To cite this document: Dubanchet, Vincent and Saussié, David and Alazard, Daniel and Bérard, Caroline and Le Peuvedic, Catherine *Modeling & control of a space robot for active debris removal.* (2014) In: ESA GNC 2014 - 9th International ESA Conference on Guidance, navigation & Control Systems, 2 June 2014 - 6 June 2014 (Porto, Portugal).

MODELING & CONTROL OF A SPACE ROBOT FOR ACTIVE DEBRIS REMOVAL

V. Dubanchet[1,2], D. Saussié[1], D. Alazard[2], C. Bérard[2], C. Le Peuvédic[3]
[1] École Polytechnique de Montréal, Montréal, Canada
[2] ISAE-Supaéro, Toulouse, France
[3] Thalès Alenia Space, Cannes, France
{vincent.dubanchet, david.saussie}@polymtl.ca, {daniel.alazard, caroline.berard}@isae.fr,
catherine.le-peuvedic@thalesaleniaspace.com

ABSTRACT

Space access and satellites lifespan are increasingly threatened by the great amount of debris in Low Earth Orbits (LEO). Among the many solutions proposed in the literature so far, the emphasis is put here on a robotic arm mounted on a « chaser » satellite to capture massive debris, such as dead satellites or launch vehicle upper stages. The modeling and control of such systems are investigated throughout the paper. Dynamic models rely on an adapted Newton-Euler algorithm, and control algorithms are based on the recent structured H_∞ method. The main goal is to efficiently track a target point on the debris while using simple PD-like controllers to reduce computational burden. The structured H_∞ framework proves to be a suitable tool to design a reduced-order robust controller that catches up with external disturbances and is simultaneously compatible with current space processors capacities.

1. Introduction

After 60 years of intensive space use and uncontrolled debris proliferation, Earth orbits have now reached a shifting point where human intervention is necessary. Known as the « Kessler syndrome », a 200 years forecast stated that space access would almost disappear if nothing is done [1, 2]. As shown by the NASA scientist J.C. Liou through [3], at least 5 massive objects, like dead satellites or rocket upper stages, need to be de-orbit each year to reverse the current trend and stabilize the debris population. It is clearly shown in Fig. 1 how recent events and human carelessness increased dramatically the number of objects in low Earth orbits, especially by the Fengyun Chinese satellite break-up and the Iridium/Cosmos collision.

In order to mitigate debris proliferation, space agencies around the world have investigated all kinds of solutions. Among them, one can find an Earth-based pulsed laser [5], electrodynamic tethers [6], foam sprays [7], or even debris captures based on harpoons and nets solutions [8]. Nevertheless, considering the urgent situation, the robotic arm approach has appeared as the best capture mechanism to get a prototype ready to fly in the next years. Its advantage over the previous ideas lies in its *Technology*

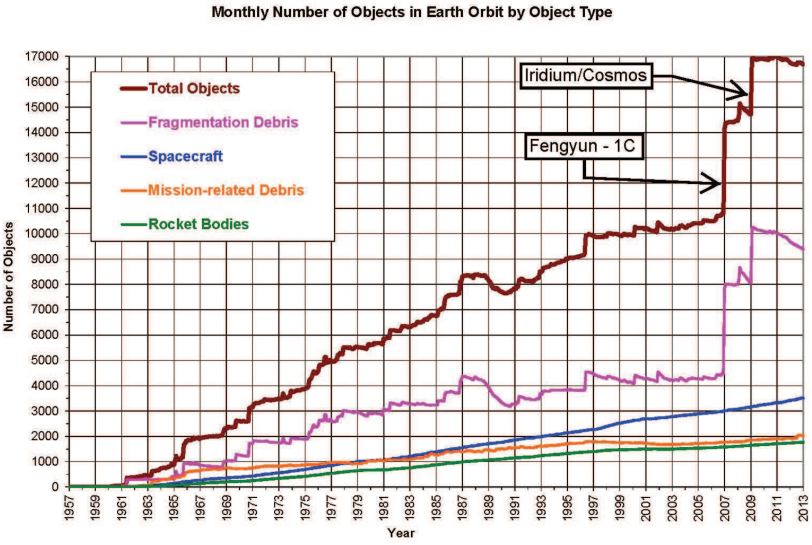


FIGURE 1 – Debris population evolution during the last 60 years [4]

Readiness Level (TRL), since robotics has been embedded in space systems for 30 years, and has known a tremendous development since then, in both research and industry.

Robotics is about to replace astronauts for safety reasons and difficulty of the performed tasks. Extra-Vehicular Activities (EVA) are indeed becoming more and more risky as debris multiply, and robotic arms have been introduced in space systems to reduce their frequency. The first attempt was made with the Canadian arm, *Canadarm*, attached to the Space Shuttle in the 80's [9]. Many others have been added on the *International Space Station* (ISS) afterwards, like the *DEXTRE* arm, the *European Robotic Arm* or the *JEMRMS*. Apart from operating in micro-gravity, these examples are quite similar to classic terrestrial arms whose base is fixed on the ground, since the ISS is so massive it almost experiences no reaction from the arms motions. The dynamic coupling between the arm and its base started being a real issue for missions like the Japanese *Engineering Test Satellite-VII* (ETS-VII) [10], or the American *Orbital Express* project [11], for which the arm was embedded on a satellite and created a strong dynamic coupling. In the last years, the German space agency [1] has been working on the *DEutsche Orbitale Servicing* (DEOS) mission, and takes advantage of its robotic facilities to test and validate robotic sensors and hardware configurations [12].

These past and actual missions are based on a very active and prolific research field. This paper tries to summarize the main issues through the literature review in Section 2., by focusing on the major contributions in modeling and control of space robots. Section 3. presents the algorithms used to model and simulate their dynamics, before applying the *structured* H_∞ method for control purpose in Section 4.. A 1-Dimensional example is used to highlight the control strategy and is then applied to a full space robot. Eventually, Section 5. gathers the paper main results and proposes future research avenues.

2. Literature Review

Modeling and control of space robotic manipulators is mixing spacecrafts and robotics theories. As a simplifying hypothesis, it will be considered that the satellite equipped with the robotic arm is floating freely in space such that orbital mechanics can be neglected at this stage [2]. In the sequel, we will refer to such a system as a « space robot ».

A. Kinematics and Singularities

Kinematics of a robot is based on the relation linking the end-effector position and orientation to the joints angles : $\mathbf{x_e} = f(\mathbf{q})$, where $\mathbf{x_e} \in \mathcal{R}^{6\times 1}$ contains the position and Euler angles of the end-effector frame, while $\mathbf{q} \in \mathcal{R}^{N\times 1}$ is the arm's configuration (i.e. joints angles). When derived with respect to time, this relation results in a linear dependency of the velocities with the Jacobian matrix $\mathbf{J_e} \in \mathcal{R}^{6\times N}$:

$$\dot{\mathbf{x}}_\mathbf{e} = \mathbf{J_e}(\mathbf{q})\,\dot{\mathbf{q}} \tag{1}$$

Singularities are then defined as the configuration where the Jacobian matrix is loosing its full rank [14]. For a fixed-based arm, these configurations correspond to the alignment of two axis of rotation or when the arm's workspace is reached. Singularities are critical for motion planning and control phases. For fixed-based arms, they are only functions of the configuration $\mathbf{q}$. In this case, they are called *kinematic singularities* [15].

Conversely, in the case of a space robot, it has been shown that additional *dynamic singularities* appear [16]. This analysis lies on the fact that the new Jacobian matrix is now function of the base attitude. To describe the inertial state of the effector, the direct kinematics is given by : $\mathbf{x_e} = f(\mathbf{x_b}, \mathbf{q})$, where $\mathbf{x_e}$ is the base position and attitude. It yields the following velocity relation :

$$\dot{\mathbf{x}}_\mathbf{e} = \mathbf{J_b}\,\dot{\mathbf{x}}_\mathbf{b} + \mathbf{J_e}(\mathbf{q})\,\dot{\mathbf{q}} \tag{2}$$

where J_b, the base's Jacobian matrix, is constant and only function of the arm's location on the spacecraft.
Now using the fact that the global momentum is conserved for a *free-floating* space robot (i.e. the base

[1] *Deutsches zentrum für Luft- und Raumfahrt* (DLR)

[2] More insights about the full orbital mechanics of the space robot can be found in [13].

is moving freely in reaction to the arm's motion [17]), an additional relation is found between $\dot{\mathbf{x}}_{\mathbf{b}}$ and $\dot{\mathbf{q}}$ that leads to the *Generalized Jacobian Matrix* (GJM) of Umetani and Yoshida [18] :

$$\mathbf{J}^{*} = \mathbf{J_e} - \mathbf{J_b}\, \mathbf{I_b}^{-1}\, \mathbf{I_m} \tag{3}$$

with $\mathbf{I_b}$ and $\mathbf{I_m}$ the mass matrices of, respectively, the base and the manipulator.

