POLITECNICO DI TORINO

Master's Degree in Aerospace Engineering

Master's Degree Thesis

Implementation and analysis of adaptive control strategies for the on-orbit capture of target with a Space Manipulator System

Supervisors

Prof. BATTIPEDE Manuela
Dr. KRAÏEM Sofiane
Dr. ROGNANT Mathieu

Candidate

MERCADANTE Pier Luigi

December 2025

Abstract

The rapid growth of space exploration industry and the increasing need for debris mitigation missions have led to rising interest in Active Debris Removal (ADR) and On-Orbit Servicing (OOS) operations. Among robotic systems, Space Manipulator Systems (SMS) stand out for their adaptability and versatility, making them the preferred solution for OOS tasks such as satellite grasping, berthing, repair, and maintenance. Consequently, the design of proper strategies to control them have become critical. However, these control laws are often ineffective due to the strongly coupled dynamics of space systems and the difficulty in estimating the characteristics of the target. This Master's thesis aims to find new control strategies that contribute to address this problem, by proposing a robust control against the disturbances and perturbations that derive from the capture of an unknown target. A new dynamics model is proposed, in order to handle the momentum transfer during post-capture stabilization. Two different control strategies will be investigated, and their results will be compared. The first one is based on a combination of Nonlinear Dynamics Inversion (NDI) and Nonlinear Disturbance Observer (NDO), while the second one consists of NDI and Model Reference Adaptive Control (MRAC). A gain synthesis is applied to the two strategies. Overall, both the proposed methods proved capable of efficiently counteracting the disturbance introduced by the target, allowing to compensate for the perturbation and achieve a robust control of the system. However, the focus of the thesis will be on the MRAC strategy, supported by a low-pass filter to suppress high-frequency MRAC oscillations. Two different options will be provided for NDI formulation, and MRAC will be adapted accordingly. Finally, a workspace analysis will be conducted to bound some variables useful for gains synthesis, which is performed exploiting Linear Matrix Inequalities (LMI). The results achieved show how adaptive control can significantly improve the performance of space manipulator systems, paving the way for their increasingly widespread use.

Table of Contents

List of Tables

List of Figures

Acronyms

ADR

Active Debris Removal

CoM

Center of Mass

DoF

Degrees of Freedom

EE

End Effector

LMI

Linear Matrix Inequalities

MRAC

Model Reference Adaptive Control

MSE

Mean Squared Error

NDI

Non-linear Dynamic Inversion

NDO

Non-linear Disturbance Observer

OOS

On-Orbit Servicing

PE

Persistent Excitation

RF

Reference Frame

SMS

Space Manipulator System

List of Symbols

C :	Centrifugal/Coriolis matrix	
δ :	Momentum (both angular and linear)	[kg·m/s or kg·m^2/s]
e :	Error	
F :	Force	[N]
H :	Inertia matrix	
I :	Moment of inertia	[Kg m^2]
J :	Jacobian matrix	
Λ :	Uncertainties in control effectiveness	
m :	Mass	[Kg]
η :	Generalized modal displacement vector	
P :	Lyapunov's matrix	
q :	Displacements of robot degrees of freedom	[m or rad]
$\dot{q}$:	Velocities of robot degrees of freedom	[m/s or rad/s]
$\ddot{q}$:	Accelerations of robot degrees of freedom	[m/s^2 or rad/s^2]
q_0 :	Base pose vector	[m and rad]
R :	Rotation matrix	
r :	3-D position vector in the inertial frame	[m]
σ :	MRAC regressor	
τ :	Torque	[Nm]

θ :	Euler angles	[rad]
t :	Twist vector	[m/s] and [rad/s]
ω :	Angular velocity	[rad/s]
V :	Lyapunov's function	
v :	Linear velocity	[m/s]
W :	Weights matrix	
x :	State vector	
$\mathbf{I}_N$:	Identity matrix of size NxN	
0_{NxM} :	Zeros matrix of size NxM	
X(:,i):	i-th column of generic matrix X	
$\lVert X \rVert$:	Euclidean norm of generic matrix X	
$\lVert X \rVert_F$:	Frobenius norm of generic matrix X	
$(\cdot)^x$:	Skew-symmetric matrix operator	
$\dot{(\cdot)}$:	Time derivative operator	
$(\cdot)_r$:	Reference value	
$(\cdot)_d$:	Desired value	
$\lambda(\cdot)$:	Eigenvalue operator	
$\underline{\lambda}(\cdot)$:	Minimum eigenvalue operator	
$\overline{\lambda}(\cdot)$:	Maximum eigenvalue operator	
$\tilde{\lambda}(\cdot)$:	Singular value operator	

Introduction

The significant versatility of Space Manipulator Systems (SMS) has made them suitable for multiple purposes and missions. These are systems based on a robotic multi-joints arm mounted on a base. They can adapt to accomplish a very wide range of functions, such as satellite grasping, berthing, repair, and maintenance. Space exploration increasingly rely on this kind technology, with the goal of extending the operational capability of spacecrafts beyond their traditional limits. Despite their growing importance and undoubted utility, the control of these systems introduces unique and tough challenges, primarily due to the complex and strongly coupled dynamics of space systems [1, 2]. The choice of the most effective control strategy is strongly dependent on the physical and operational scenario, including factors such as manipulator flexibility, number of arms, grasping mechanisms, and the cooperativeness of the target [3, 4].

In order to achieve a smooth capture, the velocity matching between the end-effector and the target must be ensured. If the target is non-cooperative and tumbling, a strong coupling effect between the robotic manipulator and the servicer base arises [5, 6]. Various strategies and methods have been explored to tackle this challenge. In [7], grasping and detumbling have been formulated as nonlinear optimization problems, while Aghili [8] proposed a method based on minimum-energy trajectory tasks to dissipate the angular momentum of the captured target. Furthermore, reactionless manipulation using nullspace projections was exploited for both ADR and OOS missions [6, 9, 10].

Improving the trajectory planning algorithms is not enough, since high robustness is required to compensate for the perturbations due to the capture, such as contact forces and inertia variations. Synchronizing manipulator and base motions throughout the approach and the capture maneuver becomes a complex problem and needs to be tackle in the guidance law [11]. In [9], the problem is addressed through a multi-phase strategy: bias momentum control for the approach, impedance control at impact, and distributed momentum control post-capture. Another step forward in this direction has been taken by the COMRADE project, which ensured high robustness against perturbations by the introduction of compliance control and $\mathcal{H}_\infty$ synthesis [12]. In order to stabilize the servicer base, new techniques based on hierarchical task planning and nullspace control have been investigated [10]. In [13], a robust joint-space control method is introduced for rotation-free-floating SMS with flexible appendages.

More recently, predictive control strategies have gained attention for their ability to generate optimal interaction trajectories while accounting for dynamic constraints and uncertainties. Initial efforts focused on linearized models to reduce computational load. In particular, Successive Linearization-based MPC (SL-MPC) methods [14, 15] employ piecewise-affine approximations of the nonlinear dynamics, enabling faster optimization for real-time feasibility. However, these methods are constrained by limited prediction horizons, which can degrade both the stability and performance of the controller [15]. In [16], a linear tube-based MPC was proposed to enhance robustness against target motion uncertainties.

To address this problem and the inability of conventional controllers to compensate for varying loads acting on the system end-effector, adaptive control methods were proposed. They can be divided into three classes: model reference, self-tuning and gain-scheduled. The characteristic they all share is that they can automatically adjust their parameters in real time to handle uncertainties or perturbations in the robot dynamics and environment. These strategies differ from classical controllers because they learn and adapt while the system is operating. Among them, Model Reference Adaptive Control (MRAC) approach stands out for its computational simplicity and ability to reliably follow an ideal reference.

In [17], an overview on MRAC applied to robotics mechanisms is presented, highlighting the advantages of this method, but also introducing some related problems, and how other papers have dealt with them. Yucelen [18] discussed the mathematical fundamentals for MRAC implementation and Lyapunov's stability proof. However, classic MRAC can not tolerate large system uncertainties. Dogan et al. [19] proposed an adaptive robustifying strategy to relax the stability limit through the introduction of an additional term in MRAC formulation under specific conditions.

In order to achieve optimal performances and maintain the same system response to time-varying perturbation and command profile, the concept of scalability can be introduced. For this purpose, a new adaptive weight update law and a new Lyapunov's function can be built, by exploiting a time-varying scalable learning rate [20]. Non-constant adaptation rates are considered also in [21], where a combination of MRAC and Integrative Dynamic Regressor Extension and Mixing (I-DREM) methods is proposed. This allows to replace the manual selection of the adaptation rate with an automatic online scaling law based on current regressor value, while ensuring exponential monotonic convergence of the controller error. Another way of dealing with time-varying command profiles is proposed in [22], by developing a new term within the MRAC formulation through a gradient descent approach

and a minimum error criterion. This new term allows to modify both the reference model and the control signal.
Despite the effectiveness of MRAC methods, it is often difficult to ensure *a-priori* clear performance guarantees. This challenge is addressed by Arabi et al. [23] through the development of a set-theoretic MRAC, which leverages on generalized restricted potential functions to bound the controller error. This method, also applicable to nonlinear reference systems, allows to achieve high reliability, by forcing the system to evolve within specified boundaries. Another cause of deviation of the system from the ideal reference behavior is due to high-frequency oscillations caused by the high gains necessary for a fast convergence. In [24], a low-pass filter is applied to weights update law estimation, in order to dampen the oscillations while preserving asymptotic stability.
The goal of this thesis is to leverage the knowledge and achievements of all these papers to propose new innovative control strategies. Those will be applied to the control of a Space Manipulator System during the phase of capture of an unknown target, that introduces strong perturbations into the system dynamics.

Chapter 1

Modeling

In this first chapter, the mathematical models used to describe the dynamics of the SMS will be explained. First of all, an overview of the utilized robot will be provided, together with its operational profile. After that, the models that describe the general SMS dynamcis and the target capture phase will be shown and the chapter will end with the presentation of the Matlab tools that allow the implementation in the software environment.

1.1 Robot model and operational context

In this section, the main features of the robot used for all the analysis and simulations of this chapter will be described and some information about its way of operating will be provided.
The SMS (Space Manipulator System) consists of a base with a mass of 100 kg, on which 3 reaction wheels are mounted: these represent 3 DoFs in robot motion. On the base is also mounted a robotic arm, characterized by 3 revolut joints, which represent 3 additional DoFs. The manipulator ends with an end effector (EE), that is the component responsible of directly getting in touch with and grasping the target. The goal of the proposed control strategy will be to act on the 6 DoFs through proper control torques in order to make the SMS follow the desired profile of positions and velocities that lead to a safe target capture.
These profiles are obtained through specific path-following functions that act on the motion of the joints, since the control strategies presented are designed to work in the Joint Space.
The capture was initially simulated by suddenly increasing both mass and inertia of the EE to take into account the addition of the target to the robot model. It is a simplified way of simulating the target grasping, but it provides acceptable results. However, a more reliable capture model will be developed further in this chapter.

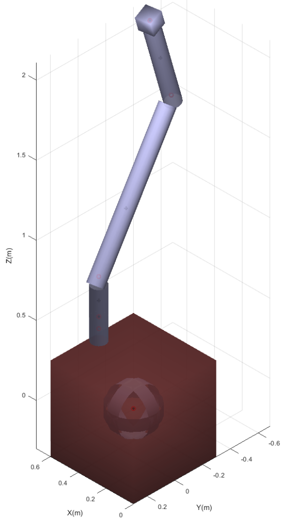

Figure 1.1: Simplified representation of the robot

1.2 Pre-capture Dynamics

Prior to capture, the servicer dynamics can be expressed as [5]:

$$\underbrace{\begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{0q} \\ \mathbf{H}_{0q}^T & \mathbf{H}_q \end{bmatrix}}_{H_s} \begin{bmatrix} \ddot{\mathbf{q}}_0 \\ \ddot{\mathbf{q}} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{0q} \\ \mathbf{C}_{q0} & \mathbf{C}_q \end{bmatrix}}_{C_s} \begin{bmatrix} \dot{\mathbf{q}}_0 \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{6\times1} \\ \boldsymbol{\tau}_q \end{bmatrix} \tag{1.1}$$

where $q \in \mathbb{R}^{n_q \times 1}$ represents the displacements of the DoFs in the kinematic chain, $H \in \mathbb{R}^{n_q \times n_q}$ represents the inertia matrix, $C \in \mathbb{R}^{n_q \times n_q}$ represents the convective matrix that collects Coriolis and centrifugal effects and $\tau_q \in \mathbb{R}^{n_q \times 1}$ represents the control torque vector. The total number of DoFs is $n_q = n_r + n_m$, where n_r is the number of reaction wheels and n_m is the number of manipulator joints.
Since no external forces or torques are applied to the servicer in this phase, its momentum is constant and can be considered as zero:

$$\delta_s = H_0\dot{q}_0 + H_{0q}\dot{q} = 0 \tag{1.2}$$

It follows:

$$\dot{q}_0 = -H_0^{-1}H_{0q}\dot{q} \tag{1.3}$$

Developing the first row of (1.1):

$$H_0\ddot{q}_0 + H_{0q}\ddot{q} + C_0\dot{q}_0 + C_{0q}\dot{q} = 0_{6x1} \tag{1.4}$$

Substituting (1.3) in (1.4), an expression for $\ddot{q}_0$ can be found:

$$\ddot{q}_0 = -H_0^{-1} H_{0q} \ddot{q} + H_0^{-1} (C_0 H_0^{-1} H_{0q} - C_{0q}) \dot{q} \tag{1.5}$$

Developing the second row of (1.1):

$$H_{0q}^T \ddot{q}_0 + H_q \ddot{q} + C_{q0} \dot{q}_0 + C_q \dot{q} = \tau_q \tag{1.6}$$

Inserting (1.3) and (1.5) into (1.6), it leads to the general equation for SMS dynamics in the joint space:

$$\underbrace{(H_q - H_{0q}^T H_0^{-1} H_{0q})}_{H^\diamond} \ddot{q} + \underbrace{[C_q - (C_{q0} - H_{0q}^T H_0^{-1} C_0) H_0^{-1} H_{0q} - H_{0q}^T H_0^{-1} C_{0q}]}_{C^\diamond} \dot{q} = \tau_q \tag{1.7}$$

This represents the SMS dynamics in the absence of disturbances and it needs to be properly modified when a perturbation occurs, like the capture of a target.

1.3 Capture Model

The target capture (i.e. the perturbation) can be simulated by increasing the mass and inertia of the EE of the unperturbed robot model. However, in order to better reproduce the capture, the simulations were improved by considering a new robot model, which precisely represents the target. Therefore, 3 different robots are used within the simulation: one represents the unperturbed system pre-capture (*robot.pre*), another one represents the target alone (*robot.target*) and the final one represents the assembly of servicer and target after capture (*robot.post*). By exploiting these three model, a more precise representation of target capture can be simulated.

Moreover, the new model will be developed starting from the conservation of momentum for the global system (robot + target): in fact, since no external forces or torques act on this system, its momenta (both angular and linear) will be constant. This consideration allows to have a capture model that results be physically valid and coherent.

In the following sections, the mathematical formulations that rule the model will be explained and some simulations will be shown to validate the model.

1.3.1 Post-capture dynamics

In the post-capture dynamics, the internal torques and forces that come from the interaction between the target and the EE must be taken into account:

$$H_s \begin{bmatrix} \ddot{q}_0 \\ \ddot{q} \end{bmatrix} + C_s \begin{bmatrix} \dot{q}_0 \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0_{6\times1} \\ \tau_q \end{bmatrix} - \begin{bmatrix} J_{0t} & J_{qt} \end{bmatrix}^T F_e \tag{1.8}$$

where $[]_s$ means that the variables are referred to the servicer and F_e represents the internal forces and torques considered as applied in the target center of mass. We can define $J_t = \begin{bmatrix} J_{0t} & J_{qt} \end{bmatrix}$ and $\dot{J}_t = \begin{bmatrix} \dot{J}_{0t} & \dot{J}_{qt} \end{bmatrix}$, where J_{0t} is the Jacobian matrix from manipulator base to target and J_{qt} is the one from joint space to target.
The target dynamics after capture is:

$$H_t \dot{t}_t + C_t t_t = F_e \tag{1.9}$$

where $[]_t$ means that the variables are referred to the target and $t_t = \begin{bmatrix} \omega_t \\ v_t \end{bmatrix}$ is the target twist.
The following relations can be used to link the target twist to joint space velocities:

$$t_t = J_{0t} \dot{q}_0 + J_{qt} \dot{q} \tag{1.10}$$

$$\dot{t}_t = \dot{J}_{0t} \dot{q}_0 + \dot{J}_{qt} \dot{q} + J_{0t} \ddot{q}_0 + J_{qt} \ddot{q} \tag{1.11}$$

Substituting these equations into (1.9):

$$F_e = (H_t \dot{J}_t + C_t J_t) \begin{bmatrix} \dot{q}_0 \\ \dot{q} \end{bmatrix} + H_t J_t \begin{bmatrix} \ddot{q}_0 \\ \ddot{q} \end{bmatrix} \tag{1.12}$$

Substituting this last equation into (1.8):

$$\underbrace{(H_s + J_t^T H_t J_t)}_{\tilde{H}} \begin{bmatrix} \ddot{q}_0 \\ \ddot{q} \end{bmatrix} + \underbrace{(C_s + J_t^T H_t \dot{J}_t + J_t^T C_t J_t)}_{\tilde{C}} \begin{bmatrix} \dot{q}_0 \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0_{6\times1} \\ \tau_q \end{bmatrix} \tag{1.13}$$

In this equation, H_s and C_s matrices are computed with reference to *robot.pre*, while the Jacobian matrices, H_t and C_t refer to *robot.target*. In particular, H_t and C_t represent the target inertia and centrifugal/Coriolis matrices computed in the target reference frame:

$$H_t = \begin{bmatrix} I_t & 0_{3x3} \\ 0_{3x3} & m_t \mathbf{I}_3 \end{bmatrix} \tag{1.14}$$

$$C_t = \begin{bmatrix} \omega_t^x I_t & m_t v_t^x \\ m_t v_t^x & m_t \omega_t^x \end{bmatrix} \tag{1.15}$$

where m_t and I_t represent target mass and moment of inertia, $[]^x$ represents the skew-symmetric matrix operator and ω_t and v_t represent target angular and linear velocities.
Defining $\tilde{H} = (H_s + J_t^T H_t J_t)$ and $\tilde{C} = (C_s + J_t^T H_t \dot{J}_t + J_t^T C_t J_t)$, the following equation for the post-capture dynamics of the system consisting of servicer + target can be defined:

$$\tilde{H} \begin{bmatrix} \ddot{q}_0 \\ \ddot{q} \end{bmatrix} + \tilde{C} \begin{bmatrix} \dot{q}_0 \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0_{6\times1} \\ \tau_q \end{bmatrix} \tag{1.16}$$

1.3.2 Conservation of momentum

To obtain a reliable model throughout the whole capture maneuver, the conservation of total system momentum (both angular and linear) can be exploited. The process is based on the following assumptions:

- before capture, the target has constant angular velocities and zero linear velocities;
- before capture, the servicer momentum can be assumed to be zero;
- before capture, since no external forces are applied, the momenta of the servicer and the target are constant;
- during and after capture, since the contact forces are applied, the momenta of both servicer and target vary;
- during and after capture, since the contact forces are internal to the system consisting of servicer + target, the global system momentum stays constant.

