Coupled Control of Satellite Platform and Robot Arm for Space Robotics Applications

Sofia Fernandes
sofiamgsfernandes@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

June 2019

Abstract

In this thesis, the capture phase of the *e.deorbit mission* (the first ADR mission) is explored. The chaser satellite tracks the target satellite, with a robotic manipulator. The chaser must maintain its pose during capture. In order to simulate the chaser dynamics and kinematics an open chain model was implemented - the TITOP model. The manipulator produces disturbances on the satellite and vice versa, therefore coupled control is considered. During control design, a fixed structured H_∞ control approach was selected. Finally, robust stability and performance were analysed and directions for future work were provided.

Keywords: Robotic Arm, TITOP Model, Coupled Control, Fixed Structured H_∞ Control, Robust Control

1. Introduction

The number of debris in space has become a great concern to operational spacecraft. Cases of uncontrolled re-entry into the atmosphere become an issue of public and property safety.

ESA is pursuing the first Active Debris Removal mission ever conducted and it is called *e.deorbit*. In this mission, a large spacecraft named ENVISAT is to be removed from its orbit. Due to its large size, it has a high risk of collision. The tumbling motion of this satellite makes it an even more challenging capture.

In the context of this dissertation, the capture of the target satellite with a redundant robotic arm attached to a chaser satellite is analysed.

The desired pose of the end effector (EE) of the manipulator is achieved by controlling joint motors using a control system. Due to dynamic coupling, the motion of the EE affects the motion of the satellite and vice-versa. In order to avoid undesirable disturbances on the manipulator motion, another approach is to turn off the Attitude Determination and Control System; these systems are called free-floating robots [11].

However, in this mission, the pose of the satellite is required to remain stable. Thus, the controller can be divided into two controllers: one for the satellite and another for the manipulator. Therefore, free-flying mode is considered.

Dynamic control of space robots such as the chaser satellite can be divided into two levels. The first is concerned with coordinate transformations. The second level requires computation of the required force and torque control commands. In [4],

an adaptive control is used for the manipulator controller. This controller is valid for non linear and time-varying plants. In [5], a Linear Quadratic Regulator (LQR) and an H_∞ controllers were implemented for a free-floating space manipulator with two arms. It was concluded that, in the presence of disturbances and parametric uncertainties, only the H_∞ controller had satisfactory performance. In [11], strategies of control in the presence of angular momentum, for a free-floating space manipulator, are presented. Point-to-point and tracking control in joint space are explored using a PD controller.

The fixed structure H_∞ control problem is explored in [8]. With this approach, it is possible to obtain a robust controller, and with low computational burden - since the control structure is previously defined.In [8], a PD-like structure was used, which resulted in a good compromise between implementation simplicity and tracking performance.

2. Relative Dynamics and Kinematics - The TITOP Model

In [2], an open chain model, where forces and torques are propagated from element to element, is presented. In this model, each node corresponds to a body and each edge to a joint between two bodies (tree terminology). Each element can be represented as a multi-input multi-output transfer with two channels: **direct dynamic model** and **inverse dynamic model**. The first one is at the connection point between some substructure and its parent (previous substructure), where accelerations are inputs and forces and torques are outputs. The second one is at the connection point between the substructure and its child (next substructure); here, forces and torques are inputs and accelerations are outputs. This model is called Two Input Two-Output Port (TITOP) model.

Such a model allows us to build the dynamic model of an entire spacecraft by assembling a TITOP model to each substructure. This model also enables the designer to isolate parameters to be optimized. In addition, the TITOP model allows us to model complex multi-body systems, such as a satellite with a robotic arm.

Consider 3 different bodies, body $\mathcal{A}$, its parent body $\mathcal{P}$, and body $\mathcal{C}$ the child of $\mathcal{A}$. Each body has 2 connection points, P and C. The former represents the connection point between some body and its parent, and the latter, the connection point between some body and its child. Point A is set to the substructure CoM.

The linear acceleration $\vec{a}_C^{\mathcal{A}}$ corresponds to an output to the indirect dynamic channel and it is obtained by

$$\vec{a}_C^{\mathcal{A}} = \vec{a}_P^{\mathcal{A}} + \vec{\omega}_P^{\mathcal{A}} \times \left(\vec{\omega}_P^{\mathcal{A}} \times \vec{r}\right) + \dot{\vec{\omega}}_P^{\mathcal{A}} \times \vec{PC} \quad (1)$$

where $\vec{a}_P^{\mathcal{A}} \in \mathbb{R}^3$ is the linear acceleration at point P, measured in frame $\mathcal{F}_{\mathbb{A}}$; $\vec{\omega}_P^{\mathcal{A}} \times \left(\vec{\omega}_P^{\mathcal{A}} \times \vec{r}\right)$ is the centrifugal acceleration, $\dot{\vec{\omega}}_P^{\mathcal{A}}$ is the angular acceleration measured in frame $\mathcal{F}_{\mathcal{A}}$.

$$\begin{cases} \vec{a}_C^{\mathcal{A}} = \mathbf{R}\vec{a}_P^{\mathcal{C}} \\ \dot{\vec{\omega}}_C^{\mathcal{A}} = \mathbf{R}\dot{\vec{\omega}}_P^{\mathcal{C}} \end{cases} . \quad (2)$$

In eq. (2), $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the Direction Cosine Matrix (DCM) that represents the transformation from frame $\mathcal{F}_{\mathcal{A}}$ to frame $\mathcal{F}_{\mathcal{C}}$. Reference frame $\mathcal{F}_{\mathcal{A}}$ is centred in $\mathcal{A}$ and rotating with an angular rate of $\vec{\omega}_C^{\mathcal{A}}$, and the latter is centred in body $\mathcal{C}$ and with an angular rate of $\vec{\omega}_P^{\mathcal{C}}$. If both rotating frames are rotating with the same inertial angular velocities, then $\boldsymbol{R}$ is constant.