Dynamic singularities of such systems are *path-dependent*, in the sense that they can be avoided depending on the effector path followed to reach an inertial position [19]. Due to the nonholonomic nature of the angular momentum conservation, base attitude is determined by the effector path, and a careful planning enables to reject the paths where the arm and base configurations result in a singularity, as shown in [20].

B. Dynamics and Modeling

Considering a moving base is equivalent to add an initial segment to a classic robotic arm, except that this one has six degrees of freedom instead of one. Based on this assumption, Newton equations or Lagrangian approach can be adapted to derive a space robot model. On the one hand, Carignan uses the recursive Newton-Euler algorithm [21] to obtain the base dynamics, by computing the resulting efforts when the arm is moving [22]. On the other hand, by augmenting the Lagrangian of the robotic arm with the base's kinetic and potential energy, the space robot dynamics are derived through [23]. The first algorithm proved to be numerically more efficient, while the second one is much more suitable to include flexible behaviors.

The dynamic model of a space robot takes the following form, based on [24] :

$$\mathbf{M}(\mathbf{q}) \begin{bmatrix} \ddot{\mathbf{x}}_{\mathbf{b}} \\ \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{h_b}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\omega_b}) \\ \mathbf{h_e}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\omega_b}) \end{bmatrix} = \begin{bmatrix} \infty \\ \prime \end{bmatrix} \mathbf{F_b} + \begin{bmatrix} \prime \\ \infty \end{bmatrix} \boldsymbol{\tau} + \begin{bmatrix} \mathbf{T_e}^{T} \\ \mathbf{J_e}^{T} \end{bmatrix} \mathbf{F_e} \tag{4}$$

with

- $\mathbf{M}$	Global mass matrix
- $(\mathbf{h_b}, \mathbf{h_e})$	Nonlinear effects (centrifugal and Coriolis)
- $\boldsymbol{\omega_b}$	Base angular rate ($\dot{\mathbf{x}}_{\mathbf{b}} = \begin{bmatrix} \dot{\mathbf{x}}_{\mathbf{G_b}} & \boldsymbol{\omega_b} \end{bmatrix}^{T}$)
- $\mathbf{F_b}$	Efforts applied at the base
- $\boldsymbol{\tau}$	Joints torques
- $\mathbf{F_e}$	Efforts applied at the end-effector
- $\mathbf{T_e}$	Spatial twist from the effector to the base
- $(\infty, \prime)$	Identity and Zero matrices of suitable size

Eventually, works based on a mixed approach were developed by Rodriguez [25] and Saha [26] to provide the *Decoupled Natural Orthogonal Complement* (DeNOC) method. By expressing the velocity constraints on each segments, analytical expressions of $\mathbf{M}$ and $\mathbf{h}$ can be obtained and a very efficient and stable algorithm is created for inverse dynamics [27].

C. Control Issues

Two main strategies exist for the control of a space robot : one couples the *Attitude and Orbital Control System* (AOCS) of the spacecraft with the robotic arm controller, while the other lets the base move freely in reaction of the arm motion. They are called, respectively, of *free-flying* and *free-floating* systems [17, 28]. It has been suggested by Papadopoulos and Dubowsky that any classic terrestrial control scheme can be applied to these types of space robots for a limited workspace, provided that dynamic singularities are avoided [29].

The *free-flying* mode enables to maintain a base attitude during an arm motion. This is a strong requirements if the space robot was to be operated from the ground, and the antennas need to be accurately pointed. The same kind of constraints could apply to ensure a sun pointing performance or to move the spacecraft to avoid an impact while the arm is moving. One strategy is then to design two separate controllers, as proposed by Oda in [30], but the AOCS can be quickly saturated if the arm moves too fast, thus degrading the overall performance. An improvement has been proposed in [31] by estimating roughly the disturbance induced by the arm, and then feedforwarding it to the AOCS. Another option is to adapt the arm motion speed to meet exactly the spacecraft's actuators saturation [32]. Eventually, a complete coordinated control is designed by Papadopoulos and Dubowsky through [33]. Based on an adapted Transposed Jacobian control, they achieve a position and orientation control of both the end-effector and the base.

When fuel consumption or arm disturbance have to be canceled, the *free-floating* mode is preferred. The momentum conservation introduces a nonholonomic relation that enables to control the base with cyclic end-effector motions [20, 34]. By inspecting the angular momentum structure, Nenchev also introduced the *Reaction Null Space*, enabling to plan end-effector trajectories that do not disturb the base position and attitude [24]. He also proposes a control scheme derived from the computed-torque strategy, which cancels exactly any disturbance from the arm for a sufficient redundancy. This is ensured for a manipulator with more than 6 Degrees-of-Freedom (DoF), or it can be obtained artificially by reducing the end-effector constraints to the angular position for example.

Considering the trajectory planning, the goal of the space robot is mainly to capture a target. A cartesian motion is proposed in [19] to move the end-effector toward a target position, while avoiding any dynamic singularity to occur. Other studies propose to catch an object with a linear motion [35], by capturing it when its trajectory crosses the arm's workspace which is free of singularities. A more realistic approach is given in [36], where the object rotation is accounted and the end-effector is controlled to follow a circular motion. Recent studies also consider the parameter uncertainties, by designing an adaptive control law in [37], or by using a Kalman filter to predict the target motion [38].

3. Modeling

The model of a space robot is now given in more detail. Starting with the classic robotics results, it is shown how the base motion can be handled in the kinematic and dynamic modeling phases by adapting traditional algorithms.

A. Kinematics

For a N DoF arm, $N+1$ frames are rigidly attached to each segment, from R_0 to R_N. The first frame R_0 is attached to the ground, when the arm is fixed, or to its base, when it is moving. The last frame is attached to the end-effector. Each of them is chosen according to the classic Denavit-Hartenberg convention [14], as illustrated in Fig. 2.

Using the homogeneous transformation framework [39], where $\mathbf{{}^{i}x} = \left[\begin{array}{cc} {}^{i}x^T & 1 \end{array}\right]^T$ stands for the x vector expressed in the R_i frame, and $\mathbf{{}^{i-1}T_i}$ for the homogeneous transformation from R_{i-1} to R_i, the direct kinematics is expressed as follows :

$$\begin{aligned} \mathbf{{}^{0}x_E} &= \mathbf{{}^{0}T_1}(q_1)\,\mathbf{{}^{1}T_2}(q_2)\ldots\mathbf{{}^{N-1}T_N}(q_N)\,\mathbf{{}^{N}x_E} \\ &= \mathbf{{}^{0}T_N(q)}\,\mathbf{{}^{N}x_E} \end{aligned}$$

with $\mathbf{{}^{0}x_E}$ the effector inertial position, and $\mathbf{{}^{N}x_E}$ its position in R_N : $\mathbf{{}^{N}x_E} = \left[\begin{array}{cc} \prime & 1 \end{array}\right]$.

The inverse kinematics is given by inverting the previous relation. An effector location and orientation is specified through the matrix $\mathbf{{}^{0}T_{N,des}}$. Then the last joint angle q_N is computed by examining the matrix equality $\mathbf{{}^{0}T_{N-1}} = \mathbf{{}^{0}T_{N,des}}\,\mathbf{{}^{N-1}T_N(q_N)}$. Once the right-hand term is known, the same process is repeated recursively to deduce the other joint angles. More insights are given about this analytical technique and other approximate methods are proposed in [14, 40].