The first assumption derives from operational considerations, since one possible function of the SMS is to capture and detumble a rotating target. However, the capture model will be proved to be valid even in presence of target initial linear velocites in one of the following sections.
Consequently, the momenta of servicer and target before capture in the inertial reference frame are:

$$\delta_s = H_0\dot{q_0} + H_{0q}\dot{q} = 0 \tag{1.17}$$

$$\delta_t = \begin{bmatrix} I_t\omega_t \\ 0_{3x1} \end{bmatrix} = M_t = constant \tag{1.18}$$

where the matrices H_0 and H_{0q} derives from a block subdivision of H_s:

$$H_s = \begin{bmatrix} H_0 & H_{0q} \\ H_{0q}^T & H_q \end{bmatrix}$$

The same subdivision is applied to matrix C_s:

$$C_s = \begin{bmatrix} C_0 & C_{0q} \\ C_{q0} & C_q \end{bmatrix}$$

After capture, the momenta can be expressed in the inertial reference frame in the following forms:

$$\delta_s = R_0^*(H_0\dot{q_0} + H_{0q}\dot{q}) \tag{1.19}$$

$$\delta_t = R_t^* t_t \tag{1.20}$$

where the transformation matrices are defined as:

$$R_0^* = \begin{bmatrix} R_0 & 0_{3x3} \\ 0_{3x3} & \mathbf{I}_3 \end{bmatrix} \tag{1.21}$$

$$R_t^* = \begin{bmatrix} I_t & m_t \vec{r}_{0t}^{\,x} \\ 0_{3x3} & m_t\mathbf{I}_3 \end{bmatrix} \tag{1.22}$$

R_0 is the rotation matrix of the servicer base with respect to the inertial frame, I_t is the target inertia matrix in the inertial frame, m_t is the target mass and $\vec{r}_{0t}$ is the position of the target CoM with respect to the base CoM.
As a result, the total momentum of the system (servicer + target) in the inertial reference frame is:

$$\delta = \delta_s + \delta_t = R_0^*(H_0\dot{q}_0 + H_{0q}\dot{q}) + R_t^* t_t = R_0^*(\tilde{H}_0\dot{q}_0 + \tilde{H}_{0q}\dot{q}) = M_t \tag{1.23}$$

Thanks to the conservation of momentum, it results to be equal to the target momentum before capture.
By substituting (1.10) within (1.23):

$$\dot{q}_0 = \underbrace{-(R_0^*H_0 + R_t^*J_{0t})^{-1}(R_0^*H_{0q} + R_t^*J_{qt})}_{A}\,\dot{q} + \underbrace{(R_0^*H_0 + R_t^*J_{0t})^{-1}}_{B}\,M_t \tag{1.24}$$

Since no external forces or torques are applied to the system, its total momentum stays constant, that means that its time-derivative is equal to 0:

$$\dot{R}_0^*(H_0\dot{q}_0 + H_{0q}\dot{q}) + R_0^*(\dot{H}_0\dot{q}_0 + H_0\ddot{q}_0 + \dot{H}_{0q}\dot{q} + H_{0q}\ddot{q}) + \dot{R}_t^* t_t + R_t^*\dot{t}_t = 0 \tag{1.25}$$

where

$$\dot{R}_0^* = \begin{bmatrix} \vec{\omega}_0^{\,x} R_0 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} \end{bmatrix}$$

$$\dot{R}_t^* = \begin{bmatrix} \dot{I}_t & m_t[\vec{v}_t - \vec{v}_0]^x \\ 0_{3x3} & 0_{3x3} \end{bmatrix}$$

By substituting (1.10), (1.11) and (1.24) within (1.25):

$$\begin{aligned}
\ddot{q}_0 =& \underbrace{-(R_0^* H_0 + R_t^* J_{0t})^{-1}(R_0^* H_{0q} + R_t^* J_{qt})}_{D}\,\ddot{q} \\
&-(R_0^* H_0 + R_t^* J_{0t})^{-1}[\underbrace{(\dot{R}_0^* H_0 + R_0^* \dot{H}_0 + \dot{R}_t^* J_{0t} + R_t^* \dot{J}_{0t})A}_{E_1} \\
&+\underbrace{(\dot{R}_0^* H_{0q} + R_0^* \dot{H}_{0q} + \dot{R}_t^* J_{qt} + R_t^* \dot{J}_{qt})}_{E_2}]\dot{q} \\
&\underbrace{-(R_0^* H_0 + R_t^* J_{0t})^{-1}(\dot{R}_0^* H_0 + R_0^* \dot{H}_0 + \dot{R}_t^* J_{0t} + R_t^* \dot{J}_{0t})B}_{F}\,M_t = \\
=&D\ddot{q}\underbrace{-(R_0^* H_0 + R_t^* J_{0t})^{-1}[E_1 + E_2]}_{E}\,\dot{q} + FM_t = \\
=&D\ddot{q} + E\dot{q} + FM_t
\end{aligned} \tag{1.26}$$

Developing the first row of (1.16):

$$\tilde{H}_0\ddot{q}_0 + \tilde{H}_{0q}\ddot{q} + \tilde{C}_0\dot{q}_0 + \tilde{C}_{0q}\dot{q} = 0_{6x1}$$

Substituting the equations (1.24) and (1.26) within that, an expression for M_t can be obtained:

$$M_t = \underbrace{-(\tilde{H}_0 F + \tilde{C}_0 B)^{-1}(\tilde{H}_0 D + \tilde{H}_{0q})}_{G}\,\ddot{q}\,\underbrace{-(\tilde{H}_0 F + \tilde{C}_0 B)^{-1}(\tilde{H}_0 E + \tilde{C}_0 A + \tilde{C}_{0q})}_{K}\,\dot{q} \tag{1.27}$$

Exploiting (1.27) inside (1.24) and (1.26), the dynamics of the base can be represented as a function of only joints velocities and accelerations:

$$\dot{q}_0 = (A + BK)\dot{q} + BG\ddot{q} \tag{1.28}$$

$$\ddot{q}_0 = (D + FG)\ddot{q} + (E + FK)\dot{q} \tag{1.29}$$

Developing this time the second row of (1.16):

$$\tilde{H}_{q0}\ddot{q}_0 + \tilde{H}_q\ddot{q} + \tilde{C}_{q0}\dot{q}_0 + \tilde{C}_q\dot{q} = \tau_q$$

By substituting (1.28) and (1.29) into this equation, the dynamics of the system can be expressed as a function of joints motion, removing any explicit reference to the base:

$$\underbrace{[\tilde{H}_{q0}(D + FG) + \tilde{C}_{q0}BG + \tilde{H}_q]}_{\tilde{H}^\diamond}\,\ddot{q} + \underbrace{[\tilde{H}_{q0}(E + FK) + \tilde{C}_{q0}(A + BK) + \tilde{C}_q]}_{\tilde{C}^\diamond}\,\dot{q} = \tau_q \tag{1.30}$$

The equation obtained results to be in the same form of the general pre-capture dynamics equation (1.7).

1.3.3 Validation in open-loop

In order to prove the validity of the capture model, an open-loop simulation was performed: a simple step control was applied to the robot (Figure 1.2) and the evolution of the system dynamics was observed, aiming at coherence in motion and conservation of momenta. The capture occurs after 1000 seconds and the target is characterized by a mass of 8 kg (small with respect to 100 kg of the base), zero initial linear velocities and initial angular velocities of 5e-5 along all the three axes. Since a random control was applied, without any kind of reference motion, the joints and the EE do not follow a particular profile, but still result to be physically coherent, as shown in Figures 1.3 and 1.4.

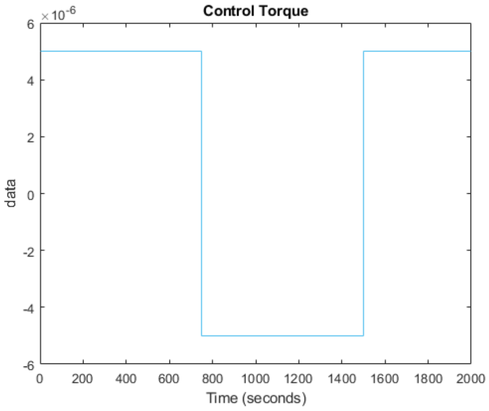

Figure 1.2: Capture validation: control torque

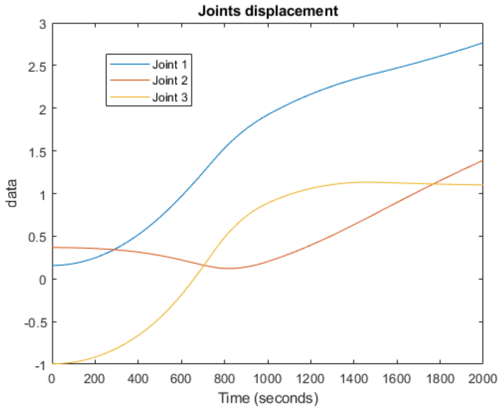

Figure 1.3: Capture validation: joints displacements

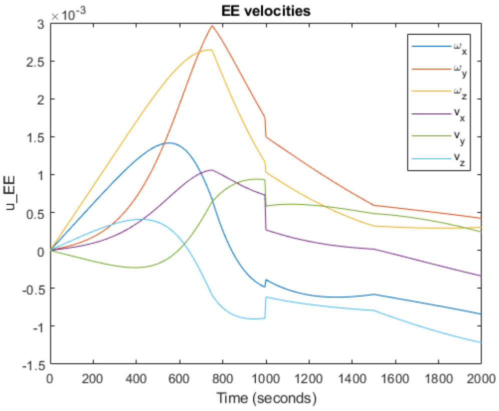

Figure 1.4: Capture validation: EE velocities

The moment of capture can be clearly identified looking at EE velocities, characterized by a step in all the components. Outside of this disturbance, the motion results to be physically acceptable and coherent with the applied control torques. In order to validate the model, the conservation of global system momentum needs to be verified. As shown in the graphs below, all the assumptions and conditions related to momentum conservation are respected: before capture, the servicer has constant zero momenta and the target has constant angular momenta and zero linear momenta; after capture, both servicer and target momenta vary, but the global system momenta given by their sum stay constant throughout the whole simulation.

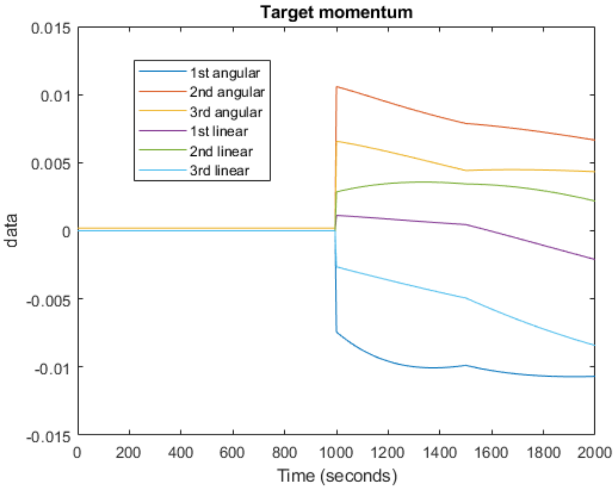

Figure 1.5: Capture validation: target momenta

It is evident how the target momenta before capture are much smaller than after capture, when the interaction between robot and target determines a big variation of all the momenta. In future simulations, one of the goals will be to reduce these variations in order to have a smoother capture, by trying to reach the capture moment with EE velocities closer to the target ones.

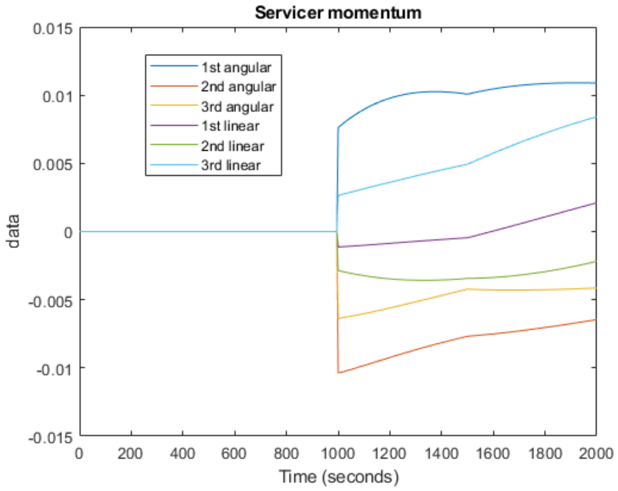

Figure 1.6: Capture validation: servicer momenta

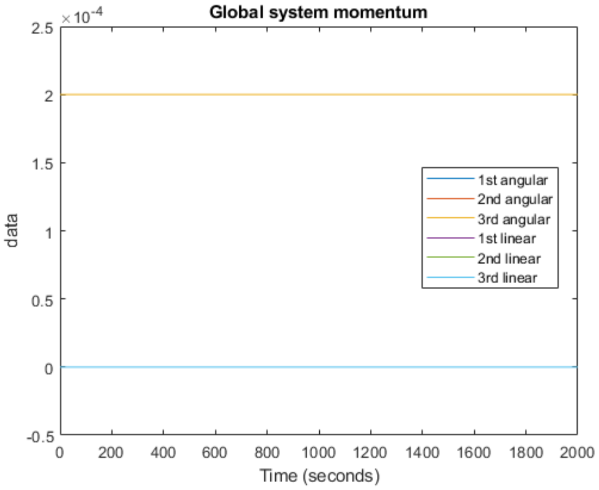

Figure 1.7: Capture validation: global system momenta

1.3.4 Validation with target initial linear velocities

The capture model was created considering a rotating target with no initial linear velocities. However, it can be proved that the model is still valid when this

hypothesis fails.
In order to do so, another simulation in open-loop was performed, with all the same parameters of the previous one, except for target initial velocity. This results to be equal 5e-5 in all its components, both angular and linear along the three axes.

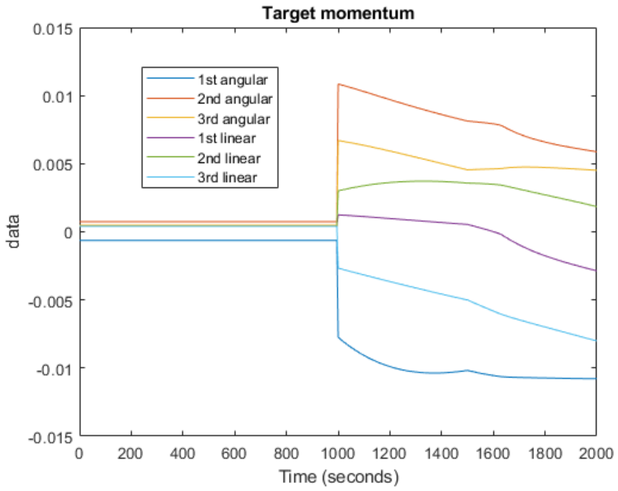

Figure 1.8: Capture validation with target linear velocites: target momenta

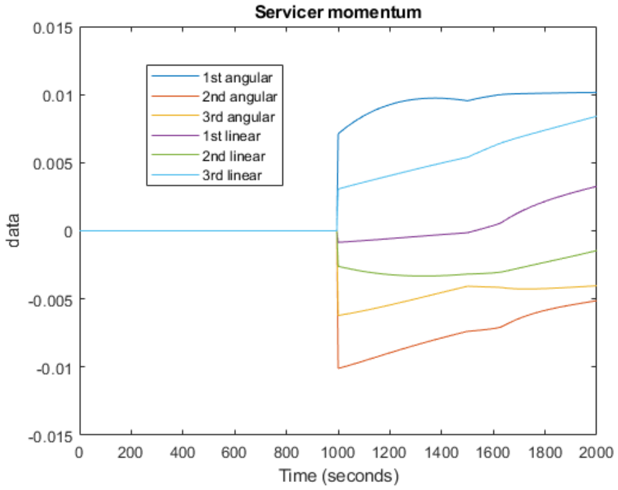

Figure 1.9: Capture validation with target linear velocites: servicer momenta

As shown in the graph below, the model is still perfectly valid, since the conservation of momentum is totally respected. Unlike the previous scenario, it can be seen how the linear momenta of the global system are now different from zero, but still constant throughout the whole simulation.

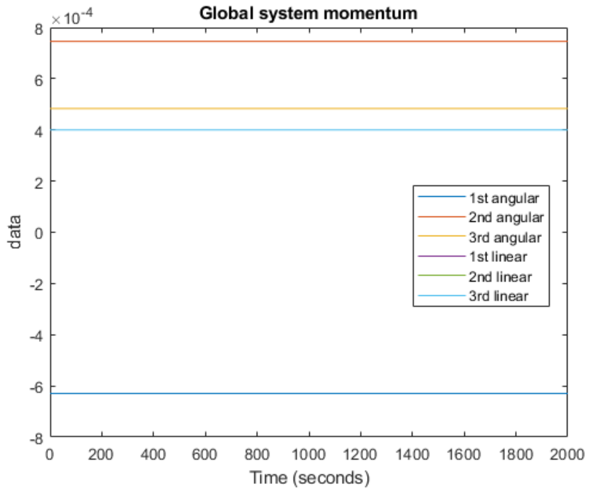

Figure 1.10: Capture validation with target linear velocites: global system momenta

1.4 Matlab Tools

All the simulations were performed in *Matlab* and *Simulink*, by exploiting some specific tools for space robotics. In particular, a Matlab open-source modeling and control toolkit for orbiting spacecraft with robotic manipulators was used. It is called SPART and it contains a lot of key functions for robot definition and dynamics evolution.

These functions support *.urdf* format for robot characteristics and specifications and allow to derive important information in several domains: kinematics, differential kinematics, dynamics and so on.

In addition, there are specific functions that help in the transition and conversion from quaternions to Euler's angles and vice-versa, so that the variables to provide in input can be expressed in the proper form.

The toolbox has been exploited basically in all the codes, from dynamics to control computation, by a careful integration with the functions created ad hoc for our purposes.

The *Simulink* codes are organized in 3 main blocks, each responsible for a specific task: reference motion, dynamics and control.

Chapter 2

NDI-NDO Control Strategy

In this chapter, a control strategy based on a combination of NDI and NDO will be presented. This will be used as benchmark to prove the efficiency of another control strategy based on NDI and MRAC, that will be introduced in Chapter 3. First of all, the new robot model used within all the simulations of the chapter will be described. After that, the control strategy will be deeply analyzed with all the related equations. Finally, some simulations will be performed to validate the model and the main results will be discussed.

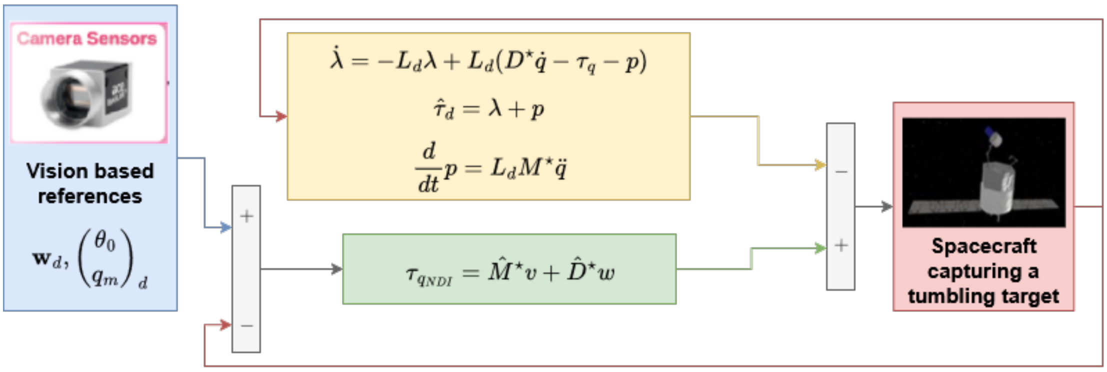
Camera Sensors
Vision based references
$\mathbf{w}_d, \begin{pmatrix}\theta_0 \\ q_m\end{pmatrix}_d$
+
−
$\dot{\lambda} = -L_d\lambda + L_d(D^\star\dot{q} - \tau_q - p)$
$\hat{\tau}_d = \lambda + p$
$\frac{d}{dt}p = L_d M^\star \ddot{q}$
$\tau_{q_{NDI}} = \hat{M}^\star v + \hat{D}^\star w$
−
+
Spacecraft capturing a tumbling target

Figure 2.1: Block diagram of the proposed control strategy

2.1 Robot model

A new servicer robot is used within the simulations of this chapter, more complex than the one described in 1.1. It is characterized by a base with a mass of 2000 kg, to which are attached 2 solar panels and a VISPA arm, that consists of 6 revolute joints and end with an end-effector. Finally, 3 reaction wheels are mounted on the base, bringing the total number of DoFs to 9. However, since the joint closest to

the EE results very hard to control, it is turned from revolut to fixed, reducing the DoFs to 8. These DoFs can be used to control the 3 angular velocities of the base (through the RWs) and the 3 linear velocities and 2 angular velocities of the EE (through the manipulator joints).
Another specific model is used to represent the target. It is characterized by a mass of 620 kg, approximately a third of the base mass. The model is created starting from the base *urdf* file, by setting to 0 all the masses and inertia of both the base and the arm and by adding a new body, that represents the target itself.
Finally, after capture, a third model is used to represent the assembly of servicer and target. The *urdf* file is built as described for the target ones, but without modifying the servicer masses and inertia.

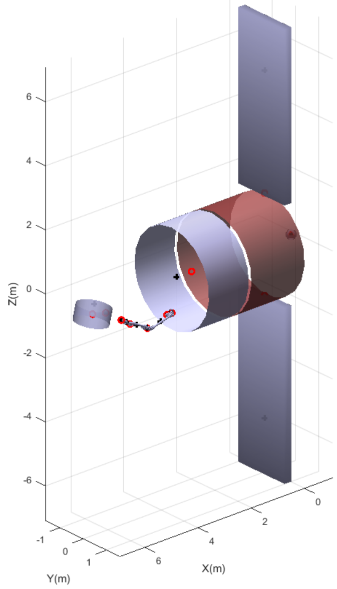

Figure 2.2: Simplified representation of the robot

2.2 NDI-NDO Control Strategy

In this section, a control strategy based on a combination of NDI (Nonlinear Dynamic Inversion) and NDO (Nonlinear Disturbance Observer) will be presented. This strategy is chosen with the goal of counteracting the strong nonlinearities that characterize the SMS dynamics.

2.2.1 NDI control

NDI is a nonlinear control technique that cancels the system nonlinearities by applying an inversion of the system dynamics. The main idea is to use the nonlinear dynamics of the system and invert it, so that the closed-loop system behaves like a linear system, which becomes easier to control.
Starting from the general equation for SMS dynamics in the joint space (1.7), in order to have a complete NDI control, its formulation should be:

$$\tau_{NDI} = H^{\diamond} K(\dot{q}_d - \dot{q}) + C^{\diamond}\dot{q} \tag{2.1}$$

where K is a diagonal matrix and q_d is derived from the desired motion. By substituting this expression in (1.7), it leads to the reference model equation:

$$\ddot{q} = K(\dot{q}_d - \dot{q}) \tag{2.2}$$

Since K is a diagonal matrix, the above system results to be perfectly decoupled.