In order to simplify this model, second order terms are neglected which results in $\boldsymbol{a_C} = \boldsymbol{\tau_{CP}}\boldsymbol{a_P}$, where $\boldsymbol{\tau_{CP}} \in \mathcal{R}^{6\times6}$ is the kinematic model and it is given by

$$\boldsymbol{\tau_{CP}} = \begin{bmatrix} \mathbf{I_3} & (\boldsymbol{CP}\times) \\ \mathbf{0_{3\times3}} & \mathbf{I_3} \end{bmatrix},$$

and $\boldsymbol{a_\star} = \begin{bmatrix} \vec{a}_\star^{\mathcal{A}} & \dot{\vec{\omega}}_\star^{\mathcal{A}} \end{bmatrix}^T$.

Using the Newton's Second Law $\vec{F} = m\vec{a}$ and assuming that second order terms of linear/angular velocity are small enough to be neglected, the dynamic set of equations introduced in the model correspond to

$$\begin{cases} \vec{F}_{\mathcal{A}/\mathcal{P}} = \vec{F}_{\mathfrak{e}/\mathcal{A}} - m^{\mathcal{A}} \left(\vec{a}_P^{\mathcal{A}} + \dot{\vec{\omega}}_P^{\mathcal{A}} \times \vec{PA} \right) \\ \vec{T}_{\mathcal{A}/\mathcal{P},P} = \vec{T}_{\mathfrak{e}/\mathcal{A},C} - \vec{CP} \times \vec{F}_{\mathfrak{e}/\mathcal{A}} + m^{\mathcal{A}} \vec{AP} \times \vec{a}_P^{\mathcal{A}} - \\ J_A^{\mathcal{A}} \dot{\vec{\omega}}_P^{\mathcal{A}} + m^{\mathcal{A}} \dot{\vec{\omega}}_P^{\mathcal{A}} \times \vec{AP} \times \vec{AP} \end{cases},$$

where $\vec{F}_{\bullet/\star} \in \mathcal{R}^3$ is the force that body $\bullet$ applies on $\bullet$. The torque that body $\bullet$ applies on $\bullet$, at point x is $\vec{T}_{\bullet/\star,x} \in \mathcal{R}^3$. Note that $\vec{F}_{\mathcal{A}/\mathcal{P}} = -\vec{F}_{\mathcal{P}/\mathcal{A}}$ and $\vec{T}_{\mathcal{A}/\mathcal{P},P} = -\vec{T}_{\mathcal{P}/\mathcal{A},P}$. $J_A^{\mathcal{A}} \in \mathcal{R}^{3\times3}$ is the inertia matrix of body $\mathcal{A}$, measured around point A.

Thus, we have

$$\boldsymbol{F_{\mathcal{A}/\mathcal{P},P}} = \begin{bmatrix} \boldsymbol{\tau_{CP}^T} & \boldsymbol{-D_{AP}} \end{bmatrix} \begin{bmatrix} \boldsymbol{F_{\mathfrak{e}/\mathcal{A},C}} \\ \mathbf{a_P} \end{bmatrix},$$

where $\boldsymbol{F_{\mathcal{A}/\mathcal{P}}} = [\vec{F}_{\mathcal{A}/\mathcal{P}}, \vec{T}_{\mathcal{A}/\mathcal{P},P}]^T$ and the dynamic model is given by [2]

$$\mathbf{D_{AP}} = \begin{bmatrix} m^{\mathcal{A}}\mathbf{I_3} & m^{\mathcal{A}}(\boldsymbol{AP}\times) \\ -m^{\mathcal{A}}(\boldsymbol{AP}\times) & \mathbf{J_A} m^{\mathcal{A}}(\boldsymbol{AP}\times)^2 \end{bmatrix}.$$

It should be noted that external commands are applied at the joints of the robotic arm, in order to control the position of its End Effector (EE). These external commands are provided as torques, at each joint.

We shall consider that joint i is positioned at point P of segment i (where $i \in [1, n]$ and n is the number of joints). The external torque $\vec{C}_m \in \mathbb{R}^3$ at the joint of some body $\mathcal{A}$ can be written as $\vec{C}_m = -\hat{r}_c \cdot \vec{T}_{\mathcal{A}/\mathcal{P},P}$, where $\hat{r}_c \in \mathbb{R}^3$ is the unit vector that represents the axis of rotation of the revolute joint (1 degree of freedom), w.r.t. the body frame corresponding to $\mathcal{A}$ [1, 2].

As a consequence of this extra torque, the angular acceleration also changes. Thus, after applying the command $\vec{C}_m$ the angular acceleration at point C of the same body becomes $\dot{\vec{\omega}}_C^{\mathcal{A}} = \dot{\vec{\omega}}_{P_b}^{\mathcal{A}} + \ddot{\vec{\theta}}$, where $\dot{\vec{\omega}}\mathcal{A}_{P_b} \in \mathbb{R}^3$ is the angular acceleration at point P of segment $\mathcal{A}$ before the command.

Given that the external torque C_m is applied at point P, this is the center of rotation when considering this torque alone.

Thus, the linear acceleration is expressed by

$$\vec{a}_j^{\mathcal{A}} = \vec{a}_P^{\mathcal{A}} + \left(\vec{\omega}_{P_b}^{\mathcal{A}} + \dot{\vec{\theta}} \right) \times \left[(\vec{\omega}_{P_b}^{\mathcal{A}} + \dot{\vec{\theta}}) \times \vec{r} \right] + \ldots \\ \cdots + \left(\dot{\vec{\omega}}_{P_b}^{\mathcal{A}} + \ddot{\vec{\theta}} \right) \times \vec{r}, \quad (3)$$

One should take into account that when transferring the linear and angular accelerations from some body to its rotating child node, the transformation matrix $\boldsymbol{R}$ is now **not** constant - θ is variable. Although the TITOP model is based on linear dynamics and kinematics equations, this model cannot be considered as linear due to the time variant DCMs corresponding to the manipulator segments.

