 28 222–28 241, 2020.
- [19] M. Ryll, D. Bicego, and A. Franchi, “A Truly Redundant Aerial Manipulator exploiting a Multi-directional Thrust Base,” in *12TH IFAC SYMPOSIUM ON ROBOT CONTROL (SYROCO 2018)*, Budapest, Hungary, Aug. 2018, p. 6p. [Online]. Available: <https://hal.laas.fr/hal-01846466>

- [20] V. Nayak, C. Papachristos, and K. Alexis, “Design and control of an aerial manipulator for contact-based inspection,” *ArXiv*, vol. abs/1804.03756, 2018.
- [21] D. Tzoumanikas, F. Graule, Q. Yan, D. Shah, M. Popovic, and S. Leutenegger, “Aerial manipulation using hybrid force and position nmpc applied to aerial writing,” *ArXiv*, vol. abs/2006.02116, 2020.
- [22] X. Qi, J. Wu, and J. Pan, “A compound controller of an aerial manipulator based on maxout fuzzy neural network,” *Complexity*, vol. 2020, pp. 1–10, 2020.
- [23] C. Kanellakis and G. Nikolakopoulos, “Guidance for autonomous aerial manipulator using stereo vision,” *Journal of Intelligent & Robotic Systems*, vol. 100, 12 2020.
- [24] K. Ogata, *Ingenieria de Control Moderna*, 3rd ed. PRENTICE HALL HISPANOAMERICANA, 1998.
- [25] K. Valavanis, *Advances in Unmanned Aerial Vehicles*. Springer, 2007.
- [26] M. Ardema, “Newton-euler dynamics,” 01 2005.
- [27] K. Lynch and F. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge Univeristy Press, 2017.
- [28] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. Society for Industrial and Applied Mathematics, 2002. [Online]. Available: <https://epubs.siam.org/doi/abs/10.1137/1.9780898719185>
- [29] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.
- [30] K. Hangos, J. Bokor, and G. Szederkényi, *Analysis and Control of Nonlinear Process Systems*, 01 2004.
- [31] G. Ellis, *Observers in Control System A Practical Guide*, 1st ed. Elsevier Science, 2002.
- [32] G. Besançon, “Nonlinear observers and applications,” 2007.

- [33] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 2nd ed. Wiley, 2020.
- [34] A. Barrientos, *Fundamentos de Robótica*. McGraw-Hill, 2007.
- [35] A. Datta, M.-T. Ho, and S. Bhattacharyya, *Structure and Synthesis of PID Controllers*, 01 2000.
- [36] C. R. Rodolfo Verdín, Germán Ramírez and G. Flores, “Teleoperated aerial manipulator and its avatar. communication, system’s interconnection, and virtual world,” *International Conference on Unmanned Aircraft Systems*, 2019.
- [37] H. Nijmeijer and T. Fossen, *New Directions in Nonlinear Observer Design*, 05 1999, vol. 244.

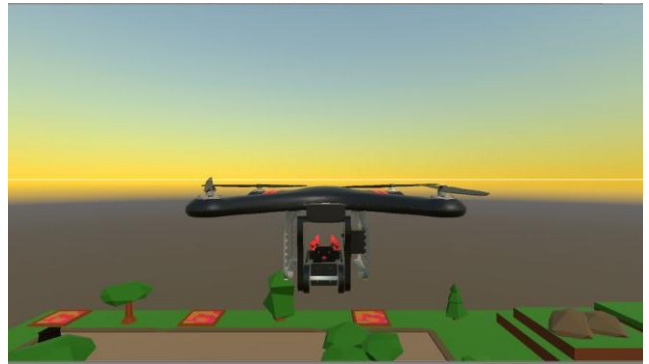
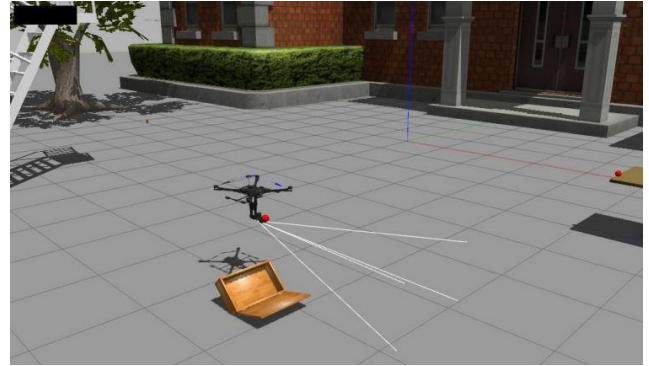
CHAPTER 6

Annex

In the conference paper [36], named "Teleoperated aerial manipulator and its avatar. Communication, system's interconnection, and virtual world" and published in the 2019s International Conference of Unmanned Aircraft Systems, the author of this work made a contribution. In that work, an aerial vehicle is controlled without considering any torque and force disturbance. The paper is more focused on the development of a virtual world interface where the user is capable to teleoperate an aerial vehicle using a virtual reality device. The simulation is made by using the PX4 firmware to simulate the micro-controller, Gazebo simulator is used to be the real-world representation and Unity 3D represents the virtual world that the user is watching via the virtual reality headset like in figure 6.1.



a)



b)

Figure 6.1: a) the operator using the HTC vive device during the simulation. The above figure in column b) shows the Gazebo simulation and the second image shows the aerial vehicle in Unity 3D [36].

The simulation results include a physical user who is moving the vehicle and the arm using an HTC Vive device to complete a routine. Information from each software program is exchanged via MAVROS messages and the output data proves the system stability of the controller and the successful communication of all the programs. The structure of the whole communication system is shown in figure 6.2. The author of this work contributed partially to the communication protocols and Gazebo simulation, he also programmed the control equations in the PX4 firmware.

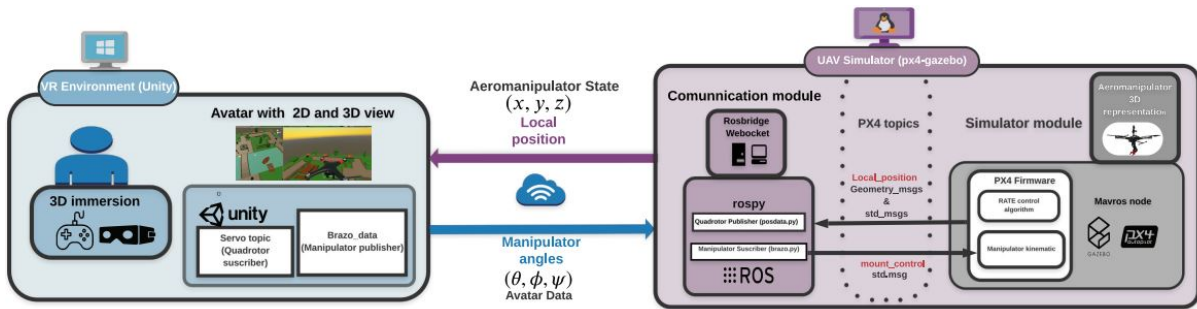


Figure 6.2: Communication system structure of the virtual reality environment [36].

The Aerial Manipulator, which belongs to the CIO's Laboratory of Perception and Robotics in charge of the researcher Gerardo Flores Colunga, is shown in figure 6.3. This Aerial Manipulator is the model used to get the physical parameters, such as weight, inertia, or link length, that are used to make whether this work and paper simulations.



Figure 6.3: Physical aerial manipulator belonging to the Laboratory of Perception and Robotics [36].