2.3 NDO control

NDI control works well with the unperturbed system, but it is not enough when we introduce some unkown perturbations to the system, as the target capture. To compensate for this kind of perturbations, a new control technique needs to be introduced. In this case, NDO is chosen.
NDO is a method that introduces a disturbance observer to estimate and compensate for external perturbations and model uncertainties in real time. NDO provides robustness, compensating for unmodeled effects and disturbances.
With the introduction of momenta and disturbances after the capture, equation (1.7) can be modified as follows:

$$\hat{H}^{\diamond}\ddot{q} + \hat{C}^{\diamond}\dot{q} = \tau_q \tag{2.3}$$

where $\hat{[]}$ represents the perturbed version after capture of system matrices.
Each perturbed matrix can be written as the sum of the unperturbed one and a disturbance:

$$\hat{X} = X + \Delta X$$

Consequently, we can rewrite equation (2.3) by isolating in the left side the unperturbed terms in common with (1.7):

$$\begin{aligned} H^{\diamond}\ddot{q} + C^{\diamond}\dot{q} &= \tau_q - \Delta H^{\diamond}\ddot{q} - \Delta C^{\diamond}\dot{q} \\ &= \tau_q + \tau_d \end{aligned} \tag{2.4}$$

where τ_q is the NDI control and τ_d represents the disturbance torque.
Since the NDI was built starting from the unperturbed equation (1.7), the modeling uncertainties need to be rejected in order to have an effective control [13]. This can be achieved by estimating the disturbance torque ($\hat{\tau}_d$) through a Nonlinear Disturbance Observer, which is developed capitalizing on the structure of the system dynamics by introducing a gain L_d [25]:

$$\dot{\hat{\tau}}_d = -L_d\hat{\tau}_d + L_d(H^\diamond\ddot{q} + C^\diamond\dot{q} - \tau_q) \tag{2.5}$$

A new variable λ can be introduced, such that:

$$\lambda = \hat{\tau}_d - p(q, \dot{q}, \dot{q}_0) \tag{2.6}$$

where the vector p can be computed from L_d as follows:

$$\frac{d}{dt}p(q, \dot{q}, \dot{q}_0) = L_d H^\diamond \ddot{q} \tag{2.7}$$

Therefore, the NDO is totally defined thorugh the following equations:

$$\begin{cases} \dot{\lambda} = -L_d\lambda + L_d(C^\diamond\dot{q} - \tau_q - p) \\ \hat{\tau}_d = \lambda + p \\ \frac{d}{dt}p = L_d H^\diamond\ddot{q} \end{cases} \tag{2.8}$$

Once the disturbance torque is computed by exploiting this method, the adaptive control can be defined as the opposite of this quantity ($\tau_a = -\tau_d$), in order to compensate for it and allow the NDI to act on a system closer to the unperturbed one.

2.4 Model Validation

Several simulations were performed to validate the control strategy explained in the previous sections. The robot model used is the one described in Section 2.1 and 6 different simulations were carried out, each one with a different perturbation obtained by randomly varying the target initial rotational velocities in the range [5e-6 : 2e-4] rad/s (the initial linear velocities are always considered as zero).
The performance of NDI-NDO control are evaluated by computing different variables representative of the system behavior for each simulation and comparing their values. Some considerations about the simulation scenario and the most relevant results are collected in the following pages.

Simulation ID	ω_x	ω_y	ω_z
Sim 1	6.5687e-05	4.9651e-05	5.9037e-06
Sim 2	3.7302e-05	3.4714e-05	1.6437e-04
Sim 3	1.3940e-04	2.0244e-05	5.5675e-05
Sim 4	1.5089e-04	9.1322e-05	1.6101e-04
Sim 5	9.2856e-05	2.5797e-05	8.9126e-05
Sim 6	2.1345e-05	1.9257e-04	1.8258e-04

Table 2.1: Target initial velocities for each simulation

2.4.1 Base stabilization

As written previously, the proposed control strategy is built to work in the joint space. However, some goals of the task space can be achieved by computing the corresponding motion of robot DoFs that generates the desired velocity or position profile in the task space.
In this perspective, the task space objective of stabilizing the servicer by imposing zero angular velocities to its base was implemented within the proposed simulations. The strategy followed to achieve this goal by imposing a specific velocity profile to RWs will be explained in this section.
First of all, since the objective is to have a non-rotational base, its desired orientation around all the three spatial directions is supposed constant and imposed equal to 0 for simplicity. In addition, the real-time orientation of the base is derived from the system dynamics by integrating the base angular velocity starting from zero initial orientation. Doing so, the angular error between the real and the desired base orientation can be computed and a simple proportional control can be exploited to calculate the desired angular velocity profile of the base:

$$\omega_{0_{des}} = k_\omega(\theta_{0_{des}} - \theta_0) \tag{2.9}$$

At this stage, the reference velocities that the RWs need to follow to guarantee the desired base velocity profile can be computed by leveraging the equations derived from the conservation of momentum. Specifically, in the pre-capture phase, we can compute $\dot{q}_r$ from equation (1.17), in which we consider only the angular components of the momentum and we distinguish the RWs contribute from the joints one. After defining $H_{0q_r} = H_{0q}(1:3, 1:n_r)$, $H_{0q_m} = H_{0q}(1:3, n_r{+}1 : n_r{+}n_q)$ and $H_{0_0} = H_0(1:3, 1:3)$, it follows:

$$\dot{q}_{r_{ref}} = -H_{0q_r}^{-1} \cdot (H_{0q_m}\dot{q}_m + H_{0_0}\omega_{0_{des}}) \tag{2.10}$$

By analogy, in the post-capture phase, we apply the same procedure to equation (1.19):

$$\dot{q}_{r_{ref}} = H_{0q_r}^{-1} \cdot (R_0^{-1}\delta_s - H_{0q_m}\dot{q}_m - H_{0_0}\omega_{0_{des}}) \tag{2.11}$$

Therefore, the control is applied in the joint space with the goal of making the RWs follow the reference velocity profiles defined above.

2.4.2 Joints motion

While the RWs velocities are computed so that the base will stay stable, the reference motion followed by the joints has no connections with the task space, but it is directly defined in the joint space by exploiting a path-following approach.
Specifically, given the joints initial positions and the desired final ones, a smooth motion with zero initial and final velocities is imposed to each joint and the correspondent velocity profiles are used as reference for the control strategy. By doing so, similar profiles are obtained for all the joints, with the same shape but scaled differently.
The capture happens after 780 s, when all the joints velocities are very close to zero. Therefore, the main goal of the control strategy is to be able of capturing the target while keeping the joints stable and driving their velocities to zero after a short transitory.

2.4.3 Tuning and gain synthesis

Since observer and controller dynamics are coupled, it becomes crucial to simultaneously design a gain synthesis. Observer and control gains are computed through the resolution of the following LMI problem:

$$\begin{bmatrix} (\mathbf{A}_z\mathbf{Q}_z + \mathbf{B}_z\mathbf{W}_z)^{\mathbf{s}} & \mathbf{B}_{zd}\mathbf{Q}_d & \mathbf{B}_{zw} & \mathbf{Q}_z\mathbf{C}_z^T \\ * & -(\mathbf{Q}_d)^{\mathbf{s}} + \dot{H}^{\diamond} & \mathbf{0} & \mathbf{0} \\ * & * & -\gamma\mathbf{I} & \mathbf{0} \\ * & * & * & -\gamma\mathbf{I} \end{bmatrix} < 0 \tag{2.12}$$

where $\mathbf{X}^{\mathbf{s}} = \mathbf{X} + \mathbf{X}^T$, Q_z and P_d are positive definite matrices, $Q_d = P_d^{-1}$, W_z and W_d are matrices of appropriate dimensions, γ_0 is a positive scalar and the following auxiliary matrices are defined:

$$
\begin{cases}
\mathbf{A}_z = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \Delta\mathbf{C}^\diamond \end{bmatrix} & \text{(2.13a)} \\
\mathbf{B}_z = \begin{bmatrix} \mathbf{0} \\ -(\mathbf{H}^\diamond + \Delta\mathbf{H}^\diamond) \end{bmatrix} & \text{(2.13b)} \\
\mathbf{B}_{zd} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix} & \text{(2.13c)} \\
\mathbf{B}_{zw} = \begin{bmatrix} \mathbf{0} \\ -\Delta\mathbf{C}^\diamond \end{bmatrix} & \text{(2.13d)} \\
\mathbf{C}_z = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} & \text{(2.13e)}
\end{cases}
$$

The LMI problem defined in (2.12) is solved by implementing it on YALMIP, a Matlab toolbox for modeling and optimization. The outputs of the optimization process are the matrix K used inside the NDI control and the matrix $P_d = Q_d^{-1}$ exploited in the NDO control.

2.4.4 Results

As shown in Fig.2.3, the angular velocity of the servicer base is affected by a strong increase at the capture time, due to the momentum transfer between target and servicer. However, the stabilization strategy described in Section 2.4.1 allows to quickly dampen this effect, driving the base angular velocity back to 0 after a short time interval.

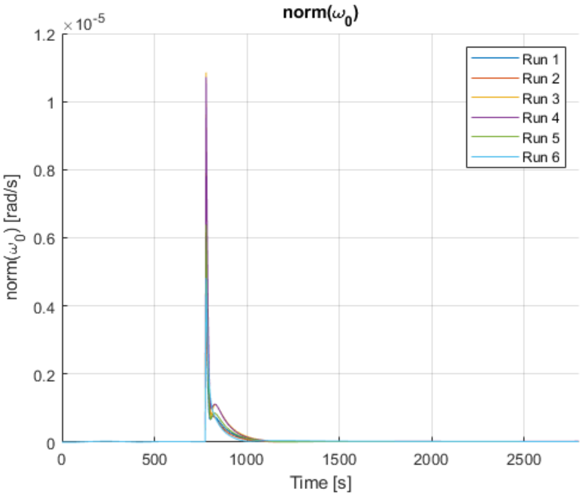

Figure 2.3: Angular velocity of the servicer base

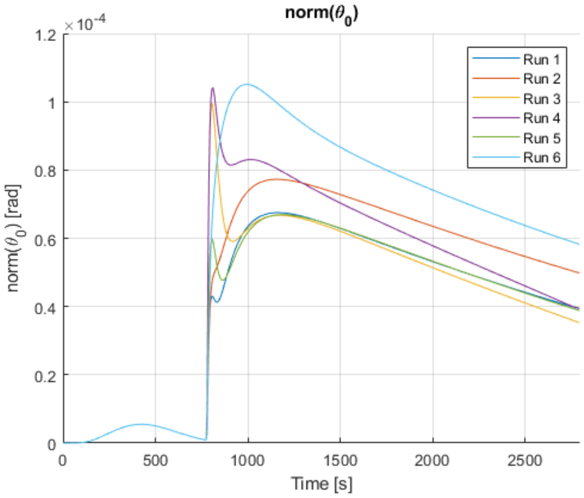

Figure 2.4: Angular displacement of the servicer base

The NDO control (Fig.2.5) becomes significant only after capture, since perturbations and disturbances are negligible before that moment, when NDI is perfectly able to control the system. Moreover, since the perturbation introduced by the target capture acts far stronger on joints than on RWs, the components of NDO control related to joints are higher than the ones correspondent to RWs.

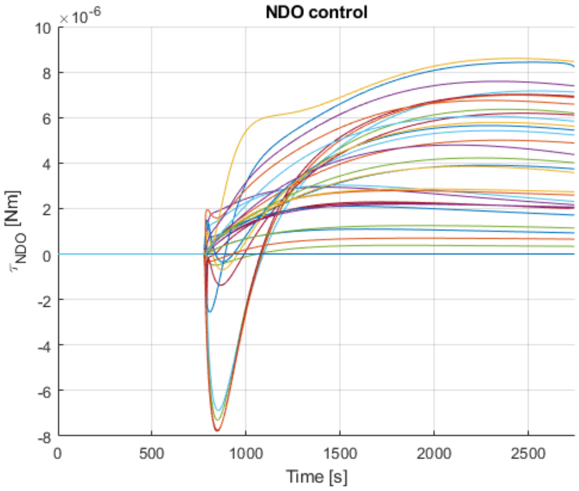

Figure 2.5: NDO control torque

As expected, all the joints velocities (Fig.2.6) stay stable and close to zero after capture. However, while the joints closest to the base quickly converge to zero, the control is not equally effective on the most distant ones. Consequently, the velocities of the fourth and fifth joints do not diverge, but at the same time their convergence is extremely slow.

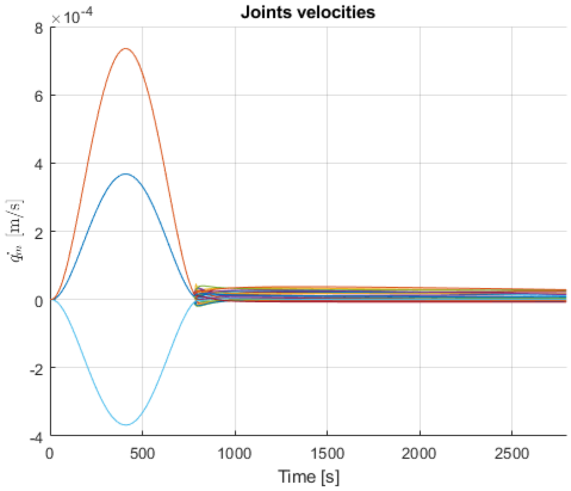

Figure 2.6: Joints velocities

On the other hand, RWs velocities (Fig.2.7) rapidly converge to the values required to stabilize the servicer base, that are strongly different as initial target conditions change.

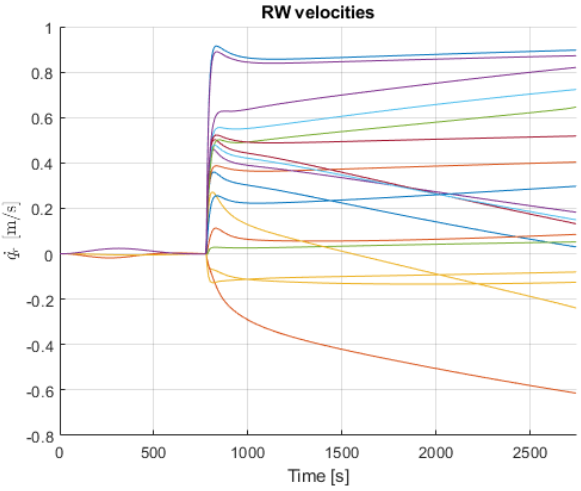

Figure 2.7: RWs velocities

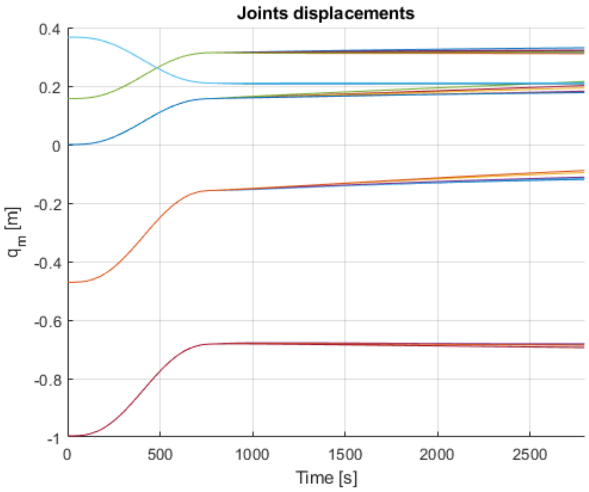

Figure 2.8: Joints displacements

Since the control strategy is developed in the joints space, the EE motion is not directly controlled. However, the stop of motion imposed to all the joints induced an analog effect on the EE (Fig.2.9), that follows a smooth reference motion similar to the joints one. Due to its proximity to the target, it is strongly influenced by the perturbations it introduces, resulting again in an imperfect and very slow convergence to zero.

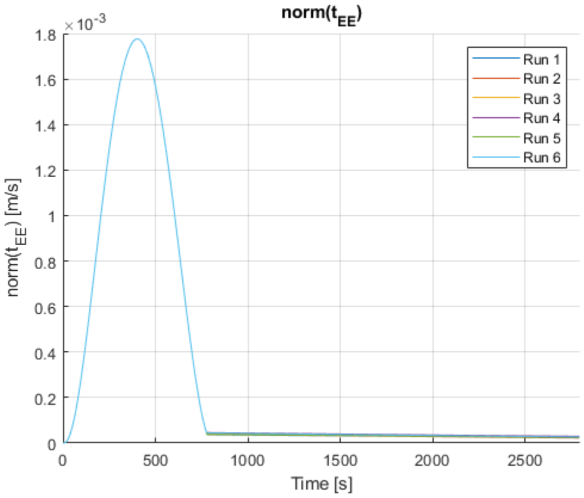

Figure 2.9: EE velocity

2.4.5 Conclusions

Overall, the combination of NDI and NDO controls show satisfactory results, allowing to properly stabilize the base and achieve the expected velocity profile for joints. However, the complexity of the robot (characterized by a high number of DoFs) and the significant mass of the target (resulting in strong perturbations) make the convergence to the desired profile slower than expected.
For this reason, in the next section a new study case will be presented, that results to be more representative of the effectiveness of the control strategy. The same study case will be utilized in the next chapter to analyze the MRAC control strategy, so that some comparisons between the two methods can be conducted.

2.5 Study case

To illustrate the efficiency of the proposed method, the following study case is presented: a rotation-free floating SMS is used to capture a tumbling target. The servicer is composed of a 100kg base, on which are mounted 3 reaction wheels and a 3-DoFs robotic manipulator (see Section 1.1). The control torques are computed with the goal of stabilizing the base (zero angular velocities) and making the robotic joints follow a desired step motion profile, that ends with zero desired velocities for all the joints. The capture happens at $t_c = 1500s$, basically at the end of the first step. The objective is to be able to compensate for the unknown perturbations introduced by the target, in order to stabilize the system after capture. The chosen MRAC formulation is the one described in Section 3.3.3, with an online tuning of parameters γ_i.
Firstly, 10 different simulations were performed, by randomly varying target angular velocities between 5e-6 and 2e-4 rad/s, with a constant target mass of 8 kg. Finally, 4 more simulations were conducted, with a constant target angular velocity of 5e-5 rad/s along all the directions and a varying target mass of 8, 24, 40 and 80 kg respectively. Since the perturbations introduced by varying the target velocities and mass resulted to have approximately the same effects on the system, only simulations showing variations in speed will be reported for simplicity.

Simulation ID	ω_x	ω_y	ω_z
Sim 1	9.0555e-05	5.8825e-05	1.5150e-04
Sim 2	7.9404e-05	1.3754e-04	5.4744e-05
Sim 3	1.5428e-04	1.3274e-04	1.0366e-04
Sim 4	1.6006e-04	3.6709e-05	1.4132e-04
Sim 5	4.1440e-05	2.8205e-05	1.7873e-04
Sim 6	1.0050e-04	1.0218e-04	1.9206e-04
Sim 7	9.1889e-05	1.9215e-04	1.1171e-04
Sim 8	1.3103e-04	7.1375e-05	3.2032e-05
Sim 9	1.4333e-04	1.1913e-04	3.4112e-05
Sim 10	1.5216e-04	4.8643e-05	5.5214e-05

Table 2.2: Target initial velocities for each simulation

2.5.1 Simulation results

One the main goal of the control strategy is to be able of stabilizing the base. That means to bring base angular velocities down to zero after the perturbation and to keep them stable around this value. As shown in Fig.2.10, this objective was perfectly achieved, since in all the simulations the base angular velocity is strongly perturbed by the capture, but immediately dampened and brought back to zero by the control torques.

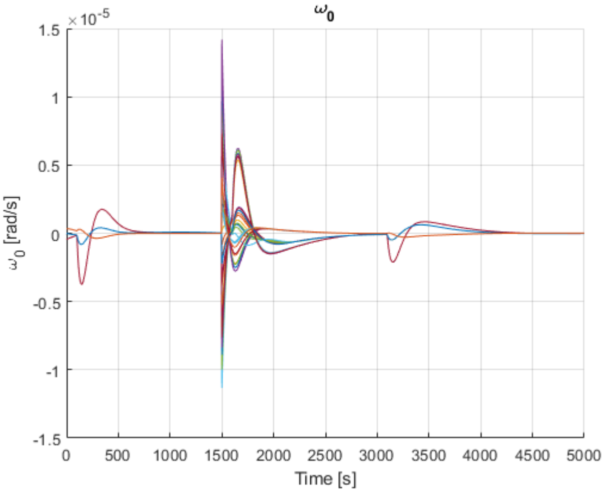

Figure 2.10: Base angular velocities for each run

In Fig.2.11 and Fig.2.12, it is clear the difference in joints motion control performance with and without using NDO control. The first figure is characterized by the utilization of only NDI: the convergence is imperfect and slow and the velocities can not reach their reference values within the time duration of the step. On the other hand, as shown in the second figure, the combination of NDI and NDO allows to fasten the convergence and improve the overall performance, so that all the joints velocities reach their desired final values by the end of simulation.

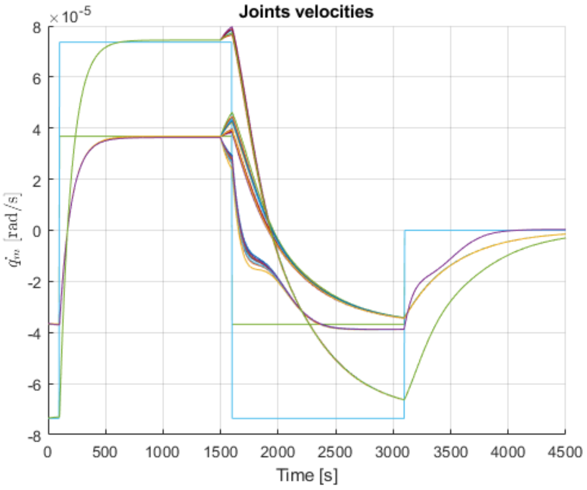

Figure 2.11: Joints velocities without NDO for each run

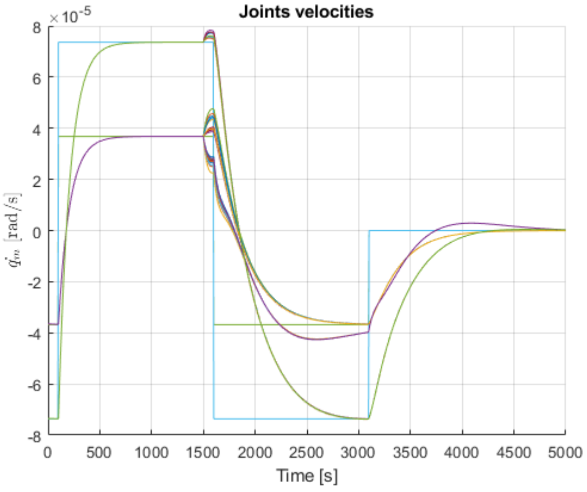

Figure 2.12: Joints velocities with NDO for each run

By comparing NDI and NDO torques in Fig.2.13 and Fig.2.14, a similar behavior can be observed, with the highest peaks right after the capture and secondary peaks corresponding to each step. Overall, NDO control torques result be approximately one order of magnitude lower than NDI ones.