The acceleration at point C of body $\mathcal{A}$ is, now, given by:

$$\boldsymbol{a_C} = \begin{bmatrix} \boldsymbol{\tau_{CP}} & (\boldsymbol{CP}\times)\boldsymbol{r_c} \end{bmatrix} \begin{bmatrix} \boldsymbol{a_P} \\ \ddot{\theta} \end{bmatrix}$$

and the forces and torques actuating on point P correspond to

$$\boldsymbol{F_{\mathcal{A}/\mathcal{P},P}} = \begin{bmatrix} \boldsymbol{\tau_{CP}^T} & \boldsymbol{-D_{AP}} & (\boldsymbol{CP}\times)\boldsymbol{r_c} \end{bmatrix} \begin{bmatrix} \boldsymbol{F_{\mathfrak{e}/\mathcal{A},C}} \\ \mathbf{a_P} \\ \ddot{\theta} \end{bmatrix},$$

Finally, the joint command is given by

$$Cm = \begin{bmatrix} \begin{bmatrix} \boldsymbol{r_c^T}(\boldsymbol{CP}\times)\boldsymbol{r_c} & -\mathbf{r_c^T r_c} \end{bmatrix} \cdots \\ \begin{bmatrix} m^{\mathcal{A}} \boldsymbol{r_c^T}(\boldsymbol{AP}\times)\boldsymbol{r_c} & \mathbf{r_c^T J_A r_c} - m^{\mathcal{A}} \boldsymbol{r_c^T}(\boldsymbol{AP}\times)^{\mathbf{2}}\boldsymbol{r_c} \end{bmatrix} \cdots \\ \mathbf{r_c^T J_A r_c} - m^{\mathcal{A}} \boldsymbol{r_c^T}(\boldsymbol{AP}\times)^{\mathbf{2}}\boldsymbol{r_c} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{F_{\mathfrak{e}/\mathcal{A},C}} \\ \mathbf{a_P} \\ \ddot{\theta} \end{bmatrix}.$$

2.1. Model Implementation

First, one ought to divide the entire structure of the spacecraft into smaller substructures.

The chaser was divided into the following substructures: hub (main body, where the CoM of the spacecraft is located), fuel tank, oxidizer tank, base of the robotic arm, and 7 segments of the robotic arm, each with a revolute joint. The chaser platform is composed by all substructures except for the manipulator segments.

Each substructure is represented by a TITOP block with two channels. All of these elements, except for the hub, have an inverse and a direct dynamic channels. The hub is a special element because it is represented by a TITOP block with two inverse dynamic channels (forces and torques applied by the **child** and the **parent** are inputs). Another special feature is that it is the only body with more than 1 Child block. The children of the hub are the fuel tank, the oxidizer, and the base of the robotic arm. The parent of the hub is considered to be its surroundings. Therefore, the forces and torques applied on this body are environmental disturbances, as well as the resulting forces and torques provided by the thrusters. For this, the connection point P, *i.e.* the point where its parent applies the commands, is set to the hub CoM (point A), which is constant in the body frame.

An important feature of the TITOP model is the freedom in the initialization of the blocks. For a robotic arm, a systematic choice of reference frames is the *Denavit Hartenberg Representation*, also called DH convention. Besides the consistency in the reference frames definition, another advantage is its inherent computational efficiency. This is explained by the fact that the only variable parameter of a DCM of some segment i is θ_i. The DH convention considered in this project is the modified DH convention.

3. Inverse Dynamics and Kinematics

Some important background to compute the commands forces and torques that actuate on the chaser is presented. First, the Inverse Kinematic Problem (IKP) is introduced, then Dynamics Coupling between the platform and the robotic arm is presented.

3.1. Inverse Kinematic Problem

Linear and angular velocities can be related to joints angular rate by $\dot{\boldsymbol{x}} = \boldsymbol{J}\dot{\boldsymbol{\theta}}$, where $\dot{\boldsymbol{x}} \in \mathbb{R}^m$ is, in this case, the linear and angular velocities of the EE, $\dot{\boldsymbol{\theta}} \in \mathbb{R}^n$ represents the joint rates. The Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m\times n}$ maps the spacial velocity of the EE into joints rate [9, 14].

However, the Inverse Kinematic Problem (IKP) is concerned with what joint angles are required to reach the desired pose, using the known angular and linear velocities of the end effector. Hence, we have $\dot{\theta} = \mathbf{J}^{-1}\dot{\mathbf{x}}$.

In the context of this thesis, a robotic arm with 7 revolute joints is considered, in a 6 Degrees of Freedom (DoFs) problem. Therefore, $\boldsymbol{J}$ is not a square matrix and cannot be inverted. Redundant manipulators have more internal DoFs than the required to perform a specified task, *i.e.* $n > m$ and have enough DoFs to perform the desired task while constraining the workspace. Constraints can be used to execute smooth trajectories as well as to avoid solutions with no physical meaning. The IKP for a redundant manipulator has multiple solutions. If proper constraints are enforced, a more efficient EE trajectory could be obtained.

The Jacobian $\boldsymbol{J}$ can be divided into the Jacobian for linear velocity $\boldsymbol{J_v}$ and the Jacobian for angular velocity $\boldsymbol{J_\omega}$: $\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J_v} & \boldsymbol{J_\omega} \end{bmatrix}^T$. Considering the modified DH convention, we have for the ith column of the Jacobian for the angular velocity

$$\boxed{\boldsymbol{J_{\omega_i}} = \boldsymbol{z_i^C}}, \tag{4}$$

and for the ith column of the Jacobian for the linear velocity as

$$\boxed{\boldsymbol{J_{v_i}} = (\boldsymbol{z_i^C}\times)(\boldsymbol{O_{EE}^C} - \boldsymbol{r_i^C})}, \tag{5}$$

where $(\boldsymbol{z_i^C}\times) \in \mathbb{R}^{3\times3}$ is the Skew Symmetric matrix of $\boldsymbol{z_i^C}$.