These algorithms are easily adapted for a space robot, noting that the base can be considered as a special segment with 6 DoF. The frame R_b is then attached to its Center of Gravity (CoG), and an inertial frame R_{in} needs to be defined. It is worth noticing that R_0 and R_b are fixed one over the other, as seen in Fig. 2, but R_0 is fixed to the arm's base according to the Denavit Hartenberg convention. Thus the transformation matrix from the inertial origin O_{in} to the frame R_0 is given by :

$$\mathbf{{}^{in}T_0} = \mathbf{{}^{in}T_b}\,\mathbf{{}^{b}T_0} = \left[\begin{array}{cc} {}^{in}R_b(\mathbf{\Phi_b}) & \mathbf{X_{G_b}} \\ \prime_{1\times 3} & 1 \end{array}\right] \left[\begin{array}{cc} {}^{b}R_0 & \mathbf{p_{b,0}} \\ \prime_{1\times 3} & 1 \end{array}\right] \tag{5}$$

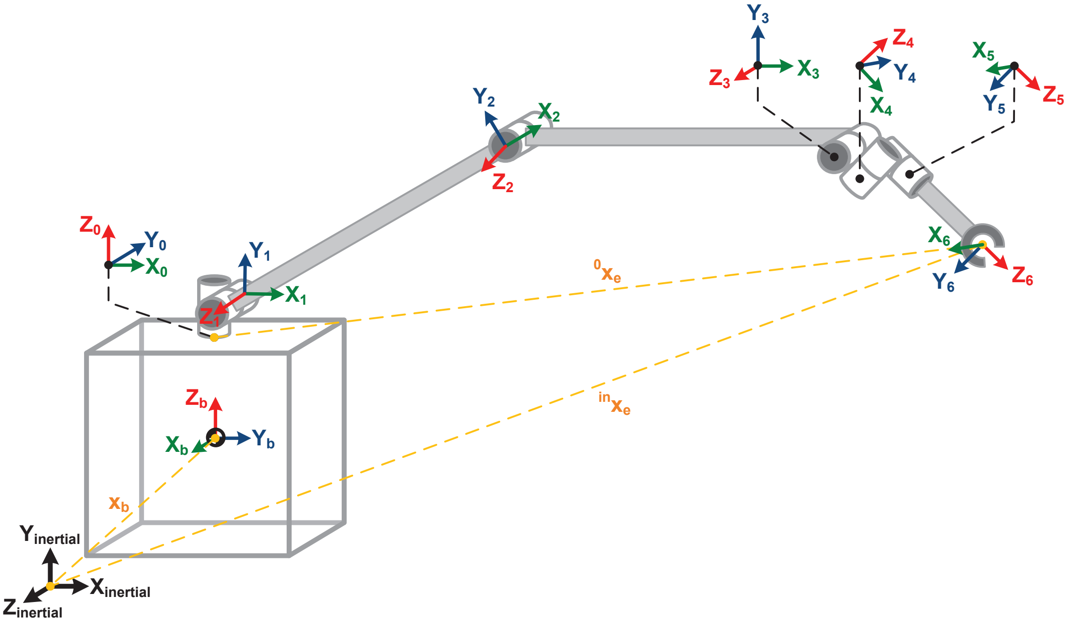


FIGURE 2 – Space robot frames using the classic Denavit-Hartenberg convention

with $\mathbf{\Phi_b}$ and $\mathbf{X_{G_b}}$, respectively, the Euler angles of the base and its CoG inertial location G_b, ${}^{in}R_b(\mathbf{\Phi_b})$ the resulting rotation matrix from the inertial frame R_{in} to R_b, ${}^{0}R_b$ the rotation matrix from the base frame R_b to the arm's initial frame R_0, and $p_{b,0}$ the vector between them : $p_{b,0} = G_b O_0$, which is constant in R_b or R_0.

The direct and inverse kinematics for a space robot are now based on the relation :

$$ {}^{\mathbf{in}}\mathbf{x_E} = {}^{\mathbf{in}}\mathbf{T_0}(\mathbf{x_b})\,{}^{\mathbf{0}}\mathbf{T_N}(\mathbf{q})\,{}^{\mathbf{N}}\mathbf{x_E} \tag{6} $$

B. Kinetics

Given the R_i frames defined previously, the linear and angular velocities of their origin are obtained recursively through the Newton-Euler relations, specified here for rotating joints :

$$
\begin{aligned}
\boldsymbol{\omega_i} &= \boldsymbol{\omega_{i-1}} + q_i\,\mathbf{z_i} && (7)\\
\mathbf{v_i} &= \mathbf{v_{i-1}} + \boldsymbol{\omega_i} \times \mathbf{p_{i-1,i}} && (8)\\
\dot{\boldsymbol{\omega}}_\mathbf{i} &= \dot{\boldsymbol{\omega}}_\mathbf{i-1} + q_i\,(\boldsymbol{\omega_i} \times \mathbf{z_i}) && (9)\\
\dot{\mathbf{v}}_\mathbf{i} &= \dot{\mathbf{v}}_\mathbf{i-1} + \dot{\boldsymbol{\omega}}_\mathbf{i} \times \mathbf{p_{i-1,i}} + \boldsymbol{\omega_i} \times (\boldsymbol{\omega_i} \times \mathbf{p_{i-1,i}}) && (10)
\end{aligned}
$$

with

- $(\mathbf{v_i}, \boldsymbol{\omega_i})$ Linear/Angular velocities of the frame R_i attached at O_i
- $(\dot{\mathbf{v}}_\mathbf{i}, \dot{\boldsymbol{\omega}}_\mathbf{i})$ Linear/Angular accelerations
- q_i i^{th} Axis of rotation
- $\mathbf{z_i}$ i^{th} Axis of rotation
- $\mathbf{p_{i-1,i}}$ Vector $O_{i-1}O_i$ expressed in R_i

The base velocity and acceleration affect the initial conditions of the previous algorithm. Indeed, the values of $(\mathbf{v_0}, \boldsymbol{\omega_0})$ and $(\dot{\mathbf{v}}_\mathbf{0}, \dot{\boldsymbol{\omega}}_\mathbf{0})$ are no longer 0, but functions of the base parameters instead. Recording that $p_{b,0}$ is the vector between the frames R_b and R_0, one obtains :

$$
\begin{aligned}
\boldsymbol{\omega_0} &= \boldsymbol{\omega_b} && (11)\\
\mathbf{v_0} &= \mathbf{v_b} + \boldsymbol{\omega_b} \times \mathbf{p_{b,0}} && (12)\\
\dot{\boldsymbol{\omega}}_\mathbf{0} &= \dot{\boldsymbol{\omega}}_\mathbf{b} && (13)\\
\dot{\mathbf{v}}_\mathbf{b} &= \dot{\mathbf{v}}_\mathbf{b} + \dot{\boldsymbol{\omega}}_\mathbf{b} \times \mathbf{p_{b,0}} + \boldsymbol{\omega_b} \times (\boldsymbol{\omega_b} \times \mathbf{p_{b,0}}) && (14)\\
& && (15)
\end{aligned}
$$

The new initial conditions for the algorithm are $(\mathbf{v_b}, \boldsymbol{\omega_b})$ and $(\dot{\mathbf{v}}_\mathbf{b}, \dot{\boldsymbol{\omega}}_\mathbf{b})$,which are respectively the linear/angular velocities and accelerations of the spacecraft's CoG. These quantities are given by its rigid body dynamic equations, detailed in the sequel.

C. Dynamics

To complete the kinetic part of the Newton-Euler equations, the dynamic equations are computed through a descending recursion, from the efforts applied by the environment on the effector, to the efforts applied by the arm on its base. As presented in [39] or [14], the classic set of recursive equations starts with $f_{N+1} = f_{\mathit{eff}}$ and $n_{n+1} = n_{\mathit{eff}}$ and continues $\forall i = N \dots 1$:

$$\begin{aligned}
\dot{\mathbf{v}}_{\mathbf{G_i}} &= \dot{\mathbf{v}}_\mathbf{i} + \dot{\boldsymbol{\omega}}_\mathbf{i} \times \mathbf{r_i} + \boldsymbol{\omega_i} \times (\boldsymbol{\omega_i} \times \mathbf{r_i}) & (16)\\
\mathbf{F_i} &= m_i \, \dot{\mathbf{v}}_{\mathbf{G_i}} & (17)\\
\mathbf{f_i} &= \mathbf{f_{i+1}} + \mathbf{F_i} & (18)\\
\mathbf{N_i} &= \mathbf{I_i} \, \dot{\boldsymbol{\omega}}_\mathbf{i} + \boldsymbol{\omega_i} \times (\mathbf{I_i} \, \boldsymbol{\omega_i}) & (19)\\
\mathbf{n_i} &= \mathbf{n_{i+1}} + \mathbf{p_{i-1,i}} \times \mathbf{f_i} + \mathbf{N_i} + \mathbf{r_i} \times \mathbf{F_i} & (20)\\
\boldsymbol{\tau_i} &= \mathbf{n_i}^T \, \mathbf{z_i} & (21)
\end{aligned}$$

with

- m_i Mass of the i^{th} segment
- $\mathbf{v_{G_i}}$ Linear velocity of its CoG G_i
- $\mathbf{r_i}$ Position of G_i from the frame R_i (i.e. vector O_iG_i)
- $\mathbf{I_i}$ Inertia matrix of the i^{th} segment, expressed at G_i, in R_i
- $\mathbf{F_i}$ Net force acting on the i^{th} segment
- $\mathbf{f_i}$ Force applied by the $(i-1)^{th}$ segment on the i^{th} one, at O_{i-1}
- $\mathbf{N_i}$ Net moment acting on the i^{th} segment
- $\mathbf{n_i}$ Moment applied by the $(i-1)^{th}$ segment on the i^{th} one
- $\boldsymbol{\tau_i}$ i^{th} joint torque, applied along $\mathbf{z_i}$

This approach is used in [21] to get the closed-form equations describing the arm's dynamics. An update of these results has been done by Featherstone in [41].