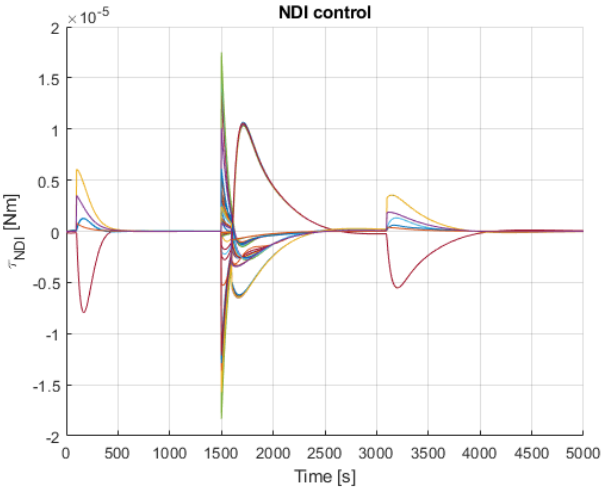

Figure 2.13: NDI control torques for each run

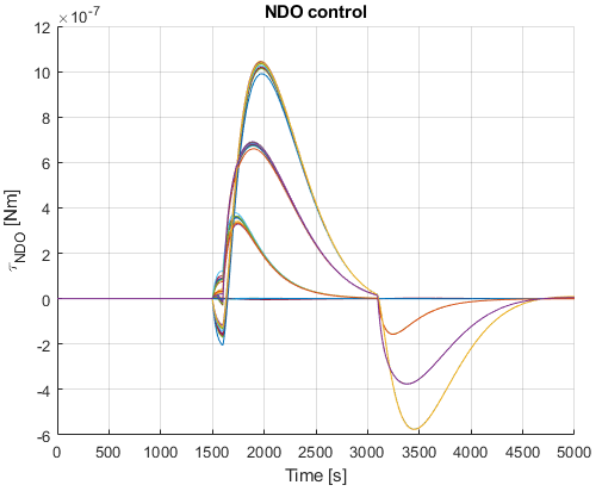

Figure 2.14: NDO control torques for each run

2.5.2 Conclusions

Despite the difficulties in adapting to the sudden, rapid and strong variations imposed by the step motion, the proposed control strategy results to act on the system effectively: the base is well stabilized and the joints velocities converge rapidly to

the reference values. The NDO control effectively estimates and counteracts the disturbance introduced by the target capture, allowing the NDI to acts on a system very close to the unperturbed one.
To quantify the improvements brought by the new method, the table Tab.2.3 compares for each simulation the mean squared errors (MSEs) in the computation of joints velocities with just NDI control and with the combination of NDI and NDO. For our purpose, the MSE can be computed as follows:

$$MSE = \frac{1}{T_{sim}} \int_0^{T_{sim}} ||\dot{q}_m - \dot{q}_{m_r}||^2 \, dt \tag{2.14}$$

where T_{sim} is the simulation time, $\dot{q}_m$ is the real joints velocities vector and $\dot{q}_{m_r}$ is the reference joints velocities vector.
As shown in the table, the addition of NDO control to NDI allows the error to be reduced by half. Overall, both the strategies are characterized by very low error values, meaning that the control is effective.

Simulation ID	**Without NDO**	**With NDO**
Sim 1	1.0034e-09	5.1862e-10
Sim 2	9.5732e-10	5.0352e-10
Sim 3	9.8275e-10	5.1492e-10
Sim 4	1.0017e-09	5.2036e-10
Sim 5	1.0157e-09	5.2079e-10
Sim 6	1.0230e-09	5.2634e-10
Sim 7	9.8285e-10	5.1330e-10
Sim 8	9.5120e-10	5.0267e-10
Sim 9	9.5172e-10	5.0389e-10
Sim 10	9.6251e-10	5.0696e-10

Table 2.3: Mean squared errors in joints velocities with NDO

In Chapter 3, the same comparison will be conducted between NDO and MRAC, in order to assess the most effective control strategy for the proposed study case.

Chapter 3

MRAC-NDI Control Strategy

In this chapter, a control strategy based on the combination of NDI and MRAC will be presented. All the fundamental equations will be deeply explained and the model will be validated through multiple simulations on *Matlab* and *Simulink*. In first place, the simulations will be performed with a simple robot and a simplified dynamics, in which the target capture is reproduced by increasing the EE mass and inertia, in order to validate the control. A comparison between a classical complete version of NDI control and a simplified one will be shown, highlighting differences and similarities. After that, new simulations will be performed with the complete model, characterized by the capture dynamics described in Chapter 1. In addition, a workspace analysis will be performed to define the boundaries within which the inertia matrix may vary.

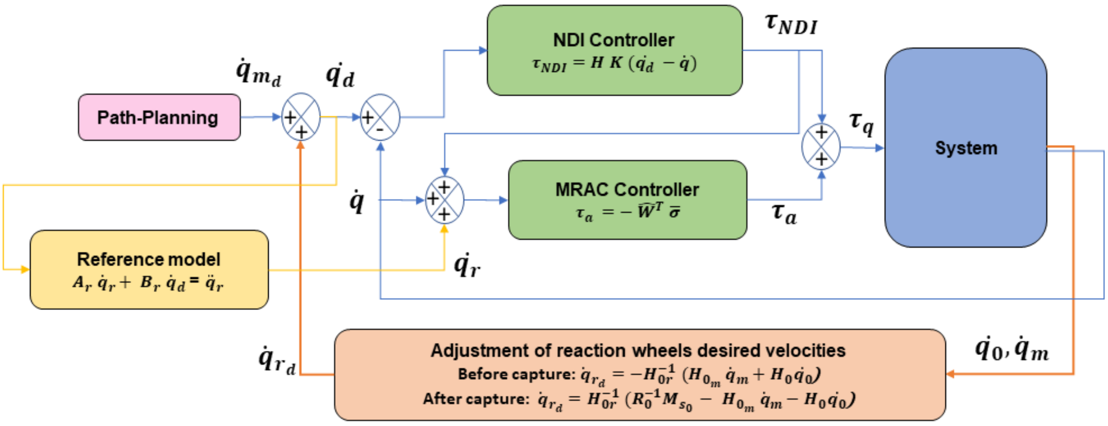
NDI Controller
$\tau_{NDI} = H\,K\,(\dot{q}_d - \dot{q})$
τ_{NDI}
$\dot{q}_{m_d}$
$\dot{q}_d$
Path-Planning
+
+
+
-
τ_q
System
$\dot{q}$
+
+
+
MRAC Controller
$\tau_a = -\widehat{W}^T\,\bar{\sigma}$
τ_a
+
+
Reference model
$A_r\,\dot{q}_r + B_r\,\dot{q}_d = \ddot{q}_r$
$\dot{q}_r$
$\dot{q}_{r_d}$
Adjustment of reaction wheels desired velocities
Before capture: $\dot{q}_{r_d} = -H_{0r}^{-1}\,(H_{0_m}\,\dot{q}_m + H_0\dot{q}_0)$
After capture: $\dot{q}_{r_d} = H_{0r}^{-1}\,(R_0^{-1}M_{s_0} - H_{0_m}\,\dot{q}_m - H_0\dot{q}_0)$
$\dot{q}_0, \dot{q}_m$

Figure 3.1: Block diagram of the proposed control strategy

The goal of this chapter is to show how efficiently MRAC can compensate for the perturbations derived by the capture of a target through the robotic arm.

In addition, it helps NDI control in linearizing the system dynamics, in order to achieve the decoupling of actuators motion.
Finally, to prove the efficiency of this control strategy, a comparison with NDI-NDO will be made, highlighting the main differences in terms of performance and results.

3.1 Control strategy with a simplified NDI

The first presented strategy of the two mentioned above is the one including a simplified version of NDI.
As shown in Section 2.2.1, the complete NDI control formulation is:

$$\tau_{NDI} = H^{\diamond} K(\dot{q}_d - \dot{q}) + C^{\diamond}\dot{q} \tag{3.1}$$

This leads to the reference model:

$$\ddot{q} = K(\dot{q}_d - \dot{q}) \tag{3.2}$$

However, from simulations it can be argued that the centrifugal term is negligible with respect to the inertial one in NDI control. As a consequence, a modification to the previous model is proposed, where the NDI is applied in the following simplified form:

$$\tau_{NDI} = H^{\diamond} K(\dot{q}_d - \dot{q}) \tag{3.3}$$

All the equations that rule the behavior of this new model will be explored in the following sections.

3.1.1 Perturbed system

Assuming of adding some unknown perturbations to the system (given for example by the capture of the target), the goal becomes to express the system dynamics in the following general form [18]:

$$\dot{x} = Ax + BW^T\sigma + B\Lambda u \tag{3.4}$$

where A and B are the unperturbed system matrices, x is the state vector, u is the control vector, Λ represents the uncertainties in control effectiveness and $\Delta = W^T\sigma$ represents the non-linear system uncertainties, modeled as the product between an unknown weight matrix W and a known basis functions vector σ, also called regressor.
Starting from the unperturbed system in equation (1.7), we consider a perturbation in both $H^{\diamond}$ and $C^{\diamond}$ matrices:

$$\hat{H}^{\diamond}\ddot{q} + \hat{C}^{\diamond}\dot{q} = \tau_q \tag{3.5}$$

The perturbation in inertia matrix can be expressed as $\hat{H}^{\diamond} = \alpha H^{\diamond}$. Isolating $\ddot{q}$ and adding a null term:

$$\ddot{q} = 0_{n_q \times n_q}\dot{q} - H^{\diamond -1}\alpha^{-1}\hat{C}^{\diamond}\dot{q} + H^{\diamond -1}\alpha^{-1}\tau_q \tag{3.6}$$

The system is now written in the general reference form: x $= \dot{q}$, A $= 0_{n_q \times n_q}$, B $= H^{\diamond -1}$, $W^T = -\alpha^{-1}\hat{C}^{\diamond}$, $\sigma = \dot{q}$, $\Lambda = \alpha^{-1}$ and $u = \tau_q$.

3.1.2 Reference model and nominal control

To obtain the nominal behavior of the new model, i.e. the behavior in absence of perturbations, we need to impose $\Delta = W^T\sigma = 0$, $\Lambda = \mathbf{I}$ and $\tau_q = \tau_{NDI}$. It follows that:

$$\ddot{q} = 0_{n_q \times n_q}\dot{q} + H^{\diamond -1}\tau_{NDI} \tag{3.7}$$

This corresponds to the unperturbed system equation: $\dot{x} = Ax + Bu$. Matrix A results to be null as a consequence of neglecting the centrifugal term in NDI control. Basically, we evaluate the cetrifugal-Coriolis term as an additional perturbation to compensate for through MRAC. At this point, a NDI control able to decouple the system dynamics must be designed:

$$\tau_{NDI} = H^{\diamond}(K_2\dot{q}_d - K_1\dot{q}) = -\hat{K}_1\dot{q} + \hat{K}_2\dot{q}_d \tag{3.8}$$

where $\hat{K}_i = H^{\diamond}K_i$.
By substituting this expression in (3.7), the reference model dynamics is obtained:

$$\ddot{q}_r = (0_{n_q \times n_q} - H^{\diamond -1}\hat{K}_1)\dot{q}_r + H^{\diamond -1}\hat{K}_2\dot{q}_d = A_r\dot{q}_r + B_r\dot{q}_d \tag{3.9}$$

where the reference model matrices are $A_r = A - B\hat{K}_1 = -H^{\diamond -1}\hat{K}_1 = -K_1$ and $B_r = B\hat{K}_2 = H^{\diamond -1}\hat{K}_2 = K_2$. Since these matrices are diagonal, the reference dynamics results to be decoupled.
We can now define the matrix E_p as a selector matrix of the components of the state vector we want to control: $z = E_p x$, where z is the vector containing the controlled state components. Since in our case we want to control all the velocities, $z = x$ and then $E_p = \mathbf{I}$. A_r and B_r need to satisfy the equation:

$$-E_p {A_r}^{-1} B_r = \mathbf{I} \rightarrow B_r = -A_r \rightarrow K_2 = K_1 = K \tag{3.10}$$

In conclusion, the reference model dynamics and the NDI control result to be:

$$\ddot{q}_r = -K(\dot{q}_r - \dot{q}_d) = A_r\epsilon_r \tag{3.11}$$

$$\tau_{NDI} = H^{\diamond}K(\dot{q}_d - \dot{q}) = \hat{K}(\dot{q}_d - \dot{q}) \tag{3.12}$$

where $\epsilon_r = \dot{q}_r - \dot{q}_d$ represents the error between reference and desired motions.

3.1.3 Perturbed dynamics

We can manipulate equation (3.6) by adding and subtracting two couples of equal terms:

$$\ddot{q} = 0_{n_q \times n_q}\dot{q} - H^{\diamond^{-1}}\alpha^{-1}\hat{C}^{\diamond}\dot{q} + H^{\diamond^{-1}}\alpha^{-1}\tau_q + H^{\diamond^{-1}}(\hat{K}\dot{q}_d - \hat{K}\dot{q}_d + \hat{K}\dot{q} - \hat{K}\dot{q})$$

$$\ddot{q} = (0_{n_q \times n_q} - H^{\diamond^{-1}}\hat{K})\dot{q} + H^{\diamond^{-1}}\hat{K}\dot{q}_d + H^{\diamond^{-1}}\alpha^{-1}\tau_q + H^{\diamond^{-1}}[(-\alpha^{-1}\hat{C}^{\diamond})\dot{q} + \hat{K}\dot{q} - \hat{K}\dot{q}_d]$$

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond^{-1}}\alpha^{-1}[\tau_q - \hat{C}^{\diamond}\dot{q} + \alpha\hat{K}\dot{q} - \alpha\hat{K}\dot{q}_d]$$

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond^{-1}}\alpha^{-1}[\tau_{NDI} + \tau_a - \hat{C}^{\diamond}\dot{q} - \alpha\tau_{NDI}]$$

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond^{-1}}\alpha^{-1}[(\mathbf{I} - \alpha)\tau_{NDI} + \tau_a - \hat{C}^{\diamond}\dot{q}] \tag{3.13}$$

We managed to get an equation in the form [18]:

$$\dot{x} = A_r x + B_r x_d + B\Lambda[(\mathbf{I} - \Lambda^{-1})u_n + u_a + \Lambda^{-1}W^T\sigma]$$

where u_n represents the nominal control (NDI) and u_a the adaptive one (MRAC). Defining $\bar{\sigma} = [\sigma; u_n] = [\dot{q}; \tau_{NDI}]$ and $\bar{W}^T = [\Lambda^{-1}W^T,\ \ \mathbf{I} - \Lambda^{-1}] = [-\hat{C}^{\diamond},\ \ \mathbf{I} - \alpha]$:

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond^{-1}}\alpha^{-1}[\bar{W}^T\bar{\sigma} + \tau_a] \tag{3.14}$$

That corresponds to the general form:

$$\dot{x} = A_r x + B_r x_d + B\Lambda[\bar{W}^T\bar{\sigma} + u_a]$$

Since K matrix is diagonal, the first part of equation (3.14) can be easily decoupled for the different components of q:

$$\ddot{q}_i = -K_i(\dot{q}_i - \dot{q}_{d_i}) + H^{\diamond^{-1}}(i,:)\ \alpha^{-1}[\bar{W}^T\bar{\sigma} + \tau_a] \tag{3.15}$$

3.1.4 MRAC

To compensate for perturbations, we need to add an adaptive component to our control strategy (τ_a), and we design that through MRAC.
MRAC is an adaptive control strategy where the system is required to follow the behavior of a predefined reference model that specifies the desired dynamics. While a reference model describes the ideal closed-loop response, an adaptive law continuously updates the controller parameters so that the system output tracks the reference model output, even in the presence of uncertainties or parameter variations. By exploiting the stability theory (e.g. Lyapunov-equation), it can be ensured that the closed-loop system remains stable while the controller parameters are updating.

Looking at equation (3.14), the ideal solution to cancel the disturbances would be $\tau_a = -\bar{W}^T\bar{\sigma}$. It would lead to a system behavior exactly equal to the nominal one, with a perfectly decoupled dynamics: $\dot{x} = -K(x - x_d)$. However, since the weight matrix $\bar{W}$ is unknown, that is not possible. As a consequence, we model our adaptive control as $\tau_a = -\hat{W}^T\bar{\sigma}$, where $\hat{W}$ is an estimation of $\bar{W}$ obtained through a parameter adjustment mechanism, and we define $\tilde{W} = \hat{W} - \bar{W}$. Equation (3.14) becomes then:

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond -1}\alpha^{-1}[\bar{W}^T\bar{\sigma} - \hat{W}^T\bar{\sigma}] = -K\dot{q} + K\dot{q}_d - H^{\diamond -1}\alpha^{-1}\tilde{W}^T\bar{\sigma} \tag{3.16}$$

Error Computation

$$\begin{aligned} e &= x - x_r \\ \dot{e} &= \dot{x} - \dot{x}_r \\ \dot{e} &= A_r x + B_r x_d - B\Lambda\tilde{W}^T\bar{\sigma} - A_r x_r - B_r x_d \\ \dot{e} &= A_r e - B\Lambda\tilde{W}^T\bar{\sigma} = -Ke - H^{\diamond -1}\alpha^{-1}\tilde{W}^T[\dot{q};\tau_{NDI}] \end{aligned} \tag{3.17}$$

By leveraging this equation and through some considerations on Lyapunov's equation, it can be stated that $e \to 0$ asymptotically if the update law for $\hat{W}$ that will be computed in the next section is satisfied.

Weight matrix update law

MRAC aims to estimate $\hat{W}$ in a way such that $\tilde{W}$ will be as near to zero as possible. In order to do so, we can leverage Lyapunov's function [18]:

$$V = e^T P e + \gamma^{-1} tr(\tilde{W}\Lambda^{\frac{1}{2}})^T(\tilde{W}\Lambda^{\frac{1}{2}}) \tag{3.18}$$

$$\begin{aligned} \dot{V} &= 2e^T P(A_r e - B\Lambda\tilde{W}\bar{\sigma}) + 2\gamma^{-1} tr(\Lambda\tilde{W}^T\dot{\hat{W}}) \\ \dot{V} &= 2e^T P A_r e - 2e^T P B\Lambda\tilde{W}\bar{\sigma} + 2\gamma^{-1} tr(\Lambda\tilde{W}^T\dot{\hat{W}}) \\ \dot{V} &= -||e||_2{}^2 - 2tr(\Lambda\tilde{W}^T\bar{\sigma}e^T PB) + 2\gamma^{-1} tr(\Lambda\tilde{W}^T\dot{\hat{W}}) \\ \dot{V} &= -||e||_2{}^2 - 2\gamma^{-1} tr(\Lambda\tilde{W}^T[\gamma\bar{\sigma}e^T PB - \dot{\hat{W}}]) \end{aligned}$$

If

$$\dot{\hat{W}} = \gamma\bar{\sigma}e^T PB \tag{3.19}$$

then

$$\dot{V} = -||e||_2{}^2 \leq 0 \to \text{Lyapunov's stability}$$

We have obtained the updating law for the weight matrix, where γ is an adaptive adjustment gain that needs to be determined and P is the Lyapunov matrix, obtained by solving the Lyapunov equation:

$$A_r{}^T P + P A_r + R = 0$$

where we can assume $R = \mathbf{I}$.
It was decided to create n_q different canals for MRAC, one for each component of τ_a, in order to better control the system by using different gains for each of its DoF. As a result, the update law for each canal becomes:

$$\dot{\hat{W}}_i = \gamma_i \bar{\sigma} e^T P B(:, i) \tag{3.20}$$

in which we take into account only the i-th column of B matrix.
Finally, since the manipulator joints motion results to be poorly influenced by reaction wheels behavior, it was decided to substitute $\bar{\sigma}$ with $\bar{\sigma}_m = [\dot{q}(n_r + 1 : n_q); \tau_{NDI}]$ in the last n_m channels of MRAC. Basically, we are using just the components of state vector related to manipulator joints to determine the components of τ_a that should closely control the manipulator motion. Since MRAC is not decoupled, the effectiveness of this solution will be investigated.

3.1.5 Model validation

To validate the model, the above control strategy has been applied to different types of motion of the given robot. The system dynamics was perturbed at a certain time during simulation by changing the robot model with another one characterized by an increase in end-effector mass and inertia by a factor of 5.

Step motion of last manipulator joint (Motion 1)

The first motion investigated is a step motion between positive and negative values of equal amplitude for the last manipulator joint (the closest one to EE), while the other joints stay stable at null velocity. The perturbation occurs during the first step.
By imposing $\tau_a = -\bar{W}^T \bar{\sigma}$, we find that only the 5th and 6th components of τ_a need to have significant values in order to compensate for the disturbances. As a consequence, we set $\gamma_{1,2,3,4}$ at very low values and we iterate on the values of just γ_5 and γ_6 until we get satisfying results. Although MRAC is not decoupled, a strong correspondence between control components and joints motions have been observed. From the following graphs, it can be argued that keeping $\gamma_5 = \textit{constant}$ and increasing γ_6 the errors relative to the motion of the first and second joints stay constant, while the error on the third joint strongly decreases. This results is

particularly relevant since in a perfectly decoupled model the adjustment parameter γ_6 is exactly the one that would control the motion of the third joint. The same considerations hold when $\gamma_6 = constant$ and γ_5 is increased. In this case, a small increase in the error on Joint 3 motion is observed, but it is far smaller than the reduction of the error on Joint 2.