Since the Jacobian matrix for redundant manipulators cannot be inverted, we can make use of the pseudo-inverse (or Moore-Penrose inverse) of $\boldsymbol{J}$, *i.e.* $\boldsymbol{J}^\dagger \in \mathbb{R}^{n\times m}$, given by

$$\boldsymbol{J}^\dagger = \left(\boldsymbol{J}\boldsymbol{J}^{\boldsymbol{T}}\right)^{-1}\boldsymbol{J}^{\boldsymbol{T}}.$$

when $\boldsymbol{J}$ has full row rank.

The pseudo-inverse calculation method can also be obtained using the Singular Value Decomposition (SVD) of $\boldsymbol{J}$.

According to [9], generating a velocity component in certain directions at the EE and close to a singular posture, requires very hight joint rates, which are not affordable by the manipulator. Hence, the Jacobian damped least square method is proposed, in [6, 7] . Here, a small diagonal term λ is added to the pseudo-inverse. This method only provides an approximate solution to the IKP, but offers satisfactory results regarding singularity avoidance and it is defined by

$$\boxed{\dot{\boldsymbol{\theta}}_{\boldsymbol{min}} = \left(\boldsymbol{J}^{\boldsymbol{T}}\boldsymbol{J} + \lambda^2\boldsymbol{I}\right)^{-1}\boldsymbol{J}^{\boldsymbol{T}}\dot{\boldsymbol{x}}}. \tag{6}$$

In this project, it is considered a constant $\lambda = 0.2$.

Figures 1 and 2 reflect the importance of using the damped least squares method as opposed to the pseudo-inverse, in the IKP. In the pictures the position error at the end effector (EE) is shown, considering a non feasible target point, e.g. where the length of the manipulator is not enough to reach it.

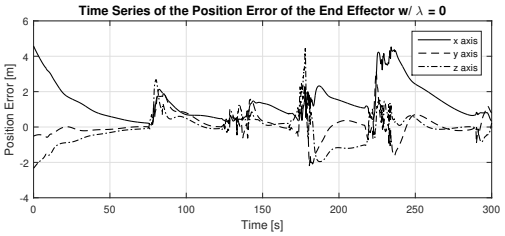

Figure 1: Position error of the EE. The pseudo inverse is used in the IKP.

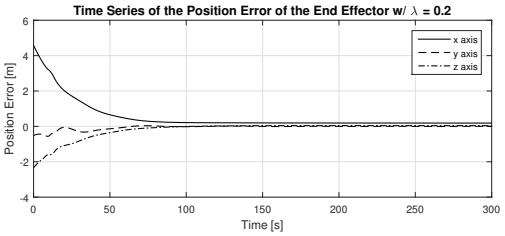

Figure 2: Position error of the EE. The least squares method is used in the IKP.

3.2. Inverse Dynamics

The linearised dynamics of a satellite and the robotic arm can be expressed by

$$\begin{bmatrix} \boldsymbol{H_p} & \boldsymbol{H_{pm}} \\ \boldsymbol{H_{pm}^T} & \boldsymbol{H_m} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{x}}_{\boldsymbol{p}} \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{F_p} \\ \boldsymbol{Cm} \end{bmatrix}. \tag{7}$$

The notation used in (7) is listed below, considering an n-jointed manipulator:

- $\boldsymbol{x_H} \in \mathbb{R}^6$: pose at the hub CoM
- $\boldsymbol{H_p} \in \mathbb{R}^{6\times6}$: platform inertia matrix
- $\boldsymbol{H_{pm}} \in \mathbb{R}^{6\times n}$: coupling inertia matrix, between the platform and the manipulator
- $\boldsymbol{H_m} \in \mathbb{R}^{n\times n}$: manipulator inertia matrix
- $\boldsymbol{F_p} \in \mathbb{R}^6$: external forces and torques actuating on the platform
- $\boldsymbol{Cm} \in \mathbb{R}^n$: external torques actuating on the joints of the robotic arm

Note that the platform is composed by the hub, the fuel and oxidizer tanks, and the base of the manipulator and that gravitational forces are introduced in the control system as disturbances through input $\boldsymbol{F_p}$.

The kinematic relation among the pose of the EE $\boldsymbol{x_{EE}}$, the pose of the hub $\boldsymbol{x_p}$, and joint angles $\boldsymbol{\theta}$ is given by $\dot{\boldsymbol{x}}_{\boldsymbol{EE}} = \boldsymbol{J_m}\dot{\boldsymbol{\theta}} + \boldsymbol{J_p}\dot{\boldsymbol{x}}_{\boldsymbol{p}}$, where $\boldsymbol{J} \in \mathbb{R}^{6\times n}$ is the manipulator Jacobian derived above and $\boldsymbol{J_p} \in \mathbb{R}^{6\times 6}$ is called the platform Jacobian and it is given by

$$\boldsymbol{J_p} = \begin{bmatrix} \boldsymbol{I} & (\boldsymbol{r_{H\rightarrow EE}}\times) \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{8}$$

where $(\boldsymbol{r_{H\rightarrow EE}}\times) \in \mathbb{R}^{3\times 3}$ is the skew symmetric matrix of the vector from the hub CoM to the EE [12].