Eventually, Carignan adds the spacecraft's dynamics in this fixed-based model [22]. By writing the rigid-body equations, he expresses the efforts applied by the arm on the base as an external disturbance given by $-f_1$ and $-n_1$ in the previous algorithm. It yields :

$$\begin{aligned}
m_b \, \dot{\mathbf{v}}_\mathbf{b} &= \mathbf{f_b} - \mathbf{f_1} & (22)\\
\mathbf{I_b} \, \dot{\boldsymbol{\omega}}_\mathbf{b} + \boldsymbol{\omega_b} \times (\mathbf{I_b} \, \boldsymbol{\omega_b}) &= \mathbf{n_b} - \mathbf{n_1} - \mathbf{p_{b,0}} \times \mathbf{f_1} & (23)
\end{aligned}$$

with

- $\mathbf{f_b}$ External forces acting on the spacecraft (including its thrusters. . .)
- $\mathbf{n_b}$ External moments acting on the spacecraft (including its thrusters, reaction wheels. . .)

Using the notations introduced in [24], the closed-form equation for a space robot is given by the equation presented in (4), by merging equation (21) with (22) and (23) :

$$\mathbf{M}(\mathbf{q}) \begin{bmatrix} \dot{\mathbf{v}}_\mathbf{b} \\ \dot{\boldsymbol{\omega}}_\mathbf{b} \\ \ddot{\mathbf{q}} \end{bmatrix} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\omega_b}) = \begin{bmatrix} \mathbf{F_b} \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \mathbf{T_e}^T \\ \mathbf{J_e}^T \end{bmatrix} \mathbf{F_e} \qquad (24)$$

where $\mathbf{M}$ is the *Mass Matrix*, $\mathbf{h}$ the vector of the nonlinear effects (centrifugal and Coriolis forces), $\mathbf{F_b}$ and $\mathbf{F_e}$ concatenate the forces and moments applied, respectively, at the base and at the end-effector, $\mathbf{J_e}$ is the arm's Jacobian matrix, and $\mathbf{T_e}$ is the twist propagation matrix, from the effector O_N to the spacecraft's CoG G_b : $\begin{bmatrix} f_{\mathit{eff},b} & n_{\mathit{eff},b} \end{bmatrix}^T = \mathbf{T_e}^T \begin{bmatrix} f_{\mathit{eff}} & n_{\mathit{eff}} \end{bmatrix}^T$. Its expression is given as follows, with $[v^\times]$ the cross-product matrix defined by $v \times x = [v^\times] \, x$:

$$\mathbf{T_e} = \begin{bmatrix} \infty_{3\times3} & -[(G_bO_N)^\times] \\ {\prime}_{3\times3} & \infty_{3\times3} \end{bmatrix} \qquad (25)$$

Numeric and symbolic implementations of this algorithm have been used for both simulation and control purposes. The numeric version is embedded in a `Matlab/Simulink` scheme to perform the

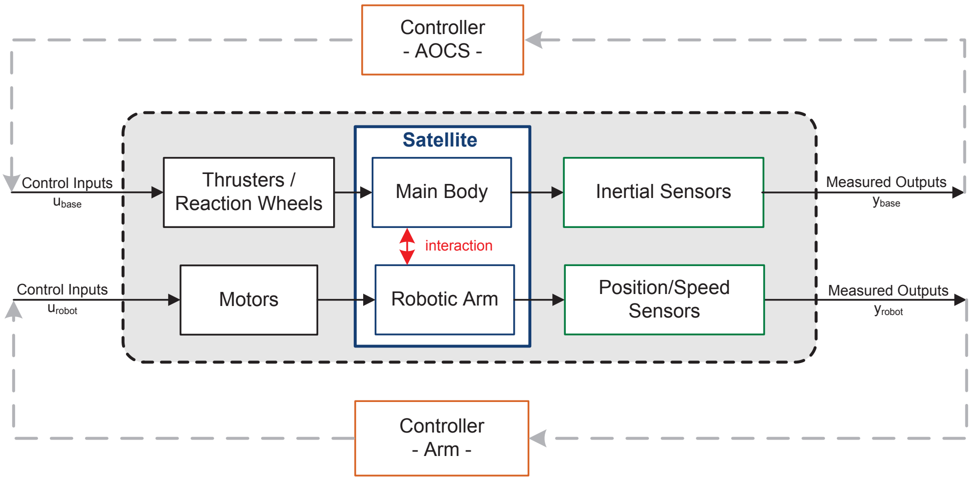


FIGURE 3 – Decentralized control architecture between AOCS and arm's controller

computation of $\mathbf{M}$ and $\mathbf{h}$ based on [22], while the symbolic version is implemented in `Maple` to compute symbolic models for robust control (e.g. the disturbance matrices in the case of parametric uncertainties).

4. Control Design

A control law is now investigated for the considered space robot to follow a given effector trajectory. A choice has to be done on the control architecture, between a *free-flying* and a *free-floating* configuration, and the controller gains are synthesized based on the well-known *structured* H_∞ framework. Two examples come to validate this approach, by means of a 1D model illustrating the main issues, and by a full arm with 6DoF, mounted on an actuated satellite.

A. Control Architecture

As mentioned in Section 2., two main architectures are used to control a space robot. Their choice over the other strongly depends on the processor capacities and on the need for a centralized or a decentralized control law between the arm and the spacecraft. It is worth recalling that actual space processors are significantly less powerful that terrestrial ones, due to their robustness to the hard environment in which they work. Even though, in theory, any terrestrial robot controller could be used in space according to [29], a low complexity control architecture will be preferred in this study to anticipate on a limited on-board processor.

The free-flying is chosen to control the space robot to extend the cartesian workspace of its manipulator and to improve its accuracy. Following the approach used in the Japanese mission ETS-VII [30], two separate controllers will be designed based on the spacecraft specifications, on the one side, and on the arm's specifications, on the other side. The AOCS design is kept very simple with a *Propotional-Derivative* controller, since the main purpose of the study is to underline the dynamic coupling between the arm and its support, and to show how to deal with it. The global control architecture is illustrated in Fig. 3. For the arm's controller, two main structures are used : one is centralized and gathered all measurements before specifying the joints command, and the other is decentralized on each joint by giving the command torque based on the local measurements. It will be shown how a decentralized controller, though much simpler to implement, will give similar results in our trajectory tracking issue.

B. Control Synthesis

H_∞ synthesis framework

The so-called H_∞ method has known a growing interest in the last years, with its implementation in a `Matlab` toolbox and the development of the *structured* H_∞ synthesis [42, 43]. Among the many advantages that offers the classic H_∞ framework, it can be emphasized that it enables to consider :

- Many control constraints at the same time, using weighting functions [44] ;
- Robustness and performance at the same time in the synthesis ;
- Multi-model synthesis to handle worst-cases scenarios.

The improvements with the structured method are numerous and could be summarized as follows :

- A low-order controller can be specified (skipping the reduction of the optimal full controller obtained by classic H_∞ solvers) ;
- A decentralized controller structure can be used with an axis-by-axis control ;
- Key parameters can be introduced in the design process to start a co-design study [45] ;
- Controller optimization is only focused on the transfers of interest by specifying the suitable constraints.

Furthermore, the synthesis will be based on the acceleration sensitivity function to perform an efficient disturbance rejection. Still, it will be shown that this method yields quite good results on the trajectory tracking scenario.

Linearized models

Since the H_∞ synthesis applies for linear models, the nonlinear model is first linearized around different arm's configurations. The linearization is done for a given configuration $\mathbf{q_0}$, and for zero joint velocities. This hypothesis holds well for low speed maneuvers, but should be revisited for an agile space robot concept.