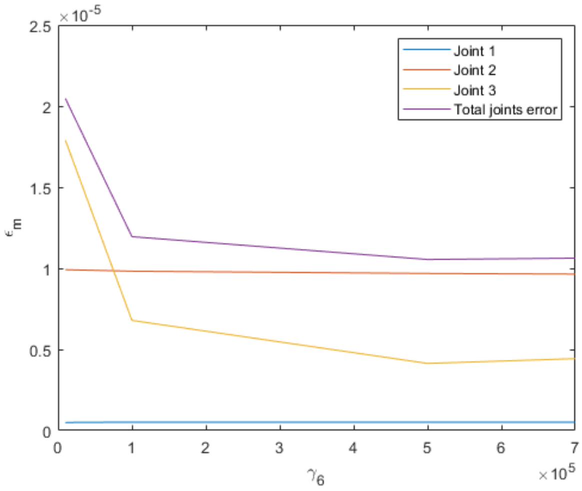

Figure 3.2: Errors as a function of γ_6

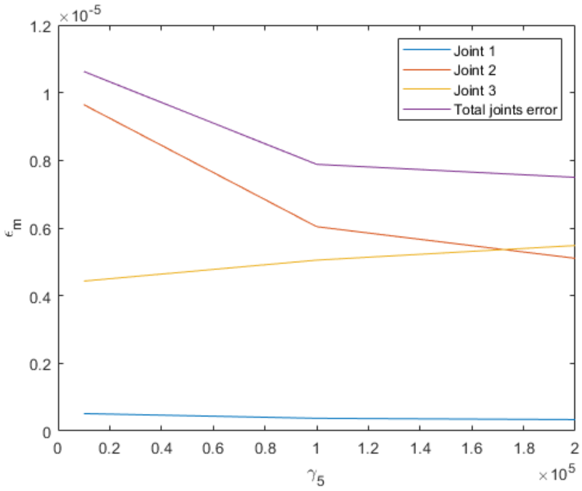

Figure 3.3: Errors as a function of γ_5

The best results have been obtained for $\gamma_5 = 2e5$ and $\gamma_6 = 7e5$, that lead to a strong reduction in the total root mean square error related to the manipulator joints motion from a value of 3.7652e-05 without MRAC to a value of 7.4973e-06 with MRAC. Here are reported the joints motion graphs with and without MRAC.

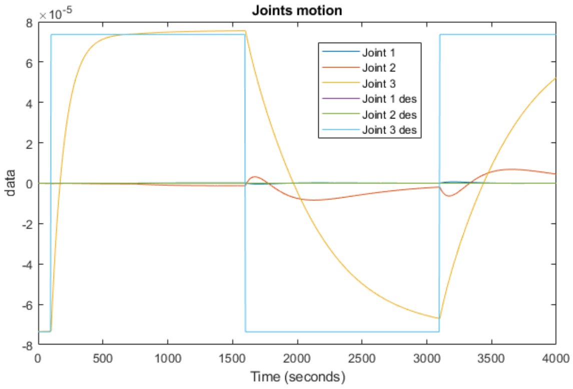

Figure 3.4: Motion1: Joints motion without MRAC

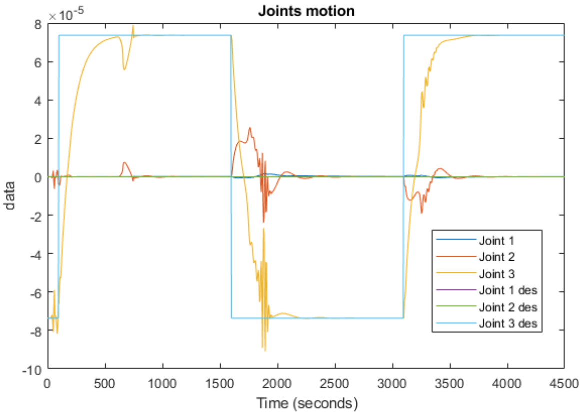

Figure 3.5: Motion1: Joints motion with MRAC

Step motion of all the joints (Motion 2)

The second motion investigated is again a step motion, but involving this time all the manipulator joints. The perturbation occurs again during the first step.
Since all the joints are involved in motion, even the parameter γ_4 is subjected to an iterative refinement process, with considerations analogue to the previous case: by increasing γ_4, the error on the correspondent joint (Joint 1) strongly decreases, while the errors on the other joints remain quite constant.

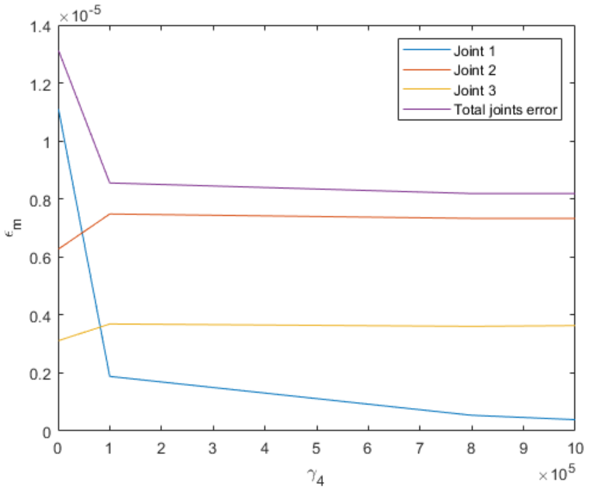

Figure 3.6: Errors as a function of γ_4

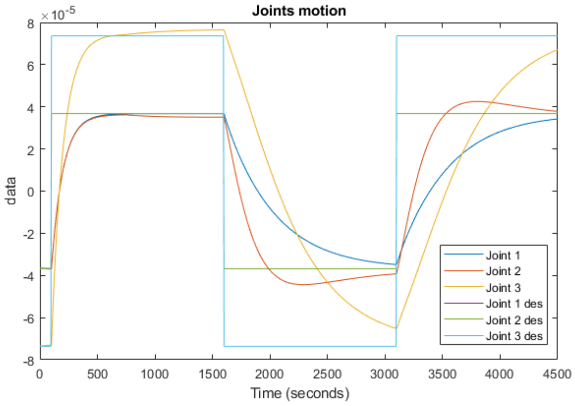

Figure 3.7: Motion2: Joints motion without MRAC

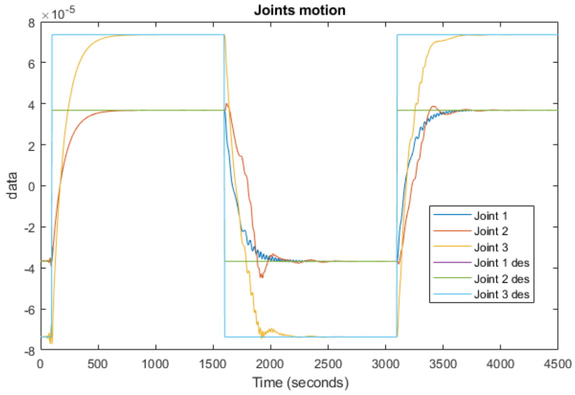

Figure 3.8: Motion2: Joints motion with MRAC

The best results have been obtained for $\gamma_4 = 8e5$, $\gamma_5 = 2e5$ and $\gamma_6 = 4e5$, that lead to a strong reduction in the total root mean square error related to the manipulator joints motion from a value of 4.5853e-05 without MRAC to a value of 8.1884e-06 with MRAC.

Step motion of all the joints followed by motion stop (Motion 3)

The third motion used for validation is analogue to the previous one, but with the addiction of final desired zero velocities, in order to evaluate if MRAC control is able to properly stop the system motion even in presence of perturbations, that occur again during the first step.
By using the same $\gamma_{4,5,6}$ parameters of the previous case, it can be observed a strong reduction in total error, from 3.6885e-05 without MRAC to 7.4302e-06 with MRAC.

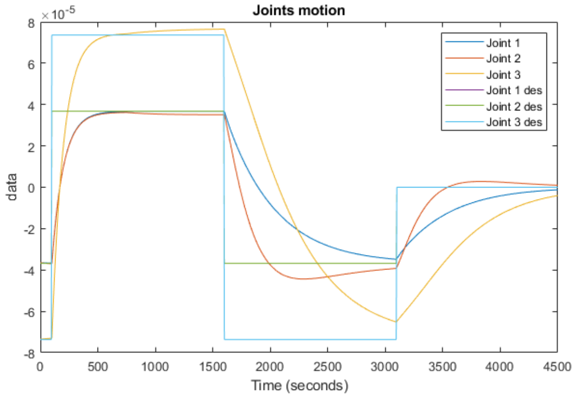

Figure 3.9: Motion3: Joints motion without MRAC

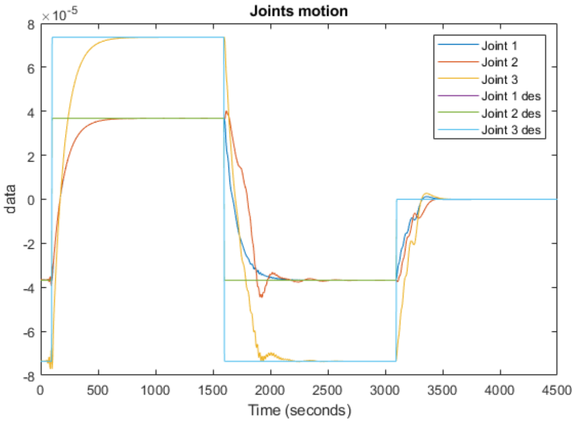

Figure 3.10: Motion3: Joints motion with MRAC

Looking at the errors, it can be seen how MRAC is able to progressively reduce their amplitude, reaching a final value of about 0 for all the joints velocities, which means a good adaptation to the reference model.

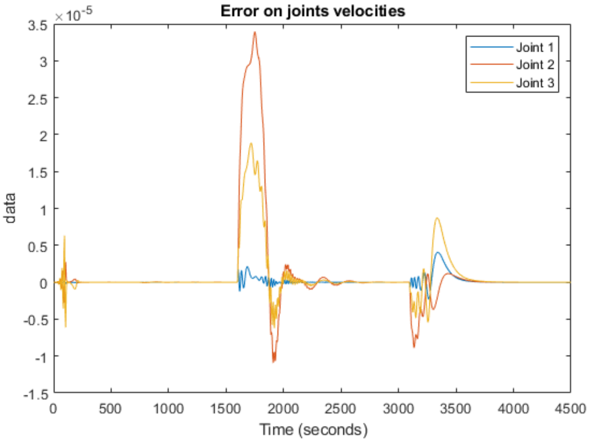

Figure 3.11: Errors on joints motion

Finally, the magnitudes of NDI and MRAC controls can be compared, with the latter that results to be an order of magnitude lower than the first one (e-6 against e-5). The MRAC peaks correspond to the steps of velocity signals and the first peak occurs when the system is not perturbed yet, to compensate for the innacuracies of NDI derived from the mathematical model.

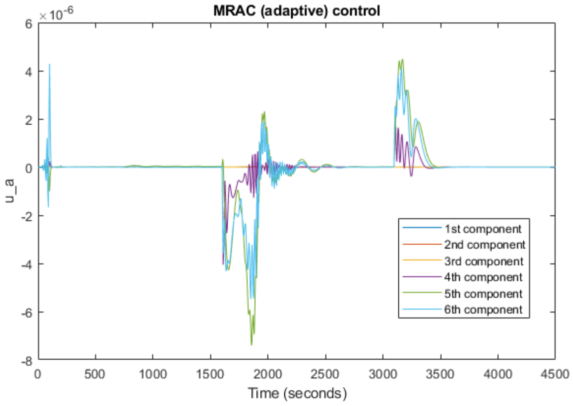

Figure 3.12: MRAC control torques

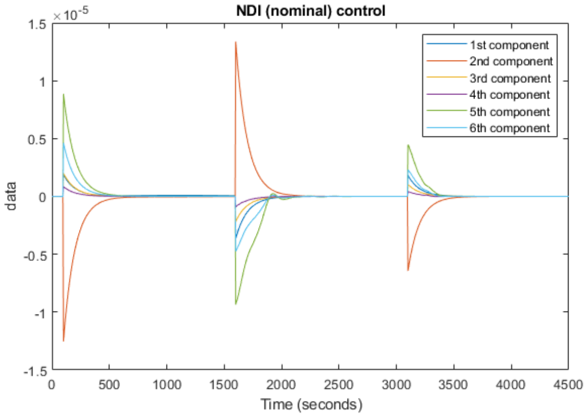

Figure 3.13: NDI control torques

Step motion of all the joints followed by motion stop and perturbations (Motion 4)

This motion is basically the same of the previous one, the only difference is the moment when the perturbation occurs: not during the first step anymore, but at the same moment of the signal of stopping motion. In this way, we can evaluate if MRAC is able to compensate for NDI inaccuracies when there are no perturbations and if it is able to stop the motion immediately after the perturbations occur.
The best results have been obtained for $\gamma_4 = 1e6$, $\gamma_5 = 1e4$ and $\gamma_6 = 3e5$, that lead to a strong reduction in the total root mean square error related to the manipulator joints motion from a value of 1.5775e-05 without MRAC to a value of 5.8039e-06 with MRAC.

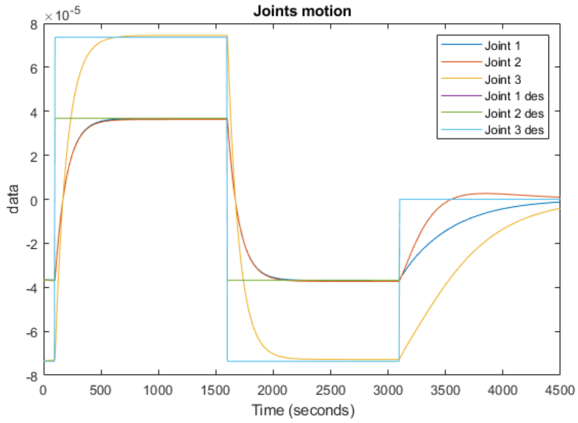

Figure 3.14: Motion4: Joints motion without MRAC

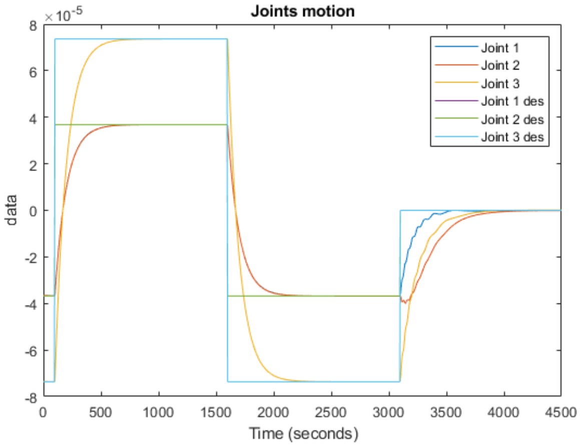

Figure 3.15: Motion4: Joints motion with MRAC

Smooth motion (Motion 5)

Finally, the last motion used to validate the model is a smooth motion where the velocities of all the joints increase, reach a peak and then go to 0. The perturbation occurs during the increase.
This motion results to be the one in which MRAC is most effective: by using the parameters $\gamma_4 = 1e5$, $\gamma_5 = 1e5$ and $\gamma_6 = 1e4$, it can be observed a reduction in error from 4.9259e-05 without MRAC to 9.0777e-07 with MRAC.

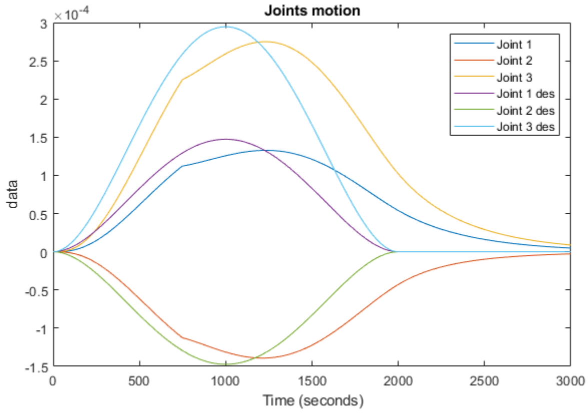

Figure 3.16: Motion5: Joints motion without MRAC

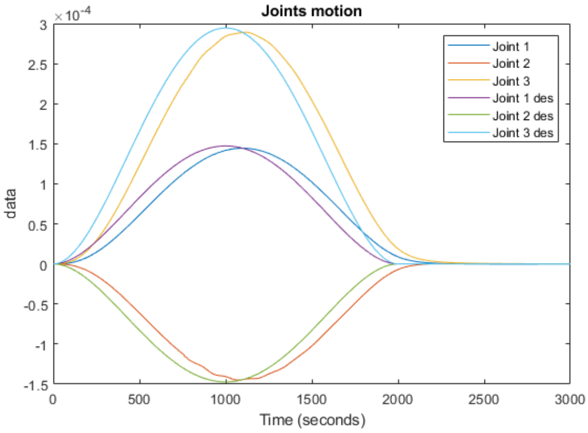

Figure 3.17: Motion5: Joints motion with MRAC

Smooth motion with different σ

As written previously, in the last n_m channels of MRAC $\bar{\sigma}_m = [\dot{q}(n_r + 1 : n_q); \tau_{NDI}]$ was used instead of $\bar{\sigma}$. To evaluate the effectiveness of this solution, a MRAC strategy that utilizes $\bar{\sigma}$ in all its channels have been applied to the smooth motion test case and its results will be compared to the ones reported in the previous sub-section.
The best result was obtained for much lower values of $\gamma_{4,5,6}$: $\gamma_4 = 1e3$, $\gamma_5 = 1e2$ and $\gamma_6 = 1e2$. The total error is strongly reduced from 9.0777e-07 to 5.8623e-07, resulting in a better adaptation to the reference model.

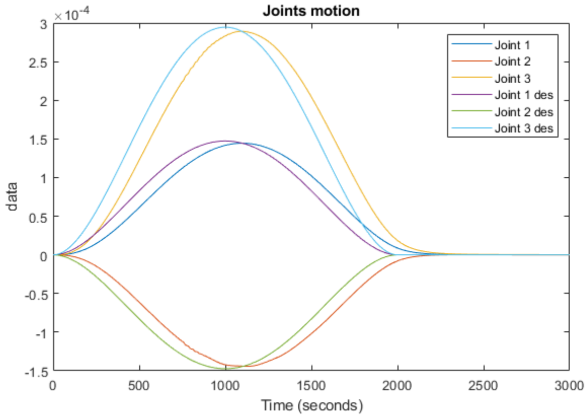

Figure 3.18: Joints motion with $\bar{\sigma}$ instead of $\bar{\sigma}_m$

However, the decoupling effect observed with $\bar{\sigma}_m$ is weakened: for example, by acting on γ_6 (and so on τ_{a_6}) we do not modify just the error on the 3rd joint motion, but also the error related to the 2nd one, resulting in a worse decoupling action and a harder fine-tuning.

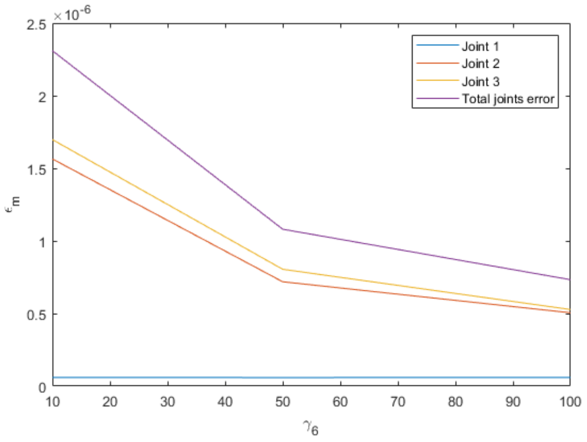

Figure 3.19: Errors as a function of γ_6

Step motion with different σ

By applying the same procedure to some types of step motion previously investigated, the following results were obtained, summarized in a table:

Type of motion	γ_4	γ_5	γ_6	Old error	New error
Motion 2	3e2	2e2	1e2	8.1884e-06	1.7113e-05
Motion 3	3e2	2e2	1e2	7.4302e-06	1.2223e-05

Table 3.1: Step motions errors with complete σ

Overall, using the complete $\bar{\sigma}$ instead of $\bar{\sigma}_m$ appears to improve the results for the smooth motion, but it worsens the ones for step motions. In addition, it weakens the decoupling effect between the components of adaptive control and joints motions previously observed. Finally, the optimal adjustment gain parameters γ_i result to be strongly reduced.

3.1.6 Modeling of momentum change

The equation (1.7) derives from considering a zero system momentum δ in the complete equation:

$$H^{\diamond}\ddot{q} + C^{\diamond}\dot{q} + C^{\diamond}_{\delta}\delta = \tau_q \tag{3.21}$$

This assumption is perfectly valid before capture, when we can consider the system momentum as constant and equal to 0, since no external forces or torques are acting. However, it is no longer true after applying the perturbation, seen as the capture of an unknown target. As a consequence, the mathematical model of the perturbed system (equation 3.5) needs to be modified:

$$\hat{H}^{\diamond}\ddot{q} + \hat{C}^{\diamond}\dot{q} + \hat{C}^{\diamond}_{\delta}\delta = \tau_q \tag{3.22}$$

The system momentum after capture can be expressed as $\delta = \hat{H}_0\dot{q}_0 + \hat{H}_{0q}\dot{q}$. This is equal to the target momentum before capture δ_t. It follows that:

$$\ddot{q} = 0_{n_q \times n_q}\dot{q} - H^{\diamond -1}\alpha^{-1}[(\hat{C}^{\diamond} + \hat{C}^{\diamond}_{\delta}\hat{H}_{0q})\dot{q} + \hat{C}^{\diamond}_{\delta}\hat{H}_0\dot{q}_0] + H^{\diamond -1}\alpha^{-1}\tau_q \tag{3.23}$$

The momentum that appears in consequence of the capture is considered as an additional perturbation to compensate for through the MRAC. By comparing this equation with (3.6), we can see that only W^T and σ change: $W^T = [-\alpha^{-1}(\hat{C}^{\diamond}_{\delta}\hat{H}_0), -\alpha^{-1}(\hat{C}^{\diamond} + \hat{C}^{\diamond}_{\delta}\hat{H}_{0q})]$, $\sigma = [\dot{q}_0; \dot{q}]$.
Equation (3.26) becomes then:

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond -1}\alpha^{-1}[(\mathbf{I} - \alpha)\tau_{NDI} + \tau_a - \hat{C}^{\diamond}_{\delta}\hat{H}_0\dot{q}_0 - (\hat{C}^{\diamond} + \hat{C}^{\diamond}_{\delta}\hat{H}_{0q})\dot{q}] \tag{3.24}$$

By defining $\bar{W}^T = [-\hat{C}^{\diamond}_{\delta}\hat{H}_0, -\hat{C}^{\diamond} - \hat{C}^{\diamond}_{\delta}\hat{H}_{0q}, \mathbf{I} - \alpha]$ and $\bar{\sigma} = [\dot{q}_0; \dot{q}; \tau_{NDI}]$, we achieve to express the system in the same form of equation (3.14). Practically, to compensate for the momentum change through MRAC, we just need to change the $\bar{\sigma}$ vector within the model, to take into account also the base velocity vector $\dot{q}_0$.