The inertia matrices presented can be obtained using the Lagrange-Euler Formulation. The manipulator inertia matrix is expressed by

$$\boldsymbol{H_m} = \begin{bmatrix} m_1\boldsymbol{J}_{\boldsymbol{v_1}}^T\boldsymbol{J}_{\boldsymbol{v_1}} + \boldsymbol{J}_{\boldsymbol{\omega_1}}^T\boldsymbol{I_1}\boldsymbol{J}_{\boldsymbol{\omega_1}} & \cdots \\ m_n\boldsymbol{J}_{\boldsymbol{v_n}}^T\boldsymbol{J}_{\boldsymbol{v_n}} + \boldsymbol{J}_{\boldsymbol{\omega_n}}^T\boldsymbol{I_n}\boldsymbol{J}_{\boldsymbol{\omega_n}} & \end{bmatrix},$$

the platform inertia matrix is given by

$$\boldsymbol{H_p} = \begin{bmatrix} m_{tot}(\boldsymbol{r}_{\boldsymbol{CoM}}^{\mathcal{I}}\times) & \boldsymbol{I}_{\boldsymbol{tot}}^{\mathcal{I}} - m_{tot}(\boldsymbol{r}_{\boldsymbol{CoM}}^{\mathcal{I}}\times)(\boldsymbol{r}_{\boldsymbol{CoM}}^{\mathcal{I}}\times) \end{bmatrix}.$$

Finally the coupling inertia matrix is obtained by

$$\boldsymbol{H_{pm}} = \begin{bmatrix} \begin{bmatrix} m_1\boldsymbol{J_{v_1}} & \cdots \\ m_n\boldsymbol{J_{v_n}} & \end{bmatrix} \\ \begin{bmatrix} bmI_1^{\mathcal{I}}\boldsymbol{J_{\omega_1}} + m_1\left(\boldsymbol{r}_{\boldsymbol{CoM_1}}^{\mathcal{I}} - \boldsymbol{r}_{\boldsymbol{CoM}}^{\mathcal{I}}\right)\times\boldsymbol{J_{v_1}} & \cdots \\ \boldsymbol{I}_{\boldsymbol{n}}^{\mathcal{I}}\boldsymbol{J_{\omega_n}} + m_n\left(\boldsymbol{r}_{\boldsymbol{CoM_n}}^{\mathcal{I}} - \boldsymbol{r}_{\boldsymbol{CoM}}^{\mathcal{I}}\right)\times\boldsymbol{J_{v_n}} & \end{bmatrix} \end{bmatrix}. \tag{9}$$

4. Control Design

The plant of the control system is presented in Figure 3.

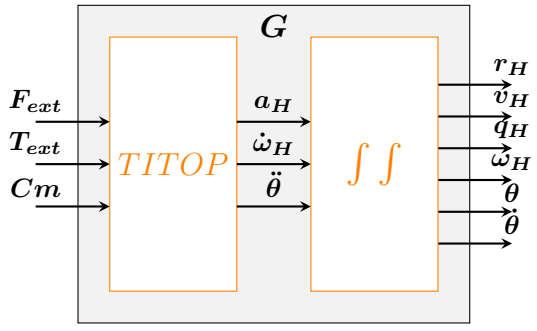

Figure 3: Plant

The block named *TITOP* is built with the model presented in section 2. The second block is built based on the set of equations given by

$$\begin{cases} \boldsymbol{x_H}(t_i) = \int_{t_{i-1}}^{t_i}\boldsymbol{v_H}dt + \boldsymbol{x_H}(t_{i-1}) \\ \boldsymbol{v_H}(t_i) = \int_{t_{i-1}}^{t_i}\boldsymbol{a_H}dt + \boldsymbol{v_H}(t_{i-1}) \\ \boldsymbol{q_H}(t_i) = \int_{t_{i-1}}^{t_i}\dot{\boldsymbol{q}}_{\boldsymbol{H}} + \boldsymbol{q_H}(t_{i-1}) \\ \dot{\boldsymbol{q}}_{\boldsymbol{H}}(t_i) = \frac{1}{2}\boldsymbol{W}(t_{i-1})\boldsymbol{q_H}(t_{i-1}) \\ \boldsymbol{\omega_H}(t_i) = \int_{t_{i-1}}^{t_i}\dot{\boldsymbol{\omega}}_{\boldsymbol{H}}dt + \boldsymbol{\omega_H}(t_{i-1}) \\ \boldsymbol{\theta}(t_i) = \int_{t_{i-1}}^{t_i}\dot{\boldsymbol{\theta}}dt + \boldsymbol{\theta}(t_{i-1}) \\ \dot{\boldsymbol{\theta}}(t_i) = \int_{t_{i-1}}^{t_i}\ddot{\boldsymbol{\theta}}dt + \dot{\boldsymbol{\theta}}(t_{i-1}) \end{cases}, \tag{10}$$

where $\boldsymbol{W} \in \mathbb{R}^{3\times 3}$ is obtained by

$$\boldsymbol{W} = \begin{bmatrix} 0 & -\boldsymbol{\omega_H} \\ \boldsymbol{\omega_H} & (\boldsymbol{\omega_H}\times) \end{bmatrix},$$

and $(\boldsymbol{\omega_H}\times) \in \mathcal{R}^{3\times 3}$ is the skew-symmetric matrix of $\boldsymbol{\omega_H}$.

Note that in the simulation, discrete time is considered. Thus, t_i corresponds to the time at step i and t_{i-1} to the previous step.

The state vector of the system $\mathbf{x} \in \mathbb{R}^{31+n}$ is given by

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{\eta} & \dot{\boldsymbol{\eta}} & \boldsymbol{x_H} & \dot{\boldsymbol{x}}_{\boldsymbol{H}} & \boldsymbol{q_H} & \boldsymbol{\omega_H} & \boldsymbol{\theta} & \dot{\boldsymbol{\theta}} \end{bmatrix}^T, \tag{11}$$

where $\boldsymbol{\eta} \in \mathbb{R}^3$ corresponds to the displacement due to sloshing in the fuel and oxidizer tanks. Remember that each tank has 3 sloshing masses with different natural frequencies and damping, $\dot{\boldsymbol{\eta}} \in \mathbb{R}^3$ is the time derivative of the displacement; $\boldsymbol{x_H} \in \mathbb{R}^3$ is the inertial position of the hub CoM, and the $\dot{\boldsymbol{x}}_{\boldsymbol{H}} \in \mathbb{R}^3$ is the inertial linear velocity at that point, $\boldsymbol{q_H} \in \mathbb{R}^4$ are the quaternions which represent the rotation from the body frame $\mathcal{F}_{\mathcal{H}}$ to the inertial frame $\mathcal{F}_{\mathcal{I}}$, and $\boldsymbol{\omega_H} \in \mathbb{R}^3$ is the hub angular velocity. It should be noted that all the elements of the platform (hub, fuel and oxidizer tanks, and the base of the robotic arm) have the same angular acceleration. Finally $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}} \in \mathbb{R}^n$ are the joint angles and joint rates, respectively.