Noting $\delta\mathbf{x}$ the variation around the nominal cases, it reads :

$$\mathbf{M}(\mathbf{q_0})\begin{bmatrix}\delta\dot{\mathbf{v}}_\mathbf{b}\\ \delta\dot{\boldsymbol{\omega}}_\mathbf{b}\\ \delta\ddot{\mathbf{q}}\end{bmatrix}=\begin{bmatrix}\mathbf{F_b}\\ \boldsymbol{\tau}\end{bmatrix} \tag{26}$$

since the nonlinear terms in h vanished because they are quadratic in the velocities [14], and the forces acting at the effector have been canceled in a pre-capture scenario.

H_∞ synthesis scheme

As mentioned earlier, the synthesis will be done on the acceleration sensitivity function to improve the disturbance rejection in priority [46]. By weighting the acceleration with the frequency template given in 27, it yields the global H_∞ scheme illustrated in Fig. 4.

$$W(s)=\frac{s^2+2\,\xi_{des}\,\omega_{des}\,s+\omega_{des}^2}{s^2} \tag{27}$$

Weighting function enables to reject efficiently any disturbance like the unmodeled dynamics of gravity gradient, atmospheric drag or solar pressure. They are used for both the base and the arm specifications, with, respectively, $W_b(s)$ defined by $(\boldsymbol{\omega_b},\boldsymbol{\xi_b})=(0.01\,rad/s,0.707)$, and $W_m(s)$ defined by $(\boldsymbol{\omega_m},\boldsymbol{\xi_m})=(20\,\boldsymbol{\omega_t},1)$. For the arm template, the angular rate of the target is fixed to $\boldsymbol{\omega_t}=1°/s$. The base's parameters are chosen to reproduce a quasi free-floating system, with a very slow AOCS in order to minimize the base consumption. On the other hand, the manipulator's parameters are taken 20 times greater than the debris tumbling rate $\boldsymbol{\omega_t}$, to ensure a **1% tracking error** in rotation.

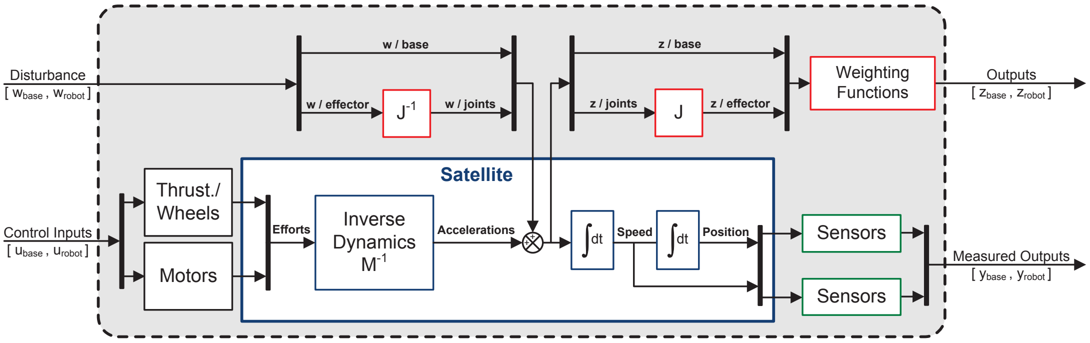


FIGURE 4 – H_∞ synthesis scheme

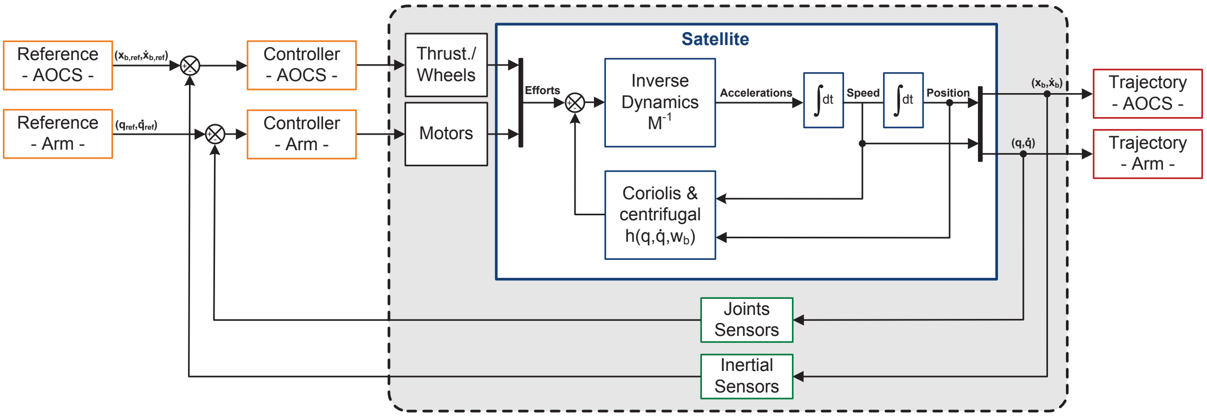


FIGURE 5 – Nonlinear simulation scheme

Validation scheme

The simulation is done on the full nonlinear model, as detailed in Fig. 5. The reference trajectory is followed at the target angular rate ω_t, and the tracking error is then observed to check the 1% requirements. The base consumption is also evaluated by integrating the base command (i.e. forces commanded to the thrusters) to compute the mass of fuel consumed, according to the following relation :

$$m_{fuel} = \frac{1}{g_0\, I_{sp}} \int_0^t u(t)dt \tag{28}$$

with $g_0 = 9.81\, m.s^{-2}$ the Earth gravity acceleration, I_{sp} the thrusters specific impulse, and $u = f_b + n_b/d$ the thrusters command according to the required AOCS force f_b and moment n_b. The distance d is the lever arm from the thrusters location to the base's CoG.

C. 1-Dimensional example

A 1-Dimensional example is firstly investigated to understand the main issues involved in the control design, and in the requirements specifications. The system structure is given in Fig. 6, and its dynamic equations in rotation are given by :

$$I_b\, \ddot{\phi}_b = n_b - \tau_m \tag{29}$$
$$I_m\, \ddot{\phi}_m = \tau_m \tag{30}$$

with I_m and I_b the inertia matrices of the manipulator and the base, ϕ_m and ϕ_b their respective attitude in the **inertial frame**, τ_m the manipulator torque, and n_b the base torque applied by on-board actuators.

It must be noted than the previous equations are written in the inertial frame, and that they neglect the Coriolis and centrifugal terms, responsible for the linear and angular coupling. In addition, the variables

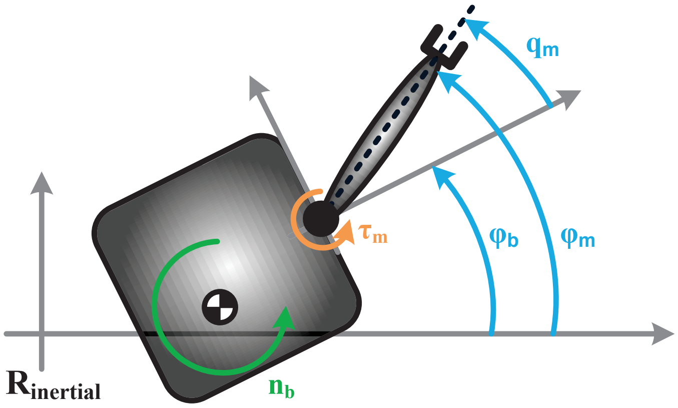


FIGURE 6 – 1-Dimensional system model

used in practice for the manipulator control are indeed relative ones. So the model can be re-written in terms of the relative coordinate $q_m = \phi_m - \phi_b$, as shown in Fig. 6, assuming that available measurements are the inertial base attitude ϕ_b (from inertial sensors) and the relative arm position q_m :

$$\begin{aligned}(I_m + I_b)\,\ddot{\phi}_b + I_m\,\ddot{q_m} &= n_b \quad (31)\\ I_m\,\ddot{\phi}_b + I_m\,\ddot{q_m} &= \tau_m \quad (32)\end{aligned}$$

Using the notations introduced in Fig. 3, the control inputs are reduced to $\left[\begin{array}{cc} n_b & \tau_m \end{array}\right]^T$, while the measured outputs are $\left[\begin{array}{cc} \phi_b & q_m \end{array}\right]^T$.