By applying the new model to the smooth motion with the complete $\bar{\sigma}$ and using the same values of $\gamma_{4,5,6}$ used for the old model, we obtain a total error on joints motion of 5.7949e-07, that is slightly lower than the value of 5.8623e-07 registered with the previous model.

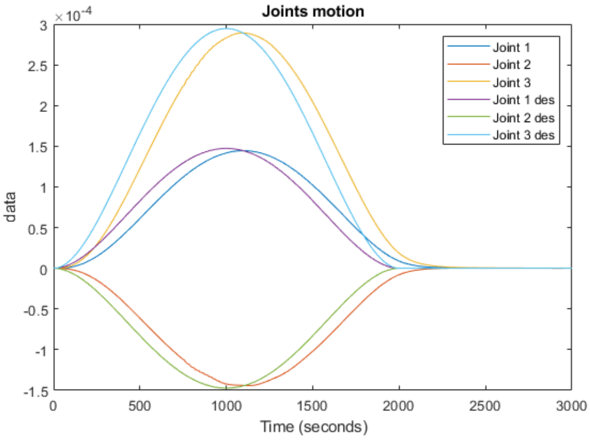

Figure 3.20: Joints motion modeling the momentum change

By applying the same procedure to two different types of step motion with $\bar{\sigma}_m$ previously investigated (again with the same values of γ_i), the following results were obtained, summarized in a table:

Type of motion	γ_4	γ_5	γ_6	Old error	New error
Motion 2	1e6	2e5	4e5	8.1884e-06	8.1679e-06
Motion 3	1e6	2e5	4e5	7.4302e-06	6.8938e-06

Table 3.2: Step motions errors with momentum change adjustment

It can be stated that modeling the system momentum change slightly improves the performance, especially in motion 3, where it reduces the oscillations in the adaptation to the last velocity step (compare Figure 3.21 with Figure 3.10).

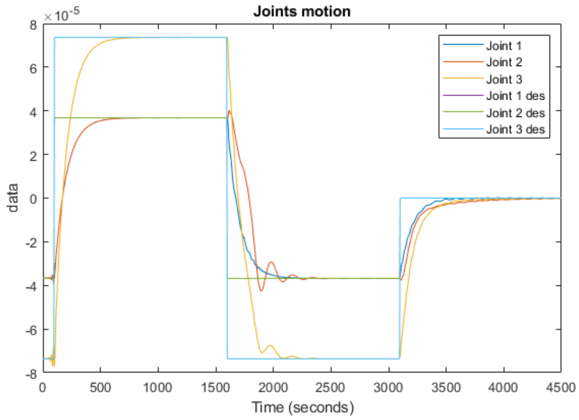

Figure 3.21: Momentum change modeling in Motion 3

In order to emphasize the effects of momentum change, the comparison regarding Motion 3 has been repeated considering bigger velocity steps (5x). Again, the total error decreases if we model the system momentum change (from 3.4269e-05 to 3.1332e-05), but the percentage decrease is not much bigger with respect to the case with minor momentum change (from 7.22% to 8.57%). That probably means that in these simulations the perturbation introduced by momentum change is quite negligible with respect to the others. Certainly, different results may be obtained by changing significantly the target mass, inertia and/or motion. The following graphs compare the two models (with and without modeling the momentum change) by overlapping each component of joints motion.

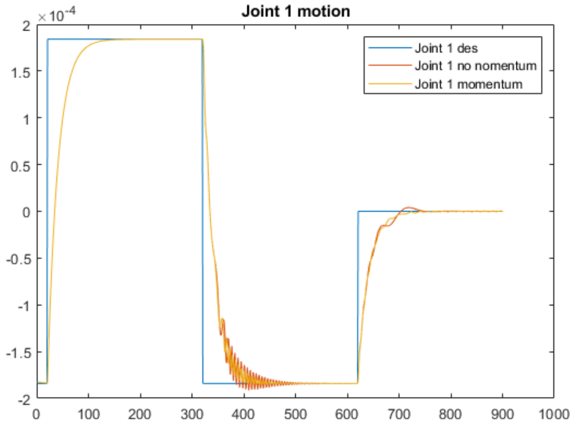

Figure 3.22: Comparison about momentum modeling: Joint 1

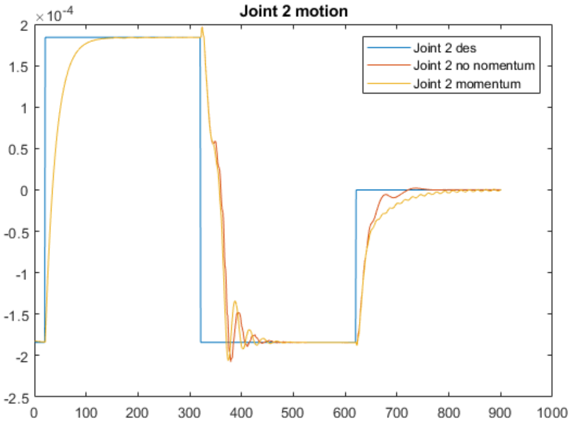

Figure 3.23: Comparison about momentum modeling: Joint 2

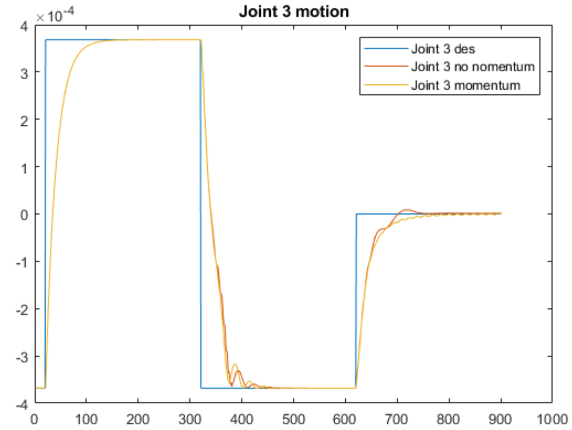

Figure 3.24: Comparison about momentum modeling: Joint 3

For each joint, the modeling of momentum change reduces oscillations and slightly improves overall performance.

3.2 Control strategy with a complete NDI

The second proposed version is the one with a complete NDI, that includes also a centrifugal/Coriolis term, which was instead compensated by the MRAC in the previous version.

3.2.1 Perturbed system

Assuming of adding some unknown perturbations to the system (given again by the capture of the target), the goal becomes to express the system dynamics in the following general form:

$$\dot{x} = Ax + BW^T\sigma + B\Lambda u \tag{3.25}$$

where A and B are the unperturbed system matrices, x is the state vector, u is the control vector, Λ represents the uncertainties in control effectiveness and $\Delta = W^T\sigma$ represents the non-linear system uncertainties, modeled as the product between an unknown weight matrix W and a known basis functions vector σ.
Starting again from the unperturbed system in equation (1.7), this time we consider a perturbation in both $H^\diamond$ e $C^\diamond$ matrices:

$$\hat{H}^\diamond\ddot{q} + \hat{C}^\diamond\dot{q} = \tau_q \tag{3.26}$$

The perturbation in inertia matrix can be expressed as $\hat{H}^\diamond = \alpha H^\diamond$, while the one in centrifugal matrix is expressed as $\hat{C}^\diamond = C^\diamond + \Delta C^\diamond$. The perturbations acting on the C matrix are now computed within the NDI and are no longer compensated for by the MRAC.
Isolating $\ddot{q}$:

$$\begin{aligned}\ddot{q} &= -H^{\diamond-1}\alpha^{-1}C^\diamond\dot{q} - H^{\diamond-1}\alpha^{-1}\Delta C^\diamond\dot{q} + H^{\diamond-1}\alpha^{-1}\tau_q\\ \ddot{q} &= -H^{\diamond-1}(\alpha^{-1} - \mathbf{I} + \mathbf{I})C^\diamond\dot{q} - H^{\diamond-1}\alpha^{-1}\Delta C^\diamond\dot{q} + H^{\diamond-1}\alpha^{-1}\tau_q\\ \ddot{q} &= -H^{\diamond-1}C^\diamond\dot{q} + H^{\diamond-1}[-(\alpha^{-1} - \mathbf{I})C^\diamond - \alpha^{-1}\Delta C^\diamond]\dot{q} + H^{\diamond-1}\alpha^{-1}\tau_q \qquad (3.27)\end{aligned}$$

The system is now written in the general reference form: x = $\dot{q}$, A = $-H^{\diamond-1}C^\diamond$, B = $H^{\diamond-1}$, $W^T = -(\alpha^{-1} - \mathbf{I})C^\diamond - \alpha^{-1}\Delta C^\diamond$, $\sigma = \dot{q}$, $\Lambda = \alpha^{-1}$ and $u = \tau_q$.

3.2.2 Reference model and nominal control

To obtain the nominal behaviour of the new model, we need to impose $\Delta = W^T\sigma = 0$, $\Lambda = \mathbf{I}$ and $\tau_q = \tau_{NDI}$. Doing so, we get the equation:

$$\ddot{q} = -H^{\diamond-1}C^\diamond\dot{q} + H^{\diamond-1}\tau_{NDI} \tag{3.28}$$

This corresponds to $\dot{x} = Ax + Bu$. We need now to design an NDI control able to decouple the system dynamics:

$$\tau_{NDI} = H^{\diamond}(K_2\dot{q}_d - K_1\dot{q}) + C^{\diamond}\dot{q} = -\hat{K}_1\dot{q} + \hat{K}_2\dot{q}_d + C^{\diamond}\dot{q} \tag{3.29}$$

where $\hat{K}_i = H^{\diamond}K_i$.
By substituting this expression in (3.28), we obtain the reference model dynamics:

$$\ddot{q}_r = -K_1\dot{q}_r + K_2\dot{q}_d = A_r\dot{q}_r + B_r\dot{q}_d \tag{3.30}$$

where the reference model matrices are $A_r = -K_1$ and $B_r = K_2$.
We can define once again the matrix E_p as a selector matrix of the components of the state vector we want to control: $z = E_p x$, where z is the vector containing the controlled state components. Since in our case we want to control all the velocities, $z = x$ and then $E_p = \mathbf{I}$. A_r and B_r need to satisfy the equation:

$$-E_p {A_r}^{-1} B_r = \mathbf{I} \rightarrow B_r = -A_r \rightarrow K_2 = K_1 = K \tag{3.31}$$

In conclusion, the reference model dynamics and the NDI control result to be:

$$\ddot{q}_r = -K(\dot{q}_r - \dot{q}_d) = A_r\epsilon_r \tag{3.32}$$

$$\tau_{NDI} = H^{\diamond}K(\dot{q}_d - \dot{q}) + C^{\diamond}\dot{q} = \hat{K}(\dot{q}_d - \dot{q}) + C^{\diamond}\dot{q} \tag{3.33}$$

3.2.3 Perturbed dynamics

We can manipulate equation (3.27) by adding and subtracting three couples of equal terms:

$$\ddot{q} = -H^{\diamond-1}C^{\diamond}\dot{q} + H^{\diamond-1}W^T\dot{q} + H^{\diamond-1}\alpha^{-1}\tau_q + H^{\diamond-1}(\hat{K}\dot{q}_d - \hat{K}\dot{q}_d + \hat{K}\dot{q} - \hat{K}\dot{q} + C^{\diamond}\dot{q} - C^{\diamond}\dot{q})$$

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond-1}\alpha^{-1}\tau_q + H^{\diamond-1}(W^T\dot{q} + \hat{K}\dot{q} - \hat{K}\dot{q}_d - C^{\diamond}\dot{q})$$

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond-1}\alpha^{-1}[\tau_q + \alpha W^T\dot{q} + \alpha(\hat{K}\dot{q} - \hat{K}\dot{q}_d - C^{\diamond}\dot{q})]$$

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond-1}\alpha^{-1}[\tau_{NDI} + \tau_a + \alpha W^T\dot{q} - \alpha\tau_{NDI}]$$

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond-1}\alpha^{-1}[(\mathbf{I} - \alpha)\tau_{NDI} + \tau_a + \alpha W^T\dot{q}]$$

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond-1}\alpha^{-1}\{(\mathbf{I} - \alpha)\tau_{NDI} + \tau_a + [(\alpha - \mathbf{I})C^{\diamond} - \Delta C^{\diamond}]\dot{q}\} \tag{3.34}$$

Defining $\bar{\sigma} = [\sigma; u_n] = [\dot{q}; \tau_{NDI}]$ and $\bar{W}^T = [\alpha W^T,\ \mathbf{I}-\alpha] = [(\alpha-\mathbf{I})C^{\diamond} - \Delta C^{\diamond},\ \mathbf{I}-\alpha]$:

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond-1}\alpha^{-1}[\bar{W}^T\bar{\sigma} + \tau_a] \tag{3.35}$$

That corresponds to the general form:

$$\dot{x} = A_r x + B_r x_d + B\Lambda[\bar{W}^T\bar{\sigma} + u_a]$$

3.2.4 MRAC

The MRAC logic and equations are exactly the same shown previously with the simplified NDI (3.16, 3.18 and 3.19). The only difference is that here the NDI includes an unperturbed centrifugal term, so that MRAC does not need to compensate also for it, but just for its perturbed counterpart. Basically, a little part of responsibility in the control is transferred from MRAC to NDI.

3.2.5 Modeling of momentum change

The equation (1.7) derived from considering a zero total system momentum δ in the complete equation:

$$H^{\diamond}\ddot{q} + C^{\diamond}\dot{q} + C^{\diamond}_{\delta}\delta = \tau_q \tag{3.36}$$

This assumption is perfectly valid for the unperturbed system, but it is not anymore after applying the perturbation, seen as the capture of an unknown target. As a consequence, the mathematical model of the perturbed system (equation 3.26) needs to be modified:

$$\hat{H}^{\diamond}\ddot{q} + \hat{C}^{\diamond}\dot{q} + \hat{C}^{\diamond}_{\delta}\delta = \tau_q \tag{3.37}$$

The system momentum after capture can be expressed as $\delta = \hat{H}_0\dot{q}_0 + \hat{H}_{0q}\dot{q}$, leading to:

$$\begin{aligned}\ddot{q} =& - H^{\diamond-1}C^{\diamond}\dot{q} + H^{\diamond-1}\{[-(\alpha^{-1} - \mathbf{I})C^{\diamond} - \alpha^{-1}\Delta C^{\diamond} - \alpha^{-1}\hat{C}^{\diamond}_{\delta}\hat{H}_{0q}]\dot{q} - \alpha^{-1}\hat{C}^{\diamond}_{\delta}\hat{H}_0\dot{q}_0\}\\ &+ H^{\diamond-1}\alpha^{-1}\tau_q\end{aligned} \tag{3.38}$$

The momentum that appears in consequence of the capture is considered as an additional perturbation to compensate for through the MRAC. By comparing this equation with (3.27), we can see that only W^T and σ change: $W^T = [-\alpha^{-1}\hat{C}^{\diamond}_{\delta}\hat{H}_0, -(\alpha^{-1} - \mathbf{I})C^{\diamond} - \alpha^{-1}\Delta C^{\diamond} - \alpha^{-1}\hat{C}^{\diamond}_{\delta}\hat{H}_{0q}]$, $\sigma = [\dot{q}_0; \dot{q}]$.
After some mathematical manipulations, the above equation can be written as:

$$\ddot{q} = -K\dot{q} + K\dot{q}_d + H^{\diamond-1}\alpha^{-1}\{(\mathbf{I}-\alpha)\tau_{NDI} + \tau_a - \hat{C}^{\diamond}_{\delta}\hat{H}_0\dot{q}_0 + [(\alpha-\mathbf{I})C^{\diamond} - \Delta C^{\diamond} - \hat{C}^{\diamond}_{\delta}\hat{H}_{0q}]\dot{q}\} \tag{3.39}$$

By defining $\bar{W}^T = [-\hat{C}^{\diamond}_{\delta}\hat{H}_0, (\alpha - \mathbf{I})C^{\diamond} - \Delta C^{\diamond} - \hat{C}^{\diamond}_{\delta}\hat{H}_{0q}, \mathbf{I} - \alpha]$ and $\bar{\sigma} = [\dot{q}_0; \dot{q}; \tau_{NDI}]$, we achieve to express the system in the same form of equation (3.14).
Practically, to compensate for the momentum change through MRAC, we just need to change the $\bar{\sigma}$ vector inside the model, to take into account also the base velocity vector $\dot{q}_0$.

3.2.6 Model validation

Validation was conducted exactly as explained in subsection 3.1.5 for the model with the simplified NDI: the robot behavior was tested under the different motions previously investigated, using the same gains found for the other model. *Motion 1* will not be further analyzed, since it represents a simplified version of *Motion 2*. Overall, the following results were obtained for joints motion controlled by complete NDI and MRAC:

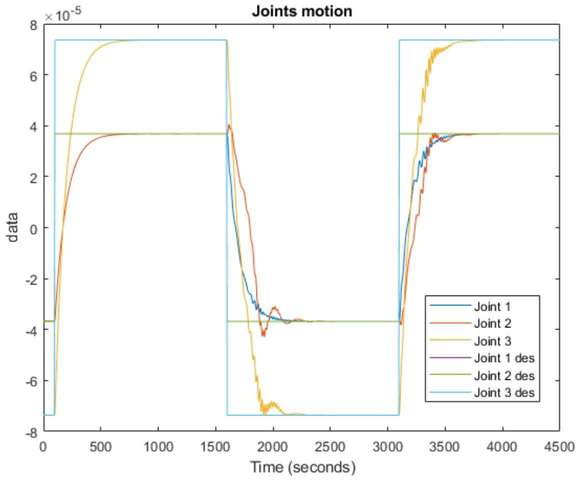

Figure 3.25: Motion 2 with complete NDI: Joints motion

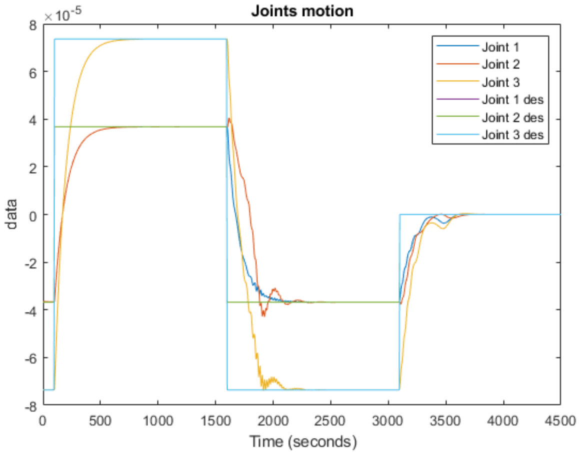

Figure 3.26: Motion 3 with complete NDI: Joints motion

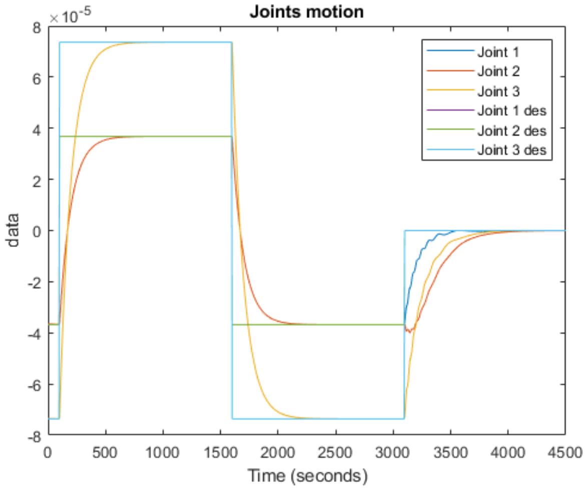

Figure 3.27: Motion 4 with complete NDI: Joints motion

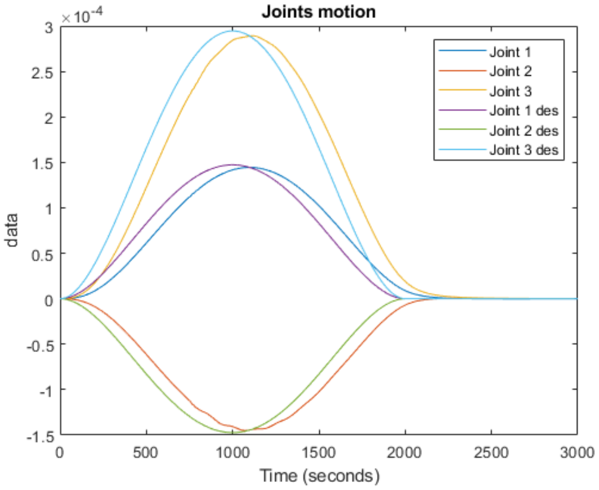

Figure 3.28: Motion 5 with complete NDI: Joints motion

After running several simulations for all the different types of motion previously investigated, we can conclude that the 2 models with simplified and complete NDI are basically characterized by the same results and performance: the error is slightly lower for the model with the complete NDI in the step motions, while the simplified version is slightly better in the smooth one. However, the differences are totally negligible. The results are summarized in the following table:

Type of motion	Simplified NDI	Complete NDI
Motion 2	8.1905e-06	8.1564e-06
Motion 3	7.4302e-06	7.2167e-06
Motion 4	5.8039e-06	5.8035e-06
Motion 5	9.0777e-07	9.0973e-07

Table 3.3: Joints motion errors comparison

This is due to the fact that in the expression $\tau_{NDI} = H^{\diamond}K(\dot{q}_d - \dot{q}) + C^{\diamond}\dot{q}$, the centrifugal/Coriolis term $C^{\diamond}\dot{q}$ is negligible with respect to the inertial one. Therefore, we can conclude that simplifying the NDI by considering only the inertial term does not have any relevant effect on final results, but it allows to benefit from a strong simplification of the mathematical model.