The input vector $\mathbf{u} \in \mathbb{R}^{6+n}$ is given by

$$\mathbf{u} = \begin{bmatrix} \boldsymbol{F_{ext}} & \boldsymbol{T_{ext}} & \boldsymbol{Cm} \end{bmatrix}^T, \tag{12}$$

where $\boldsymbol{F_{ext}}, \boldsymbol{T_{ext}} \in \mathbb{R}^3$ correspond to the external forces and torques actuating on the hub. These inputs are written in the inertial frame, but measured in the body frame. The command torques at the joints are represented by $\boldsymbol{Cm} \in \mathbb{R}^n$.

The output $\mathbf{y} \in \mathbb{R}^{31+n}$ is defined by

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{x_H} & \dot{\boldsymbol{x}}_{\boldsymbol{H}} & \boldsymbol{q_H} & \boldsymbol{\omega_H} & \boldsymbol{\theta} & \dot{\boldsymbol{\theta}} \end{bmatrix}^T. \quad (13)$$

It should be remarked that the non linearities in the plant of the system are linearized by the inertia matrix $\boldsymbol{H}$, which decoupled de dynamics of the system and by feedback control by keeping the angular rate around an operating point, which corresponds to the tumbling rate of the target satellite.

The control design approach for the Chaser is based on a PID controller, whose gains are set by applying a solving the structured H_∞ control problem. More about this in 4.1. This controller can be subdivided into two controllers: a PID controller for the platform and a PD controller for the robotic arm - represented in Figure 4 by K_{Plat} and K_M, respectively.

The PID controller for the platform is represented in Figure 5, where the linear acceleration is resolved. A constant term, representing the centrifugal acceleration was also introduced as a disturbance to the system, since this term is significant for the system dynamics.

The resolved angular acceleration is calculated according to Figure 6.

The PD controller for the manipulator is represented in Figure 7, where the angular acceleration is resolved.Integral action was not introduced because references $\boldsymbol{\theta_d}$ and $\dot{\boldsymbol{\theta}}_{\boldsymbol{d}}$ keep changing until the position and velocity error at the end effector is around zero.

4.1. Structured Control Problem

Structured H_∞ is a new approach to control design which allows the engineer to directly and efficiently tune control architectures with a given structure. As the name suggests it is based on H_∞ control theory. An H_∞ is an optimal control algorithm, which minimizes the H_∞ norm of the lower linear fractional transformation

$$\boldsymbol{z} = F_l(P, K)w,$$

where P is the extended Plant and K is the controller; w correspond to the normalized exogenous inputs of P, *i.e.* reference commands, measurement noise, and disturbances. The normalized exogenous outputs are represented by z and include performance error and the force and torque commands.

For more complex systems, one might need to give more information about exogenous signals, in addition to the signals to be minimized. This approach is called signal-based H_∞ control and it is shown in Figure 8 [13].

Solutions to the H_∞ problem tend to be of higher order than real industrial control problems can handle. With Structured H_∞ one can write the requirements as weighted frequency dependent functions as a normal H_∞ problem, and solve the minimization problem providing a pre defined shape of controller K [3, 8, 10], which in this case is two PID and one PD controllers and two weighting gains for the EE's linear and angular velocity: $X_{v_{EE}}$ and $X_{\omega_{EE}}$, respectively.

Any linear control system can be written in the standardized representation of Figure 9, where we a have a Linear Time Invariant (LTI) plant P, exogenous inputs and outputs - w and z respectively - and some controller C. All reference signals and disturbances are gathered in w and performance related outputs in z [3, 8, 10].

In Structured H_∞ control, the augmented plant P includes, not only plant G but also all the remaining fixed blocks, e.g. blocks EE and IKP in Figure 4 (excluding gain $X_{v_{EE}}$ and $X_{\omega_{EE}}$, included in block IKM). C corresponds to a diagonal matrix

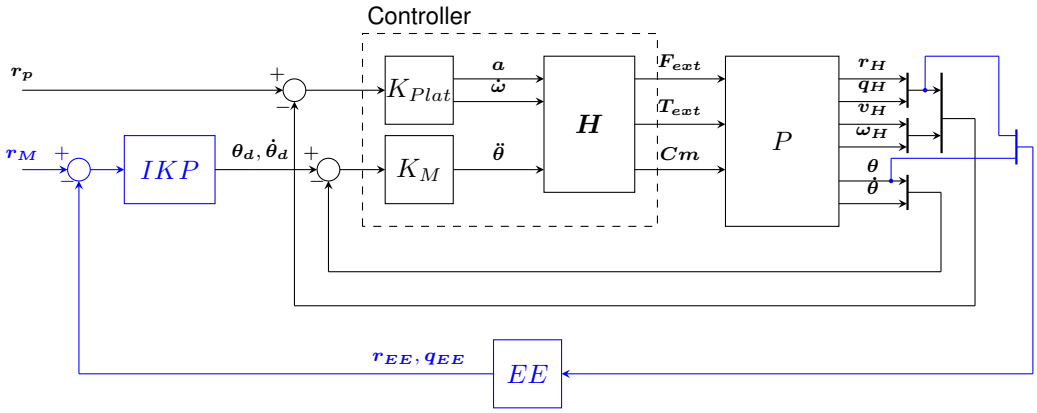

Figure 4: Simplified representation of the feedback controller. Both the inner and outer loops of the Manipulator's controller are represented.