The H_∞ scheme in Fig. 4 is then applied to this system, using two *Proportionnal-Derivative* (PD) controllers on the base's actuators and on the arm's motor command. An additional constraint has been added on the controller H_∞-norm to avoid high gains, and another on the H_2-norm of the energy consumption of the base, by integrating its command signal. This last constraint must decrease the base actuator consumption, saving by this mean the fuel consumed. The synthesis yields the following settings for the PD controllers :

$$\begin{aligned}C_b(s) &= K_{p,b} + K_{d,b}\,s = 10 + 12\,s \quad (33)\\ C_m(s) &= K_{p,m} + K_{d,m}\,s = 10 + 15\,s \quad (34)\end{aligned}$$

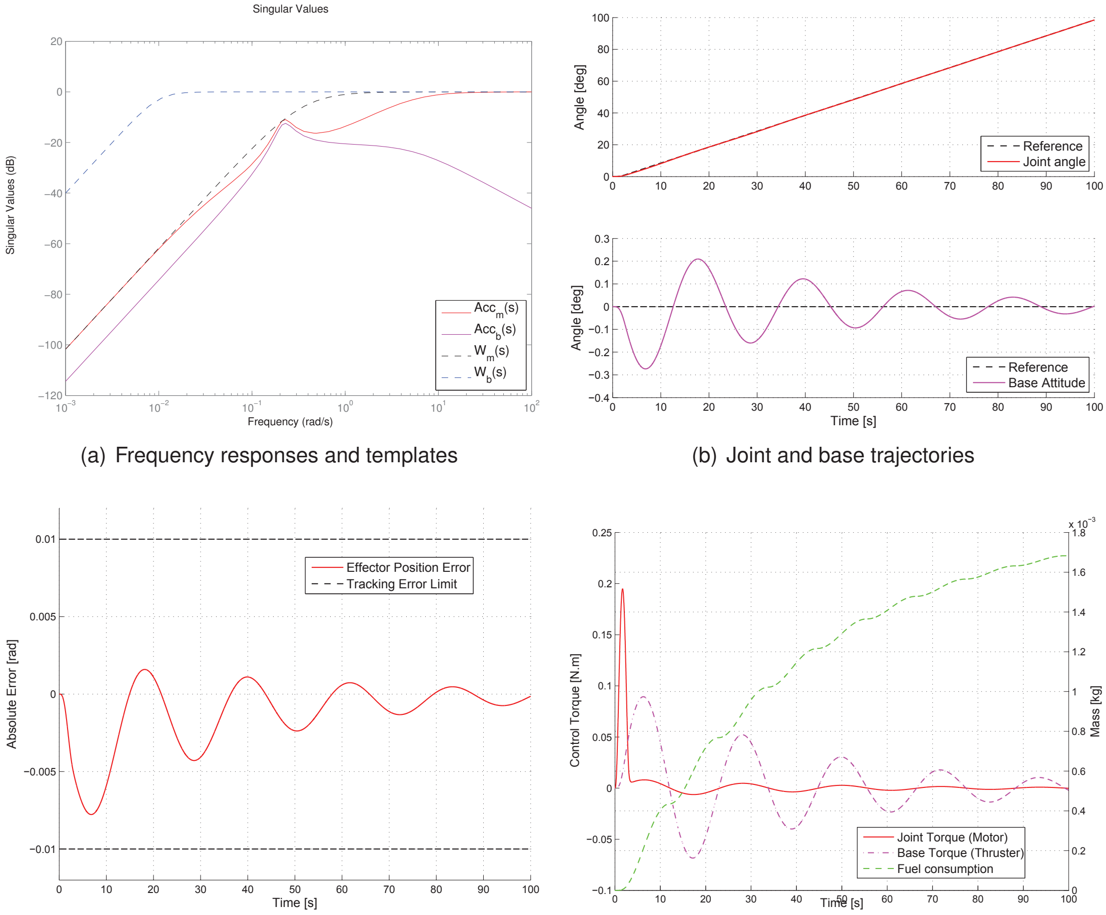
The frequency responses on the acceleration sensitivity function are given in Fig. 7(a), where the templates $W_b(s)$ and $W_m(s)$ are clearly respected. The trajectory tracking for a circle reference is also illustrated in Fig. 7(c), and the corresponding command signals and consumption in Fig. 7(d). On this last graph, the mass of fuel consumed for the base attitude control shows is very low thanks to the soft constraint specified in the H_∞ synthesis.

D. Full space robot example

The robotic arm used for the complete example has 6 DoF and its structure is illustrated in Fig. 2. It is inspired from a human arm, with a shoulder, a forearm, an arm and a wrist. The base is a cubic satellite of 200 kg and 2 m a side, while the arm weights almost 70 kg and each of its main segments are 2.6 meters long for a full extension of 5.7 meters.

Two strategies are used to control this space robot. The first one is based on a full decoupling controller by the mean of the mass matrix used in the feedback loop. The final results show that this controller is very efficient on the trajectory tracking issue. The second approach is based on a decentralized structure, with a PD-like controller on each axis. As explained in [47], trajectory tracking can be efficiently done with this type of controller, when the derivative action is sufficiently high. Nevertheless, the tracking error remains around 10 cm since a gain limit has been fixed to 1000 to prevent high command torques. Each controller is checked on a cyclic trajectory, followed at the angular rate ω_t.

Centralized controller



(a) Frequency responses and templates

(b) Joint and base trajectories

(c) Trajectory tracking

(d) Control torques & Base consumption

FIGURE 7 – Simulations results for the 1D example

For the centralized architecture, the arm controller includes the mass matrix $\mathbf{M_0}$ for one configuration of the trajectory, enabling a full decoupling and ensuring good performance for this very configuration. It takes the form :

$$K(s) = \mathbf{M_0} \left[\begin{array}{ccccc|ccccc} K_{p,1} & 0 & \dots & 0 & 0 & K_{d,1} & 0 & \dots & 0 & 0 \\ 0 & \ddots & & \vdots & 0 & 0 & \ddots & & \vdots & 0 \\ \vdots & & K_{p,i} & & \vdots & \vdots & & K_{d,i} & & \vdots \\ 0 & \dots & & \ddots & 0 & 0 & \dots & & \ddots & 0 \\ 0 & 0 & \dots & 0 & K_{p,N} & 0 & 0 & \dots & 0 & K_{d,N} \end{array} \right] \tag{35}$$

The synthesis is done using the `systune` function of the *Robust Control Toolbox* of `Matlab`. The frequency template are specified as « hard constraints », while the gain limitation and the consumption minimization are set by « soft constraints »[44].

To improve the robust performance of this controller, five different models obtained along the trajectory have been considered at the same time for the synthesis. The algorithm gives the gain setting in Tab. 1, and the frequency responses of each weighted transfer is illustrated in Fig. 8(a). It can be seen that the templates represented by two black curves are well-respected, for both the base and the manipulator.

Decentralized controller

For the decentralized architecture, each PD controller on the axis is simultaneously synthesized considering the whole arm dynamics. With this architecture, each of them only uses its local measure of

TABLE 1 – Gains setting for a **centralized** arm controller

Joint i	1	2	3	4	5	6
$K_{p,i}$	981.76	-1.09	1.94	0.44	3.33	0.22
$K_{d,i}$	301.32	45.89	34.40	47.33	131.36	377.70

position and velocity. The coupling between the joints and the base, or between the joints themselves, are actually considered directly in the synthesis since the linearized model is coupled. The synthesized controller is given by :

$$K(s) = \left[\begin{array}{ccccc|ccccc} K_{p,1} & 0 & \ldots & 0 & 0 & K_{d,1} & 0 & \ldots & 0 & 0 \\ 0 & \ddots & & \vdots & 0 & 0 & \ddots & & \vdots & 0 \\ \vdots & & K_{p,i} & & \vdots & \vdots & & K_{d,i} & & \vdots \\ 0 & \ldots & & \ddots & 0 & 0 & \ldots & & \ddots & 0 \\ 0 & 0 & \ldots & 0 & K_{p,N} & 0 & 0 & \ldots & 0 & K_{d,N} \end{array}\right] \tag{36}$$

The synthesis is done one more time with five different models along the trajectory, to obtain the gains setting given in Tab. 2. The frequency templates are mostly respected even if some peaks are degrading the H_∞-norm of the transfer, as shown in Fig. 9(a). Even though the controller is much more simpler, the trajectory tracking remains under a 10 cm error in Fig. 9(e).