3.3 Tuning and gain synthesis

As discussed previously, MRAC control has been divided into n_q different channels, one for each DoF of the robot. Every channel has its own adaptation gain γ_i, that may be different from the others. These gain were originally determined through a *Trial&Error* method, which however results to be highly aleatory and time consuming. Therefore, a new gain synthesis has been elaborated in order to determine the gains analytically and fasten the model tuning.
At the moment of capture, huge perturbations can affect the system, leading to a strong increase in both e and $\bar{\sigma}$. This in turn results in a sharp rise of $\dot{\hat{W}}$, which brings instabilities and failures. Consequently, the goal is to adapt the gain γ so that MRAC adaptation rate $\dot{\hat{W}}$ (3.19) will not overcome a limit value as the perturbation changes (variation in target velocities and/or mass). By achieving this objective, the robustness of the control strategy will significantly increase.
By following an approach based on scalability notion [26], it can be proved that modifying the disturbances and the command profile by a scaling factor α, the adaptation gains need to be multiplied by a factor equal $\frac{1}{\alpha^2}$ in order to preserve the adaptation rate. The problem moves to the choice of a factor α representative of the perturbations change.
An effective solution was found by linking the choice of the scaling factor to the average excitation of the regressor $\bar{\sigma}$ [27], which then represents our pseudo scaling factor:

$$\phi = \frac{1}{T_{obs}} \int_0^{T_{obs}} \bar{\sigma}^T \bar{\sigma} dt \tag{3.40}$$

where T_{obs} is a representative observation time, long enough to properly observe the effects of perturbations. This choice is effective only if $\bar{\sigma} = [\dot{q}; \tau_{NDI}]$ is informative for system identification and consequently for MRAC computation. After defining X(t) and U(t) as:

$$U(t) := [\tau_{NDI}(0)\ \tau_{NDI}(1)\ ...\ \tau_{NDI}(t-1)]$$

$$X(t) := [\dot{q}(0)\ \dot{q}(1)\ ...\ \dot{q}(t-1)]$$

It can be proved that $\bar{\sigma}$ results to be informative for the system if the following condition holds [28]:

$$rank(\begin{bmatrix} X(t) \\ U(t) \end{bmatrix}) = dim(\dot{q}) + dim(\tau_{NDI}) = 2n_q \tag{3.41}$$

In our simulation, the above condition turns valid much earlier than the moment of capture. Since MRAC becomes important starting from that moment, the choice of $\bar{\sigma}$ for scaling is allowed.
Furthermore, the persistent excitation (PE) condition needs to be verified [29]:

$$\int_t^{t+T} \bar{\sigma}\bar{\sigma}^T dt \geq \zeta\mathbf{I},\ \forall\ t \tag{3.42}$$

where T is an arbitrary time interval (equal to the sample time of simulations for simplicity) and ζ is a constant > 0.
By considering the update law restricted to the single channels (3.20), in order to have the same adaptation rate in the computation of each control component, a new scaling factor is defined. This allows to have different gains for the different channels, by including the i-th column of B within the computation of the scaling factor for the i-th channel:

$$s_i = \sqrt{[PB(:,i)]^T \cdot \phi\ \cdot [PB(:,i)]} \tag{3.43}$$

At this point, the adaptation gain of each channel can be computed by exploiting its inverse proportionality with respect to s_i:

$$\gamma_i = \frac{1}{T_{ad}\ \cdot\ s_i\ +\ \epsilon_\gamma} \tag{3.44}$$

where T_{ad} serves as an adaptation period, as a factor that represents the desired rate of adaptation (the MRAC will be faster the lower T_{ad} is), and $\epsilon_\gamma = 1e-10$ is a very small correction factor, added to keep the formulation valid even in the case of $s_i = 0$.
Finally, since the RWs are poorly influenced by the perturbations caused by the target capture, two different adaptation rates are set for them and for the joints. In particular, the T_{ad} used for the joints results to be 4 or 5 orders of magnitude

lower than the one used for RWs.
The gains synthesis obtained through this process was validated by running several simulations with different values of target mass and velocities. In each simulation, the calculated gains brought satisfying results for the computation of the MRAC, proving the validity of the method.
However, this tuning strategy still requires an offline estimation of the parameters, that stay constant throughout the whole simulation and cannot adapt to time-varying perturbations.

3.3.1 Online tuning

To address the problems introduced above, an online version of the tuning strategy is proposed in this section. This allows to continuously update the gain parameters during the simulation, in order to obtain a MRAC that closely adapts to the dynamical evolution of the system.
The equations are modified as follows:

$$\phi(t) = \frac{1}{T}\int_{t}^{t+T} \bar{\sigma}^T(t)\bar{\sigma}(t)dt \tag{3.45}$$

where T is an arbitrary time interval, considered equal to the sample time of simulations for simplicity.

$$s_i(t) = \sqrt{[PB(:,i)]^T \cdot \phi(t) \ \cdot [PB(:,i)]} \tag{3.46}$$

$$\gamma_i(t) = \frac{1}{T_{ad} \ \cdot \ s_i(t) \ + \ \epsilon_\gamma} \tag{3.47}$$

As a result, MRAC gains γ_i are not computed offline anymore, but they are constantly evolving during the simulation to achieve a better adaptation to system variations and disturbances.

3.3.2 Lyapunov's Stability

The utilization of online varying adjustment parameters $\gamma(t)$ requires a new law to be written to ensure Lyapunov's stability. The following Lyapunov's function is proposed [20], slightly different from the one in (3.18):

$$V = \gamma(t)e^TPe + tr(\tilde{W}\Lambda^{\frac{1}{2}})^T(\tilde{W}\Lambda^{\frac{1}{2}}) \tag{3.48}$$

Since γ is not constant anymore, but it is a function of time, its time derivative needs to be computed within the expression of the derivative of the Lyapunov function:

$$\begin{aligned}\dot{V} &= \dot{\gamma}e^T Pe + \gamma 2e^T P\dot{e} + 2tr(\Lambda\tilde{W}^T\dot{\hat{W}}) \\ &= \dot{\gamma}e^T Pe + \gamma 2e^T P(A_r e - B\Lambda\tilde{W}^T\bar{\sigma}) + 2tr(\Lambda\tilde{W}^T\dot{\hat{W}}) \\ &= \dot{\gamma}e^T Pe + \gamma 2e^T PA_r e - \gamma 2e^T PB\Lambda\tilde{W}^T\bar{\sigma} + 2tr(\Lambda\tilde{W}^T\dot{\hat{W}}) \\ &= \dot{\gamma}e^T Pe + \gamma e^T(A_r^T P + PA_r)e - 2\Lambda\tilde{W}^T(\gamma\bar{\sigma}e^T PB - \dot{\hat{W}}) \\ &= -e^T(\gamma R - \dot{\gamma}P)e - 2\Lambda\tilde{W}^T(\gamma\bar{\sigma}e^T PB - \dot{\hat{W}})\end{aligned} \tag{3.49}$$

By choosing the update law defined in (3.19), the derivative of the Lyapunov function becomes:

$$\begin{aligned}\dot{V} &= -e^T(\gamma R - \dot{\gamma}P)e \\ &= -e^T\gamma Re + e^T\dot{\gamma}Pe \leq 0\end{aligned} \tag{3.50}$$

Consequently, the Lyapunov stability ($\dot{V} \leq 0$) is ensured if the following condition is respected in every channel of the MRAC:

$$e^T(t)\ \gamma_i(t)\ R\ e(t) - e^T(t)\ \dot{\gamma}_i(t)\ P\ e(t) \geq 0\ \ \forall\ t \tag{3.51}$$

The above condition was verified and resulted to be valid in all the performed simulations, ensuring Lyapunov's stability.

3.3.3 Enhanced formulation and stability proof

Although the previous method showed effective results, the choice of the parameter T_{ad} is still arbitrary and unrelated to system stability. To overcome this problem, the tuning structure was further improved, as explained below.
The following formulation for time-varying γ is considered:

$$\gamma = \frac{\gamma_0}{||\bar{\sigma}|| \cdot ||PB|| + \epsilon} \tag{3.52}$$

where $||\cdot||$ represents a short version for $||\cdot||_2$ and γ_0 is a positive scalar to compute through a proper optimization strategy, with the constraint of satisfying Lyapunov's stability. ϵ is a very low positive scalar such that $(||\bar{\sigma}|| \cdot ||PB(:,i)|| + \epsilon) > 0$ always. The same expression can be adapted to each MRAC channel:

$$\gamma_i = \frac{\gamma_0}{||\bar{\sigma}|| \cdot ||PB(:,i)|| + \epsilon} \tag{3.53}$$

By doing so, since the norm of ϵ is totally negligible, the norm of $\dot{\hat{W}}$ results to be:

$$||\dot{\hat{W}}|| = |\gamma_0| \cdot ||e|| \tag{3.54}$$

This choice of this specific expression for γ is driven by the following reasons:

- after capture, a strong and sudden increase in both $\bar{\sigma}$ and e is observed, leading to a huge rise in $\dot{\hat{W}}$, with the risk of making the simulation crash. Using a time-varying γ that is inversely proportional to the norm of $\bar{\sigma}$ will significantly reduce this phenomenon and ensure a proper adaptation rate at any moment.

- $||\dot{\hat{W}}||$ is still proportional to $||e||$, so that a faster adaptation is ensured when the error is bigger, and vice-versa.

- the term $||PB(:,i)||$ is added in order to obtain a rate of $\hat{W}$ that is as uniform as possible across all channels of MRAC.

In order to avoid excessively high rates in case of big errors, the error itself can be included inside γ_i expression:

$$\gamma_i = \frac{\gamma_0}{||\bar{\sigma}|| \cdot ||PB(:,i)|| \cdot e + \epsilon} \tag{3.55}$$

However, doing so, the adaptation rate results to be almost constant and we are not able anymore of ensuring a faster convergence when the error is bigger.
To address this problem while avoiding the divergence of $||\dot{\hat{W}}||$ in case of excessive errors, a possible further implementation is considering the following expression of γ_i:

$$\gamma_i = \frac{\gamma_0 \cdot s(e)}{||\bar{\sigma}|| \cdot ||PB(:,i)|| \cdot e + \epsilon} \tag{3.56}$$

where $s(e)$ is a saturation function (sigmoid or similar) acting on the error, so that $||\dot{\hat{W}}||$ will stay proportional to $||e||$ only up to a specific bound, in order to avoid excessively high rates in case of big errors.
By leveraging the Lyapunov's function and imposing upper boundaries to some variables, an optimization of γ_0 can be provided.
In order to do so, the following Lyapunov's function is considered:

$$V = e^T Pe + tr(\tilde{W}\Lambda^{\frac{1}{2}})^T(\tilde{W}\Lambda^{\frac{1}{2}}) \tag{3.57}$$

Its time derivative is computed as follows:

$$\dot{V} = \underbrace{2e^T P\dot{e}}_{1} + \underbrace{2tr(\Lambda\tilde{W}^T\dot{\hat{W}})}_{2} \tag{3.58}$$

The two terms will be analyzed separately, starting from term 1, for which a modified version of Equation (3.17) will be exploited. By knowing that $\tilde{W} = \hat{W} - \bar{W}$, we can assume to apply a proportional control on $\dot{\hat{W}}$ in order to make $\hat{W}$ tend to $\bar{W}$:

$$\dot{\hat{W}} = \hat{k}(\bar{W} - \hat{W}) = -\hat{k}\tilde{W}$$

$$\tilde{W} = -\underbrace{\frac{1}{\hat{k}}}_{k}\dot{\hat{W}} \tag{3.59}$$

where $k \in \mathbb{R}$, $k > 0$.
Therefore, we can express the derivative of the error as a function of the error itself and $\dot{\hat{W}}$:

$$\dot{e} = A_r e - B\Lambda\tilde{W}^T\bar{\sigma} = A_r e + B\Lambda k\dot{\hat{W}}^T\bar{\sigma} \tag{3.60}$$

The term $B\Lambda k$ can be upper-bounded by a scalar $\bar{k}$, by taking into account the known $\bar{\lambda}(B)$, estimating a value for the proportional constant k and conducting a workspace analysis on $\Lambda = \alpha^{-1}$.
We can then write an inequality for error time derivative:

$$\begin{aligned} \dot{e} &\leq A_r e + \bar{k}\dot{\hat{W}}^T\bar{\sigma} \\ \dot{e} &\leq A_r e + \dot{\hat{W}}^T G \end{aligned} \tag{3.61}$$

where G is a vector given by $G = \bar{k}\bar{\sigma}$.
The focus can now shift to term 2, where Equation (3.59) is exploited again:

$$2tr(\Lambda\tilde{W}^T\dot{\hat{W}}) = 2tr(-\Lambda k\dot{\hat{W}}^T\dot{\hat{W}}) = -2k \cdot tr(\Lambda\dot{\hat{W}}^T\dot{\hat{W}}) \tag{3.62}$$

Using the cyclic property of trace:

$$-2k \cdot tr(\Lambda\dot{\hat{W}}^T\dot{\hat{W}}) = -2k \cdot tr(\dot{\hat{W}}^T\dot{\hat{W}}\Lambda) \tag{3.63}$$

It can be proved that $\dot{\hat{W}}^T\dot{\hat{W}}$ is a positive semidefinite symmetric matrix:

$$(\dot{\hat{W}}^T\dot{\hat{W}})^T = \dot{\hat{W}}^T(\dot{\hat{W}}^T)^T = \dot{\hat{W}}^T\dot{\hat{W}} \rightarrow \dot{\hat{W}}^T\dot{\hat{W}} \textbf{ is symmetric}$$

$$\forall\ x,\ x^T(\dot{\hat{W}}^T\dot{\hat{W}})x = (\dot{\hat{W}}x)^T(\dot{\hat{W}}x) = ||\dot{\hat{W}}x||^2 \geq 0 \rightarrow$$

$$\rightarrow \dot{\hat{W}}^T\dot{\hat{W}} \textbf{ is positive semidefinite}$$

From Von Neumann's trace inequality, the following relation can be written:

$$-2k \cdot tr(\dot{\hat{W}}^T\dot{\hat{W}}\Lambda) \leq 2k \cdot |tr(\dot{\hat{W}}^T\dot{\hat{W}}\Lambda)| \leq 2k\sum_{i=1}^{n_q}\tilde{\lambda}_i(\dot{\hat{W}}^T\dot{\hat{W}})\tilde{\lambda}_i(\Lambda) \tag{3.64}$$

where $\tilde{\lambda}$ represents the matrix singular values. The above expression can be upbounded by the maximum singular value of Λ, that coincides with its spectral norm:

$$\sum_{i=1}^{n_q}\tilde{\lambda}_i(\dot{\hat{W}}^T\dot{\hat{W}})\tilde{\lambda}_i(\Lambda) \leq \tilde{\lambda}_{max}(\Lambda)\sum_{i=1}^{n_q}\tilde{\lambda}_i(\dot{\hat{W}}^T\dot{\hat{W}}) = ||\Lambda||\sum_{i=1}^{n_q}\tilde{\lambda}_i(\dot{\hat{W}}^T\dot{\hat{W}}) \tag{3.65}$$

Since $\dot{\hat{W}}^T\dot{\hat{W}}$ is a positive semidefinite symmetric matrix, its singular values coincide with its eigenvalues:

$$\sum_{i=1}^{n_q} \tilde{\lambda}_i(\dot{\hat{W}}^T\dot{\hat{W}}) = \sum_{i=1}^{n_q} \lambda_i(\dot{\hat{W}}^T\dot{\hat{W}}) = tr(\dot{\hat{W}}^T\dot{\hat{W}})$$

Substituting this expression in (3.64) and (3.65) and using the definition of Frebenius norm:

$$-2k \cdot tr(\dot{\hat{W}}^T\dot{\hat{W}}\Lambda) \leq 2k\ ||\Lambda||\ tr(\dot{\hat{W}}^T\dot{\hat{W}}) = 2k\ ||\Lambda||\ ||\dot{\hat{W}}||_F^2 \tag{3.66}$$

where $||\cdot||_F$ represents Frobenius matrix norm.
The following inequality between $||\cdot||_F$ and $||\cdot||$ is valid:

$$||\dot{\hat{W}}||_F^2 = \sum_{i=1}^{n_q} \tilde{\lambda}_i(\dot{\hat{W}}^T\dot{\hat{W}}) = \sum_{i=1}^{n_q} \tilde{\lambda}_i^2(\dot{\hat{W}}) \leq n_q \tilde{\lambda}_{max}^2(\dot{\hat{W}}) = n_q\ ||\dot{\hat{W}}||^2$$

It follows:

$$2k\ ||\Lambda||\ ||\dot{\hat{W}}||_F^2 \leq 2k\ n_q\ ||\Lambda||\ ||\dot{\hat{W}}||^2 \leq \tilde{k}\ ||\dot{\hat{W}}||^2 \tag{3.67}$$

where $\tilde{k}$ represents an upper bound of $2kn_q||\Lambda||$, obtained through a proper workspace analysis.
Using the expressions (3.61) and (3.67) within (3.58), it follows:

$$\dot{V} \leq \underbrace{2e^T P A_r e}_{3} + \underbrace{2e^T P \dot{\hat{W}}^T G}_{4} + \tilde{k}\ ||\dot{\hat{W}}||^2 \tag{3.68}$$

Once again, terms 3 and 4 will be analyzed separately.
The term 3 can be re-written as:

$$2e^T P A_r e = e^T(A_r^T P + P A_r)e = (A_r^T P + P A_r)||e||^2 \tag{3.69}$$

Young inequality can be used to further develop the term 4. Its general expression is:

$$2x^T y \leq \alpha||x||^2 + \frac{1}{\alpha}||y||^2$$

where $\alpha \in \mathbb{R}$, $\alpha > 0$.
Applying that to our case:

$$2e^T P \dot{\hat{W}} G \leq \alpha||e||^2 + \frac{1}{\alpha}||P\dot{\hat{W}}G||^2 \tag{3.70}$$

Using (3.69) and (3.70) within (3.68):

$$\dot{V} \leq (A_r^T P + P A_r)||e||^2 + \alpha||e||^2 + \frac{1}{\alpha}||P\dot{\hat{W}}G||^2 + \tilde{k}\ ||\dot{\hat{W}}||^2$$

$$\dot{V} \leq (A_r^T P + PA_r + \alpha\mathbf{I})||e||^2 + \frac{1}{\alpha}||P\dot{\hat{W}}G||^2 + \tilde{k}\ ||\dot{\hat{W}}||^2$$

$$\dot{V} \leq (A_r^T P + PA_r + \alpha\mathbf{I})||e||^2 + \frac{1}{\alpha}||P||^2\ ||G||^2\ ||\dot{\hat{W}}||^2 + \tilde{k}\ ||\dot{\hat{W}}||^2$$

$$\dot{V} \leq \underbrace{(A_r^T P + PA_r + \alpha\mathbf{I})}_{5}||e||^2 + \underbrace{(\frac{1}{\alpha}||P||^2\ ||G||^2 + \tilde{k})}_{\beta}\ ||\dot{\hat{W}}||^2 \tag{3.71}$$

The expression 5 represents the classical Lyapunov's equation and the resultant matrix needs to be negative definite in order to have Lyapunov's stability. It follows:

$$A_r^T P + PA_r + \alpha\mathbf{I} \leq -\bar{\alpha}\mathbf{I} \tag{3.72}$$

where $\bar{\alpha}$ is an arbitrary parameter such that $\bar{\alpha} \in \mathbb{R}$, $\bar{\alpha} > 0$.
It follows:

$$\dot{V} \leq -\bar{\alpha}\ ||e||^2 + \beta\ ||\dot{\hat{W}}||^2 \leq 0 \tag{3.73}$$

Equation (3.73) represents a condition to satisfy in order to have Lyapunov's stability. From that, an estimation of $|\gamma_0|$ can be obtained:

$$||\dot{\hat{W}}||^2 \leq \frac{\bar{\alpha}}{\beta}||e||^2 \tag{3.74}$$

Comparing Equations (3.54) and (3.74), it follows:

$$\gamma_0 = \sqrt{\frac{\bar{\alpha}}{\beta}} \tag{3.75}$$

The value of parameter γ_0 computed as above allows to always ensure Lyapunov's stability, even when γ_i vary in time.