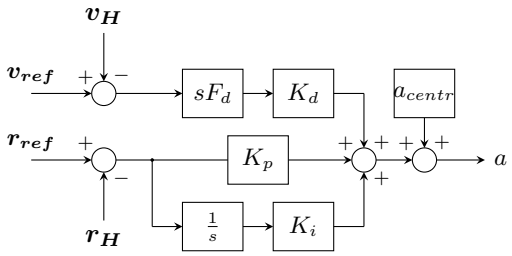

Figure 5: PID controller using the hub position error.

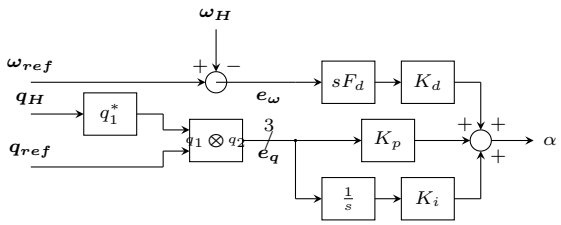

Figure 6: PID controller using hub the attitude.

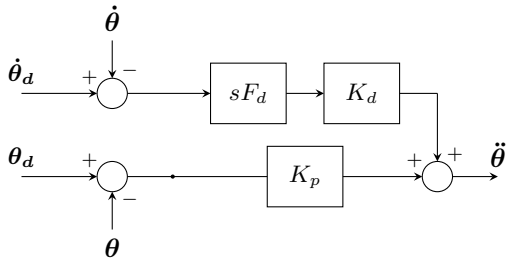

Figure 7: PD controller using the joint angles error.

with all tunable control elements, which may be repeated in matrix C. Using MATLAB, we can easily obtain the Standard Form of some system by setting tunable elements with function *realp* and by using function *slTuner* to obtain the Standard Form of a Simulink model [8].

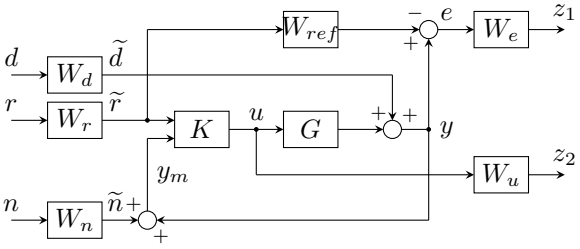

Figure 8: Representation of a signal-based H_∞ control problem.

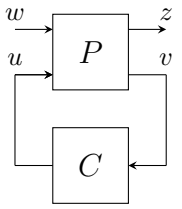

Figure 9: Standard form for Fixed Structured H_∞ control.

Requirements are specified through the weighting matrices in Figure 8.

5. Results and Conclusions

In this section, a small analysis of the controller is presented.

At first, the extended Nyquist theorem was used to study nominal stability. However, no proper results were obtained due to numerical errors. Therefore, time domain analysis followed in order to test if bounded inputs resulted in bounded outputs. This simulation was subjected to random noise and disturbances.

The initial hub position and attitude is assumed

to have a small offset w.r.t. to the target pose. In addition, there is a small initial angular rate error. In the time domain analysis the EE follows the trajectory in Figure 10, during the capture of the grappling point (GP) at the target. The position of the hub is able to stabilize and remain bounded during the capture - see Figure 11 . From Figure 12, we can see that the tumbling rate of the platform matches the tumbling rate of the target: $\boldsymbol{\omega} = \begin{bmatrix} 0 & 0.04 & 0 \end{bmatrix}^T$ rad/s.

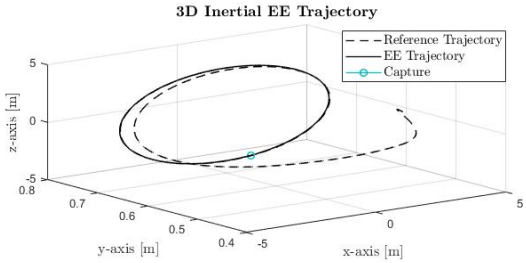

Figure 10: Representation of both the EE and the GP trajectories. The reference is plot with dashed line, and the EE trajectory with full line. The point, at which capture is achieved, is signalled with a blue circle.

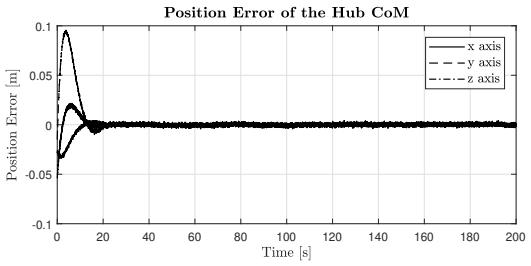

Figure 11: Position error of the Hub CoM.

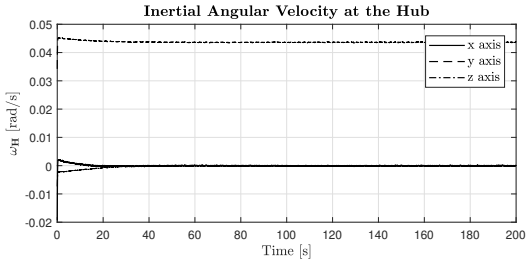

Figure 12: Inertial angular velocity of the hub.

Figures 13 and 14 shows the pose error of the EE. The capture was successful at $t = 1min40s$. Since the outputs remained bounded, nominal stability is assumed.

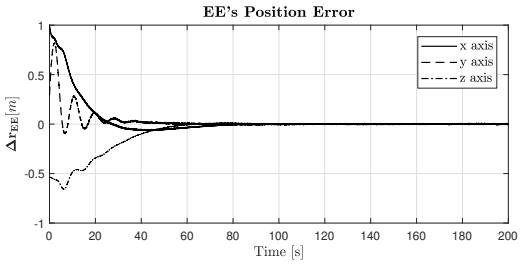

Figure 13: Position error of the EE w.r.t. the GP.

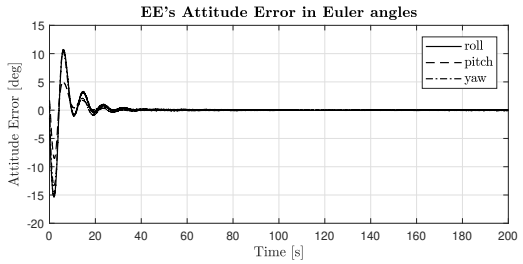

Figure 14: Attitude error of the EE, in Euler angles.