TABLE 2 – Gains setting for a **decentralized** arm controller

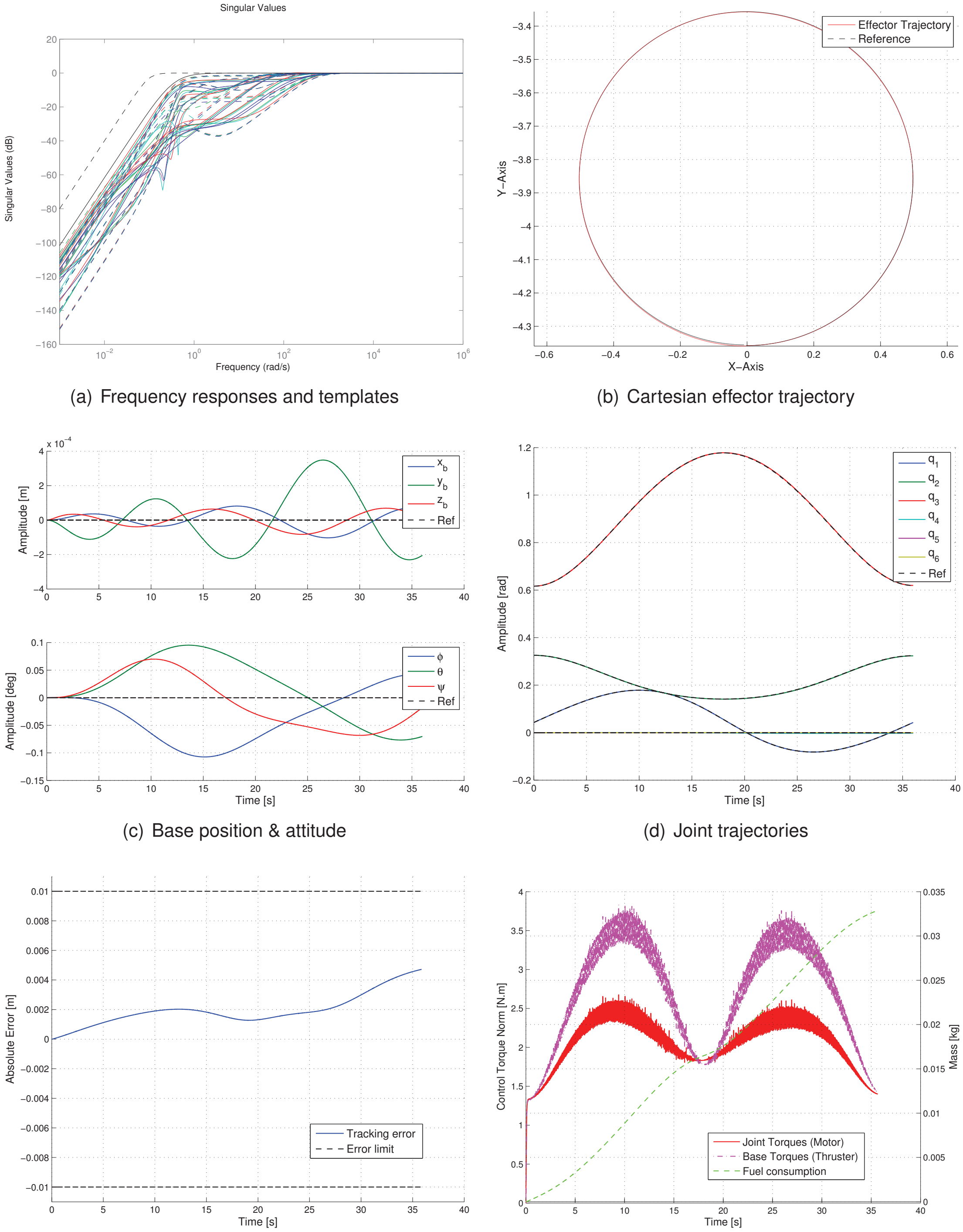
Joint i	1	2	3	4	5	6
$K_{p,i}$	107.51	406.89	106.96	98.78	1000	0.2167
$K_{d,i}$	1000	1000	1000	1000	1000	12.56

Results

To summarize, the use of a simple PD controller on each joint can be considered as a good compromise between easiness of implementation and tracking performance, according to the previous simulations results. It is clear that a centralized architecture enables a good trajectory tracking, under a 1 cm error, but at the expense of a high computational burden since the mass matrix needs to be evaluated in-line.

5. Conclusion

A space robot controller has been presented through this paper, to follow a predefined end-effector trajectory. To do so, a first part recalled the main results of classic robotics on which are based space robotics. An adaptation of the Newton-Euler algorithm enabled to compute the whole dynamic model, used in simulation. Then a H_∞ scheme has been developed to synthesize the spacecraft AOCS and the arm controller at the same time. Two main strategies were tested : either a centralized arm controller to improve decoupling on one configuration, or a decentralized one, to keep the structure as simple as possible. It has eventually been shown how this last very simple controller can provide a quite good trajectory tracking by using a multi-model based synthesis.



(a) Frequency responses and templates

(b) Cartesian effector trajectory

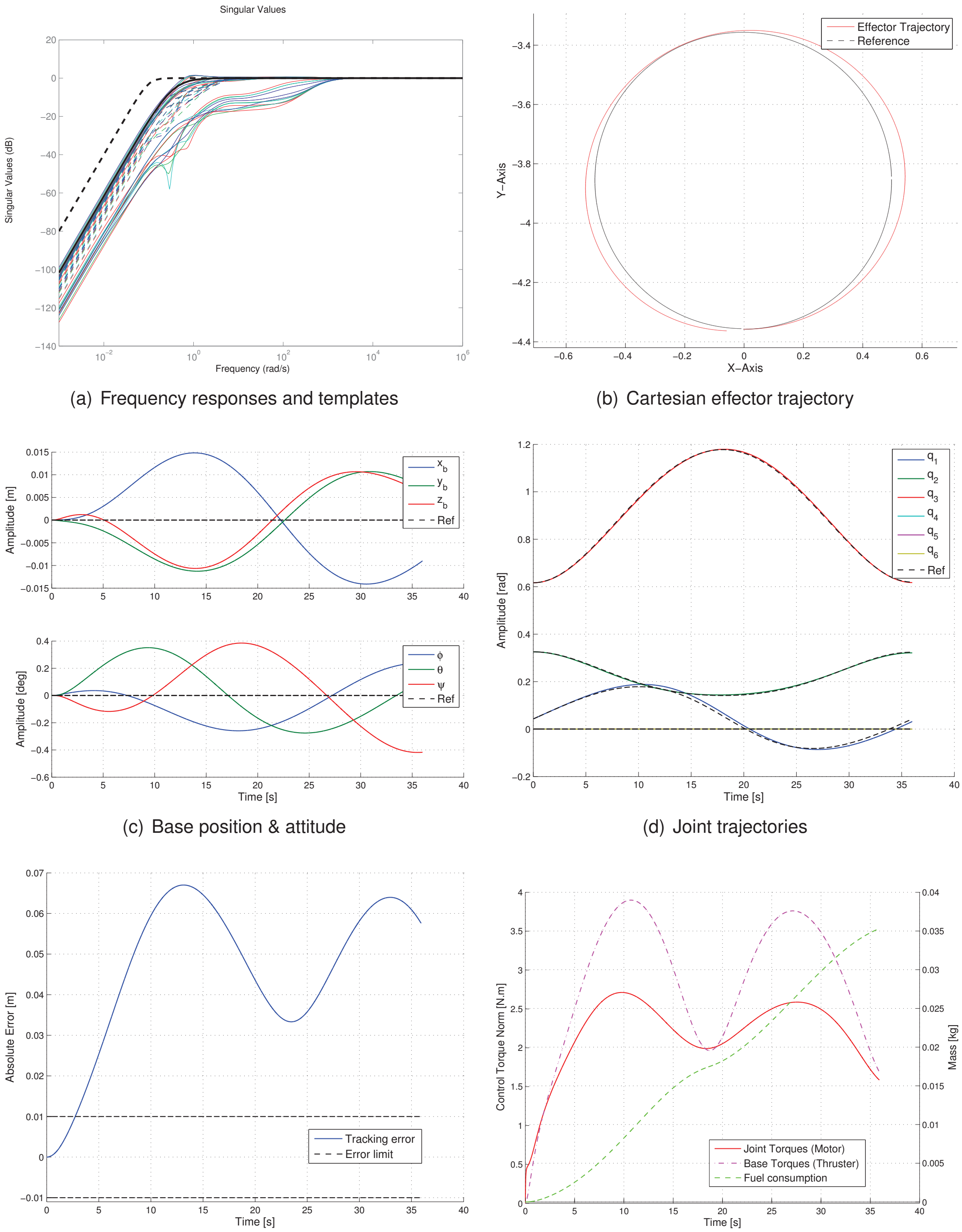
(c) Base position & attitude

(d) Joint trajectories

(e) Trajectory tracking

(f) Control torques & Base consumption

FIGURE 8 – **Centralized controller** - Simulations results for the complete space robot



(a) Frequency responses and templates

(b) Cartesian effector trajectory

(c) Base position & attitude

(d) Joint trajectories

(e) Trajectory tracking

(f) Control torques & Base consumption

FIGURE 9 – **Decentralized controller** - Simulations results for the complete space robot

6. Références

[1] Donald J. Kessler and Burton G. Cour-Palais. Collision Frequency of Artificial Satellites : The Creation of a Debris Belt. *Journal of Geophysical Research*, 83(A6) :2637–2646, 1978.