Robust stability margin was computed for three different solutions, using *robstab* method from MATLAB. The results are presented in Figure 15. All three of them outputted very similar results, and all with proper robust stability margin around the frequency range $\begin{bmatrix} 10^{-2} & 10^{-1} \end{bmatrix}$ rad/s.

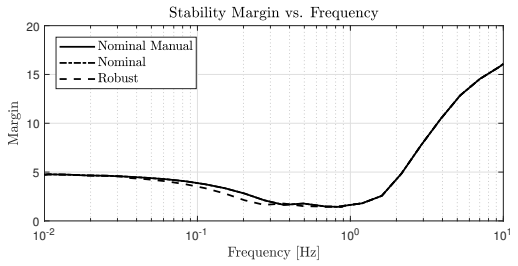

Figure 15: Stability margin

In order to study robust performance, at first, multiple simulations ran for different configurations of the uncertain parameters and the output variance was quite small for all outputs - in the range of 10^{-4} or smaller. Then, multiple simulations ran for different values of the tumbling rate of the target. The greater effect was for the hub CoM position, with a maximum variance of 0.02m. It was also noticeable that the higher the initial angular rate error between the target and the chaser, the higher the variance. This tests reflects the importance of the

angular rate in the dynamics of the system and, as a consequence, the importance of the effect of the centrifugal acceleration. For the other outputs the variance was not so significant.

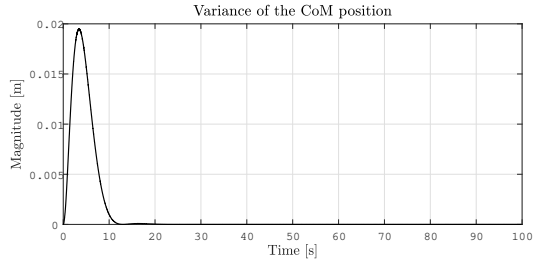

Figure 16: Variance of the position of the hub CoM, for different values of the tumbling rate.

6. Future Work

An important part of tracking the target with the robotic arm is the guidance introduced in the system. For that the IKP was used together with the damped least squares method to avoid numerical errors. In addition to this method, one can make use of the redundancy of the robotic arm (7 joints in a 6DoF problem) in order to avoid colliding with the chaser satellite and the robotic arm. An interesting study would be to provide different trajectories to the robotic arm and analyse how it behaves. Next, the results from the simulation could be compared to the results of the real robotic arm following the same trajectory, and using the controller designed. A different approach to control design could be done: solving the H_∞ problem followed by a order reduction in order to be compatible with the technology used in space.

Acknowledgements

I would like to thank professor João Miranda Lemos and Hans Strauch for the support during this thesis. A thank you to my colleagues Francisca, Marina, and Marta for accompanying me during these years at IST. And a special thank you to Lourenço for your endless patience.

References

[1] D. Alazard, C. Cumer, and K. Tantawl. *Linear Dynamic Modeling of Spacecraft with various flexible appendages and on-board angular momentums*. *Proceeding of the 7th International ESA Conference on Guidance, Navigation and Control Systems, Tralee (Ireland)*, June 2008.

[2] D. Alazard, J. A. Perez, T. Loquen, and C. Cumer. *Two-input two-output port model for mechanical systems*. *Proceeding of the 53rd AIAA Aerospace Sciences Meeting*, January 2015.

[3] P. Apkarian, editor. *Tuning Controllers Against Multiple Design Requirements*, June 2013. Proceedings of 2013 American Control Conference (ACC).

[4] A. Balestrino, G. Maria, and L. Sciavicco, editors. *Robust Control of Robotic Manipulators*. Dipartimento di Informatica e Sistemistica, University of Naples, 1984. Proceedings of IFAC: 9th Triennial World Congress.

[5] J. Bueno, W. Serrantola, R. Bezerra, and V. Grassi, editors. *LQR and H-infinity Controls of a Free-Floating Space Manipulator with Two Arms*, 2016. Proceedings of XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium.

[6] S. Buss. *Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods*. University of California, October 2003.

[7] A. Colomè and C. Torras. *Redundant Inverse Kinematics: Experimental Comparative Review and Two Enhancements*. Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona.

[8] V. Dubanchet, D. Saussié, D. Alazard, C. Bérard, and C. L. Peuvédic, editors. *Motion Planning and Control of a Space Robot to Capture a Tumbling Debris*, April 2015. Proceedings of the 3rd CEAS Specialist Conference on Guidance, Navigation and Control.

[9] Fahimi and Farbod. *Autonomous Robots: Modeling, Path Planning, and Control*. Springer-Verlag London, 2009. Chapter 2, ISBN: 978-0-387-09537-0.

[10] P. Gahinet and P. Apkarian, editors. *Structured H_∞ Synthesis in MATLAB*, 2011. Proceedings of IFAC, Milan.

[11] K. Nanos and E. Papadopoulos. *On the Dynamics and Control of Free-floating Space Manipulator Systems in the Presence of Angular Momentum*. *frontiers*, June 2017. DOI: 10.3389/frobt.2017.00026.

[12] B. Siciliano and O. Khatib. *Handbook of Robotics*. Springer-Verlag London, 2nd Edition, 2016. ISBN: 978-3-319-32550-7.

[13] S. Skogestad and I. Postlethewaite. *Multivariable Feedback Control: Analysis & Design* . John Wiley & Sons, 2nd edition edition, 2005. ISBN-13 978-0-470-01167-6.

[14] M. Wilde, S. K. Choon, A. Grompone, and M. Romano. *Equations of Motion of Free-Floating Spacecraft-Manipulator Systems: An Engineer's Tutorial*. *frontiers in Robotics and AI*, April 2018. doi: 10.3389/frobt.2018.00041.