[2] J.-C. Liou. An active debris removal parametric study for LEO environment remediation. *Advances in Space Research*, 47 :1865–1876, June 2011.

[3] J.-C. Liou, N. L. Johnson, and N. M. Hill. Controlling the growth of future LEO debris populations with active debris removal. *Acta Astronautica*, 66(5-6) :648–653, March 2010.

[4] NASA Orbital Debris Program Office. Monthly Number of Objects in Earth Orbit by Object Type. *Orbital Debris Quaterly News*, 17(1) :8, 2013.

[5] Claude R. Phipps. A laser-optical system to re-enter or lower low Earth orbit space debris. *Acta Astronautica*, 93 :418–429, January 2014.

[6] Carmen Pardini, Toshiya Hanada, and Paula H. Krisko. Benefits and risks of using electrodynamic tethers to de-orbit spacecraft. *Acta Astronautica*, 64(5-6) :571–588, March 2009.

[7] M. Andrenucci, P. Pergola, and A. Ruggiero. Active Removal of Space Debris – Expanding foam application for active debris removal. Technical report, University of Pisa (Italy) - Aerospace Engineering Dept., 2011.

[8] Christophe Bonnal, Jean-Marc Ruault, and Marie-Christine Desjean. Active debris removal : Recent progress and current trends. *Acta Astronautica*, 85 :51–60, 2013.

[9] Bruce A. Aikenhead, Robert G. Daniell, and Frederick M. Davis. Canadarm and the space shuttle. *Journal of Vacuum Science & Technology A*, 1(2) :126 – 132, 1983.

[10] Noriyasu Inaba and Mitsushige Oda. Autonomous satellite capture by a space robot : world first on-orbit experiment on a japanese robot satellite ETS-VII. In *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, volume 2, pages 1169 – 1174, 2000.

[11] Robert B. Friend. Orbital Express program summary and mission overview. In *Sensors and Systems for Space Applications II*, volume 6958, pages 1–11. Howard, Richard T. Motaghedi, Pejmun, 2008.

[12] T. Boge, T. Wimmer, O. Ma, and T. Tzschichholz. EPOS - Using robotics for RvD simulation of on-orbit servicing missions. In *AIAA Modeling and Simulation Technologies Conference*, volume 1, pages 1–15, Toronto, Canada, 2010.

[13] Feng Wang, Fuchun Sun, and Huaping Liu. Space robot modeling and control considering the effect of orbital mechanics. In *1st International Symposium on Systems and Control in Aerospace and Astronautics*, volume 1, pages 193 – 198, 2006.

[14] Mark W. Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, New York, 2006.

[15] Yoji Umetani and Kazuya Yoshida. Workspace and Manipulability Analysis of Space Manipulator. *Transactions of the Society of Instrument and Control Engineers*, E-1(1) :8, 2001.

[16] Evangelos Papadopoulos and Steven Dubowsky. On the dynamic singularities in the control of free-floating manipulators. In *ASME Winter Annual Meeting*, volume 15, pages 45–52, 1989.

[17] Steven Dubowsky and Evangelos Papadopoulos. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *IEEE Transactions on Robotics and Automation*, 9(5) :531–543, 1993.

[18] Yoji Umetani and Kazuya Yoshida. Resolved motion rate control of space robotic manipulators with generalized jacobian matrix. *IEEE Transactions on Robotics and Automation*, 5(3) :303–314, 1989.

[19] Kostas Nanos and Evangelos Papadopoulos. On Cartesian Motions with Singularities Avoidance for Free-floating Space Robots. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 5398–5403, 2012.

[20] Yoshihiko Nakamura, Ranjan Mukherjee, and Santa Barbara. Nonholonomic path planning of space robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1050–1055, 1989.

[21] Wayne J. Book. Recursive Lagrangian dynamics of flexible manipulator arms. *The International Journal of Robotics Research*, 3(3) :87–101, 1984.

[22] Craig R. Carignan and David L. Akin. The reaction stabilization of on-orbit robots. *IEEE Control Systems*, 20 :19 – 33, 2000.

[23] S. Ali A. Moosavian and Evangelos Papadopoulos. Explicit dynamics of space free-flyers with multiple manipulators via SPACEMAPLE. *Advanced Robotics*, 18(2) :223–244, 2004.

[24] Dragomir N. Nenchev. Reaction Null Space of a multibody system with applications in robotics. *Mechanical Sciences*, 4(1) :97–112, February 2013.

[25] G. Rodriguez, A. Jain, and K. Kreutz-Delgado. A Spatial Operator Algebra for Manipulator Modeling and Control. *The International Journal of Robotics Research1*, 10(4) :371–381, 1991.

[26] S. K. Saha. Dynamics of serial multibody systems using the decoupled natural orthogonal complement matrices. *Journal of applied mechanics*, 66(Dec.) :986–996, 1999.

[27] Ashish Mohan and S. K. Saha. A recursive, numerically stable, and efficient simulation algorithm for serial robots. *Multibody System Dynamics*, 17(4) :291–319, February 2007.

[28] S. Ali A. Moosavian and Evangelos Papadopoulos. Free-flying robots in space : An overview of dynamics modeling, planning and control. *Robotica*, 25(05) :537–547, March 2007.

[29] Evangelos Papadopoulos and Steven Dubowsky. On the nature of control algorithms for free-floating space manipulators. *IEEE Transactions on Robotics and Automation*, 7(6) :750–758, 1991.

[30] Mitsushige Oda. On the dynamics and control of ETS-7 satellite and its robot arm. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 1586 – 1593, 1994.

[31] Mitsushige Oda. Coordinated control of spacecraft attitude and its manipulator. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 732–738, 1996.

[32] Steven Dubowsky, Evelyn E. Vance, and Miguel A Torres. The Control of Space Manipulators Subject to Spacecraft Attitude Control Saturation Limits. In *NASA Conference on Space Telerobotics*, volume IV, pages 409–418, 1989.

[33] Evangelos Papadopoulos and Steven Dubowsky. Coordinated manipulator/spacecraft motion control for space robotic systems. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1696–1701, 1991.

[34] Z. Vafa and S. Dubowsky. On the dynamics of manipulators in space using the virtual manipulator approach. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 579–585, 1987.

[35] Kazuya Yoshida and Yoji Umetani. Control of space free-flying robot. In *29th Conference on Decision and Control*, volume 1, pages 97–102, 1990.

[36] P. Piersigilli, I. Sharf, and A.K. Misra. Reactionless capture of a satellite by a two degree-of-freedom manipulator. *Acta Astronautica*, 66(1-2) :183–192, January 2010.

[37] Thai-Chau Nguyen-Huynh and Inna Sharf. Adaptive reactionless motion for space manipulator when capturing an unknown tumbling target. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 4202 – 4207, 2011.

[38] Farhad Aghili. A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics. *IEEE Transactions on Robotics*, 28(3) :634 – 649, 2012.

[39] John J. Craig. *Introduction to Robotics : Mechanics and Control*. Addison-Wesley Publishing Company, 2nd edition, 1989.

[40] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer US, Boston, 2008.

[41] Roy Featherstone and David Orin. Robot dynamics : Equations and algorithms. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 826 – 834, San Francisco, 2000.

[42] Pierre Apkarian and Dominikus Noll. Nonsmooth h_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1) :71–86, 2006.

[43] Pascal Gahinet and Pierre Apkarian. Structured h_∞ synthesis in {MATLAB}. In *IFAC World Congress*, volume 18, pages 1435–1440, 2011.

[44] Pierre Apkarian. Tuning Controllers Against Multiple Design Requirements. In *American Control Conference*, volume 1, pages 3888–3893, 2013.

[45] D. Alazard, T. Loquen, H. de Plinval, and C. Cumer. Avionics/Control co-design for large flexible space structures. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, pages 1–15. AIAA, 2013.

[46] Daniel Alazard. *Reverse engineering in control design*. ISTE / John Wiley & Sons, London / Hoboken, 2013.

[47] Sadao Kawamura, Fumio Miyazaki, and Suguru Arimoto. Is a local linear PD feedback control law effective for trajectory tracking of robot motion ? In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1335–1340, 1988.