3.3.4 LMI optimization

An optimization of parameters through a LMI approach can be proposed.
Preliminarily, a workspace analysis on inertia perturbation matrix needs to be conducted in order to estimate the maximum possible values of Λ and $\bar{\sigma}$, from which the parameters G and $\tilde{k}$ are obtained. After that, the parameter k will be defined a priori and varied within a specific range of values, in order to refine the optimization.
Finally, the condition expressed in Equation (3.72) must be imposed and the values of α, $\bar{\alpha}$ and P that maximize the parameter γ_0 will be found.
While the Lyapunov's equation is already linear with respect to the variables to be optimized, the same does not apply to the optimization condition on γ_0. In fact, it depends on the ratio between the two parameters $\bar{\alpha}$ and β. Since $||G||$ and $\tilde{k}$ are

constant, β in turn depends on the ratio between the squared norm of P and α. Therefore, since γ_0 is typically much bigger than one, the following optimization condition will be imposed:

$$(\gamma_0)_{max} \rightarrow (\frac{\beta}{\bar{\alpha}})_{min} \rightarrow (\frac{||P||^2}{\alpha\ \bar{\alpha}})_{min}$$

This is a highly nonlinear expression, not implementable in a LMI method. Consequently, a linearization strategy is proposed, where the goal becomes to minimize a new parameter ζ that satisfies the following inequality:

$$\frac{||P||^2}{\alpha\ \bar{\alpha}} \leq \zeta \tag{3.76}$$

$$||P||^2 \leq \zeta\ \alpha\ \bar{\alpha}$$

where $\zeta \in \mathbf{R}$, $\zeta \geq 0$.
However, the right hand side of the inequality is still nonlinear, since it is characterized by the product between two optimization variables. A Rotated Second-Order Cone (RSOC) method is implemented to address the problem. It consists to introduce an additional variable that satisfies the following relation:

$$\alpha\ \bar{\alpha} \geq t \tag{3.77}$$

where $t \in \mathbf{R}$, $t \geq 0$.
This can be translated in the following matrix expression:

$$\begin{bmatrix} \alpha & t \\ t & \bar{\alpha} \end{bmatrix} \geq 0 \tag{3.78}$$

The above condition is verified when:

$$\alpha\bar{\alpha} \geq t^2$$

Since the inequality (3.77) must be respected, the following additional condition on t is imposed:

$$t \geq 1 \rightarrow t \leq t^2 \leq \alpha\bar{\alpha}$$

Going back to (3.3.4), it follows:

$$||P||^2 \leq \zeta\ t \leq \zeta\ \alpha\ \bar{\alpha} \tag{3.79}$$

$$P^T P \leq \zeta\ t\ \mathbf{I}_{n_q}$$

Leveraging the Schur complement, it can be expressed through the following matrix inequality:

$$\begin{bmatrix} \zeta\ \mathbf{I}_{n_q} & P^T \\ P & t\ \mathbf{I}_{n_q} \end{bmatrix} \geq 0 \tag{3.80}$$

Finally, the original nonlinear optimization condition has been translated in a series of linear constraints that involve the variables α, $\bar{\alpha}$, ζ and t.
These constraints are here summarized:

- $\alpha \geq 0$;
- $\bar{\alpha} \geq 0$;
- $\zeta \geq 0$;
- $t \geq 1$;
- $A_r^T P + PA_r + \alpha\mathbf{I} + \bar{\alpha}\mathbf{I} \leq 0$;
- $\begin{bmatrix} \zeta\,\mathbf{I}_{n_q} & P^T \\ P & t\,\mathbf{I}_{n_q} \end{bmatrix} \geq 0$;
- $\begin{bmatrix} \alpha & t \\ t & \bar{\alpha} \end{bmatrix} \geq 0.$

The optimization goal is to minimize the parameter ζ, that corresponds to maximize γ_0. Different values of optimal γ_0 can be found by varying the parameter k. This is related to MRAC convergence velocity: the lower k is, the higher γ_0 and the faster the convergence will be. Consequently, k can be varied within a proper range based on simulation convergence requirements.

3.3.5 Acceptability range definition for γ_0

In this section, an alternative strategy to compute γ_0 will be presented. Instead of looking for an optimal exact value as proposed in Eq.(3.75), this method aims to find an optimal range in which γ_0 can vary while ensuring the Lyapunov stability of the system.
Consider the Lyapunov function:

$$V = \gamma e^T P e + \operatorname{tr}\left((\tilde{W}\Lambda^{\frac{1}{2}})^T(\tilde{W}\Lambda^{\frac{1}{2}})\right), \tag{3.81}$$

Taking its derivative, yields:

$$\dot{V} = \dot{\gamma} e^T P e + 2\gamma e^T P (A_r - B_p \Lambda \tilde{W}^T \bar{\sigma}) + 2\operatorname{tr}(\Lambda \tilde{W}^T \dot{\hat{W}}). \tag{3.82}$$

As detailed in [20], this can be rewritten as:

$$\dot{V} = -e^T(\gamma R - \dot{\gamma} P)e - 2\Lambda \tilde{W}^T(\gamma \bar{\sigma} e^T P B_p - \dot{\hat{W}}). \tag{3.83}$$

By selecting the update law from Eq.(3.19), Lyapunov stability is ensured if the following condition holds:

$$e^T \dot{\gamma} P e - e^T \gamma R e \leq 0. \tag{3.84}$$

Following [20], this leads to the bound:

$$\dot{V}_{MRAC} \leq (\dot{\gamma}_{max} \bar{\lambda}(P) - \gamma_{min} \underline{\lambda}(R)) ||e||^2_{max}, \tag{3.85}$$

where $\bar{\lambda}(P)$ and $\underline{\lambda}(R)$ denote the maximum and minimum eigenvalues, respectively. The derivative of γ is given with (3.52) by:

$$\dot{\gamma} = -\frac{\bar{\sigma} \cdot \dot{\bar{\sigma}}}{||\bar{\sigma}||^3 \, ||PB_p||} \gamma_0 = -\frac{\bar{\sigma} \cdot \dot{\bar{\sigma}}}{||\bar{\sigma}||^2} \gamma. \tag{3.86}$$

Substituting (3.86) in (3.85) and leveraging the Cauchy-Schwarz inequality to obtain $(-\bar{\sigma} \cdot \dot{\bar{\sigma}})_{max} = |\bar{\sigma} \cdot \dot{\bar{\sigma}}|_{max} \leq ||\bar{\sigma}||_{max} ||\dot{\bar{\sigma}}||_{max}$, it follows that:

$$\dot{V}_{MRAC} \leq \left[\left(\frac{||\bar{\sigma}||_{max} \, ||\dot{\bar{\sigma}}||_{max}}{||\bar{\sigma}||^2_{min}} \bar{\lambda}(P) \right) \gamma_{max} - \gamma_{min} \underline{\lambda}(R) \right] ||e||^2_{max} \tag{3.87}$$

Furthermore, to enhance Lyapunov's stability while ensuring that convergence is not too fast, which would be a source of oscillations, the following constraint is considered:

$$\begin{aligned} -2||e||^2_{max} \leq \dot{V}_{MRAC} \leq & \\ \leq \left[\left(\frac{||\bar{\sigma}||_{max} \, ||\dot{\bar{\sigma}}||_{max}}{||\bar{\sigma}||^2_{min}} \bar{\lambda}(P) \right) \gamma_{max} - \gamma_{min} \underline{\lambda}(R) \right] ||e||^2_{max} \leq -||e||^2_{max} & \end{aligned} \tag{3.88}$$

Substituting γ from (3.52) yields:

$$-2||PB_p|| \leq \gamma_0 \left(\frac{||\bar{\sigma}||_{max} \, ||\dot{\bar{\sigma}}||_{max}}{||\bar{\sigma}||^3_{min}} \bar{\lambda}(P) - \frac{\underline{\lambda}(R)}{||\bar{\sigma}||_{max}} \right) \leq -||PB_p|| \tag{3.89}$$

Finally, the following range of acceptable values is obtained:

$$\frac{-||PB_p|| \, ||\bar{\sigma}||_{max} ||\bar{\sigma}||^3_{min}}{||\dot{\bar{\sigma}}||_{max} ||\bar{\sigma}||^2_{max} \bar{\lambda}(P) - \underline{\lambda}(R) ||\bar{\sigma}||^3_{min}} \leq \gamma_0 \leq \frac{-2||PB_p|| \, ||\bar{\sigma}||_{max} ||\bar{\sigma}||^3_{min}}{||\dot{\bar{\sigma}}||_{max} ||\bar{\sigma}||^2_{max} \bar{\lambda}(P) - \underline{\lambda}(R) ||\bar{\sigma}||^3_{min}} \tag{3.90}$$

where $||x||_{max}$ and $||x||_{min}$ are respectively the maximal and minimal values of $||x||$, estimated through a proper workspace analysis.

3.4 Workspace analysis

3.4.1 α estimation

A workspace analysis has been conducted on the robot described in Section 1.1 to identify the boundaries within the perturbation on inertia matrix α can vary. In order to do that, a three-level nested loop was created to explore all the possible robot configurations between the initial and the final ones: indeed, each of the three joints displacements was varied between its initial and final positions imposed within the simulation and the variation of each component of α and of its norm was investigated. In order to distinguish the effects of perturbation on RWs and joints, α matrix was subdivided in the following blocks:

$$\alpha = \begin{bmatrix} \alpha_r & \alpha_{rm} \\ \alpha_{mr} & \alpha_m \end{bmatrix} \tag{3.91}$$

where r stands for reaction wheels and m for manipulator joints. In particular, the focus will be on matrices $\alpha_r \in \mathbf{R^{n_r x n_r}}$ and $\alpha_m \in \mathbf{R^{n_m x n_m}}$. The terms of these matrices will be numbered from 1 to 9, proceeding in order from the first to the last row and taking the terms from left to right ($\alpha_1 = \alpha(1,1)$, $\alpha_2 = \alpha(1,2)$, and so on). Finally, this process was repeated for different target masses and inertia, in order to see the influence of target characteristics on perturbation magnitude. The results are collected in the following subsections.

1^{st} scenario

The first simulation is characterized by a target mass of 8 kg, much lower than spacecraft base mass (100 kg).

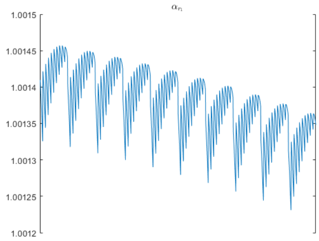

Figure 3.29: Simulation 1: α_{r_1} variation

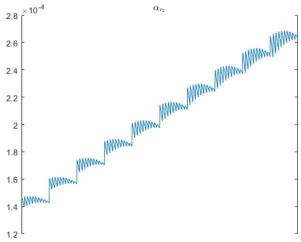

Figure 3.30: Simulation 1: α_{r_2} variation

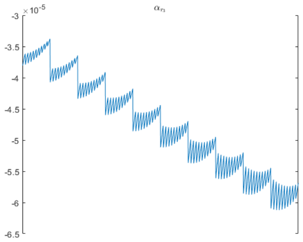

Figure 3.31: Simulation 1: α_{r_3} variation

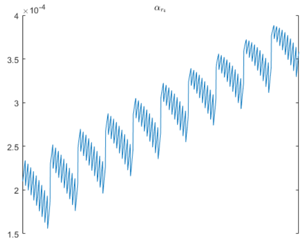

Figure 3.32: Simulation 1: α_{r_4} variation

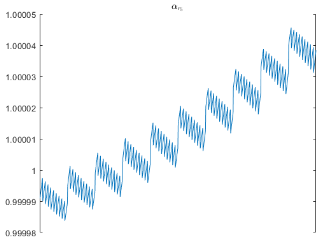

Figure 3.33: Simulation 1: α_{r_5} variation

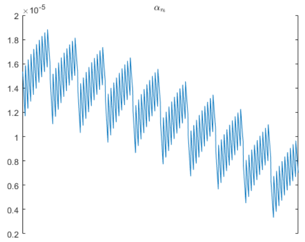

Figure 3.34: Simulation 1: α_{r_6} variation

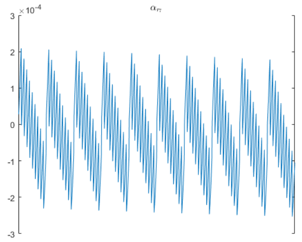

Figure 3.35: Simulation 1: α_{r_7} variation

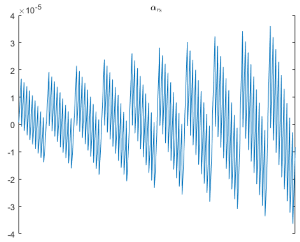

Figure 3.36: Simulation 1: α_{r_8} variation

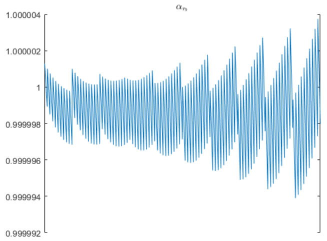

Figure 3.37: Simulation 1: α_{r_9} variation

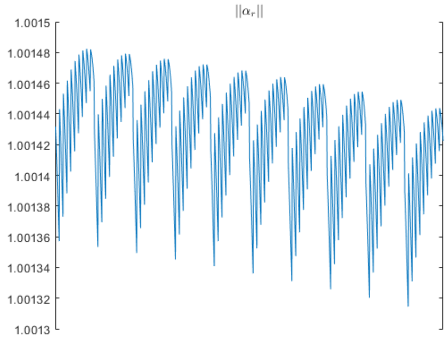

Figure 3.38: Simulation 1: norm(α_r) variation

Component	**Max**	**Min**	**Mean**
Alpha1	1.0015	1.0012	1.0014
Alpha2	2.6870e-04	1.3960e-04	2.0475e-04
Alpha3	-3.3771e-05	-6.1216e-05	-4.7637e-05
Alpha4	3.8835e-04	1.5587e-04	2.8384e-04
Alpha5	1.00005	0.99998	1.00001
Alpha6	1.8832e-05	3.3284e-06	1.1625e-05
Alpha7	2.0913e-04	-2.5306e-04	-3.5264e-05
Alpha8	3.6061e-05	-3.6345e-05	1.8409e-08
Alpha9	1.00000	0.99999	1.00000
Norm(α,2)	1.0015	1.0013	1.0014

Table 3.4: Simulation 1: α_r components

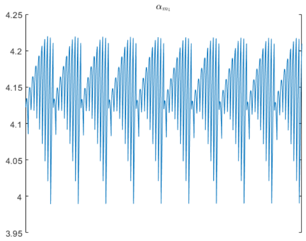

Figure 3.39: Simulation 1: α_{m_1} variation

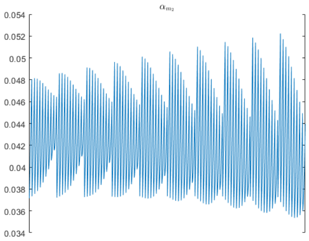

Figure 3.40: Simulation 1: α_{m_2} variation

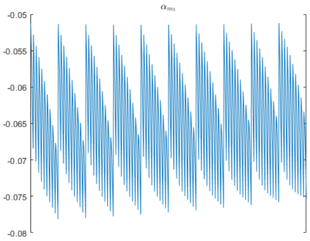

Figure 3.41: Simulation 1: α_{m_3} variation

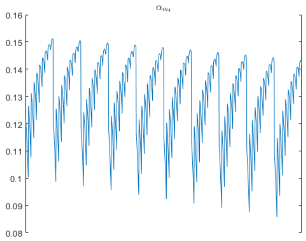

Figure 3.42: Simulation 1: α_{m_4} variation

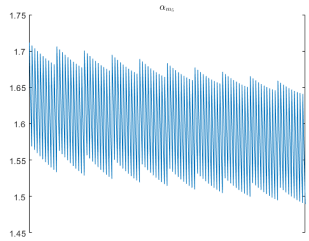

Figure 3.43: Simulation 1: α_{m_5} variation

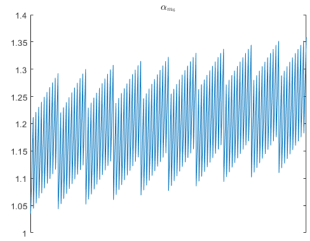

Figure 3.44: Simulation 1: α_{m_6} variation

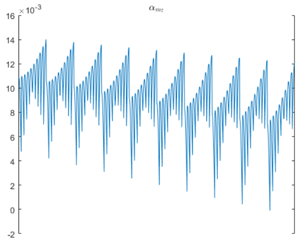

Figure 3.45: Simulation 1: α_{m7} variation

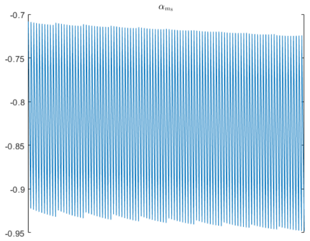

Figure 3.46: Simulation 1: α_{m8} variation

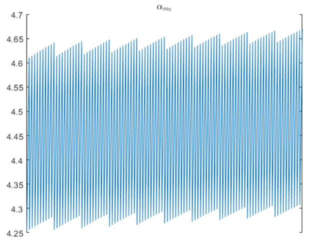

Figure 3.47: Simulation 1: α_{m9} variation

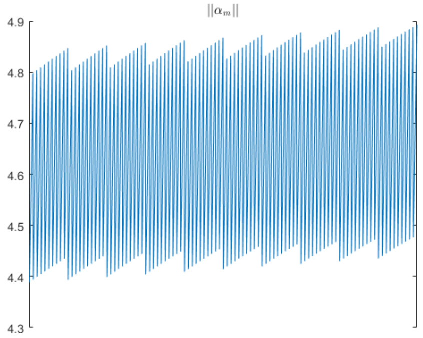

Figure 3.48: Simulation 1: norm(α_m) variation

Component	**Max**	**Min**	**Mean**
Alpha1	4.2197	3.9894	4.1425
Alpha2	0.0522	0.0354	0.0420
Alpha3	-0.0512	-0.0782	-0.0649
Alpha4	0.1513	0.0858	0.1292
Alpha5	1.7117	1.4894	1.5978
Alpha6	1.3590	1.0351	1.1976
Alpha7	0.0140	-9.1855e-05	0.0093
Alpha8	-0.7082	-0.9482	-0.8268
Alpha9	4.6693	4.2528	4.4561
Norm(α,2)	4.8924	4.3886	4.6343

Table 3.5: Simulation 1: α_m components

2^{nd} scenario

The second simulation is characterized by a target mass of 80 kg, that is close to spacecraft base mass (100 kg).

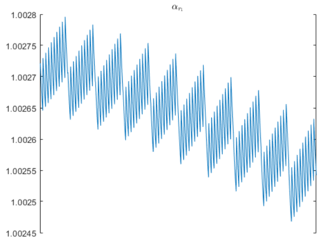

Figure 3.49: Simulation 2: α_{r_1} variation

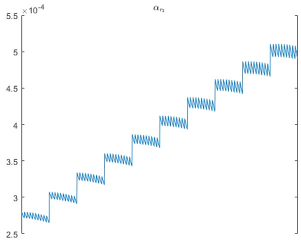

Figure 3.50: Simulation 2: α_{r_2} variation

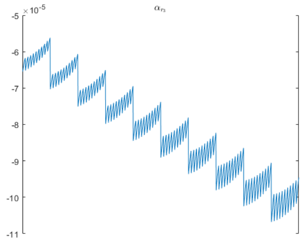

Figure 3.51: Simulation 2: α_{r_3} variation

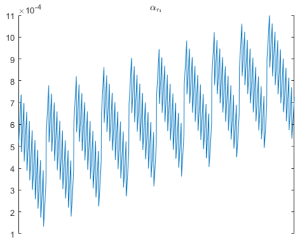

Figure 3.52: Simulation 2: α_{r_4} variation

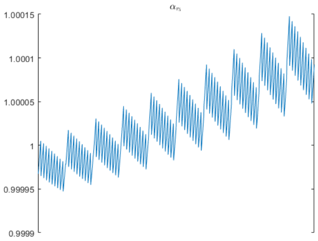

Figure 3.53: Simulation 2: α_{r_5} variation

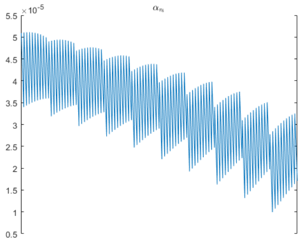

Figure 3.54: Simulation 2: α_{r_6} variation

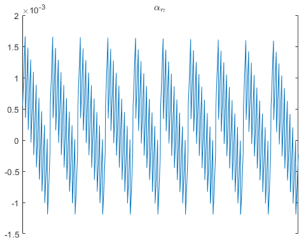

Figure 3.55: Simulation 2: α_{r_7} variation

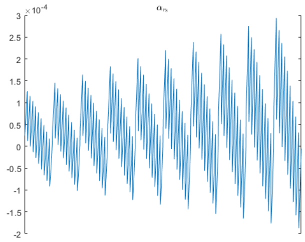

Figure 3.56: Simulation 2: α_{r_8} variation

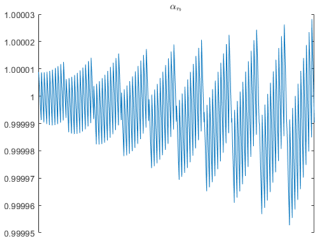

Figure 3.57: Simulation 2: α_{r_9} variation

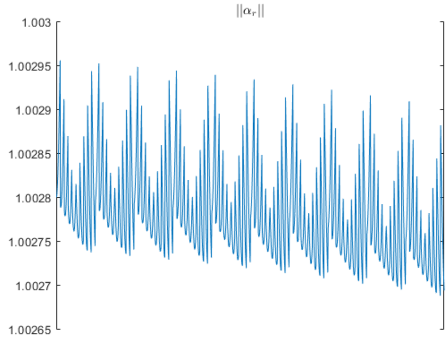

Figure 3.58: Simulation 2: norm(α_r) variation

Component	**Max**	**Min**	**Mean**
Alpha1	1.0028	1.0025	1.0026
Alpha2	5.1074e-04	2.6484e-04	3.8887e-04
Alpha3	-5.6178e-05	-1.0676e-04	-8.1353e-05
Alpha4	0.0011	1.3234e-04	6.2607e-04
Alpha5	1.0001	0.9999	1.0000
Alpha6	5.1111e-05	9.9519e-06	3.3514e-05
Alpha7	0.0017	-0.0012	1.7726e-04
Alpha8	2.9336e-04	-1.8720e-04	3.1837e-05
Alpha9	1.00003	0.99995	0.99999
Norm(α,2)	1.0030	1.0027	1.0028

Table 3.6: Simulation 2: α_r components

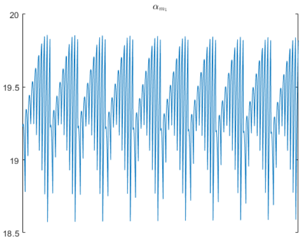

Figure 3.59: Simulation 2: α_{m_1} variation

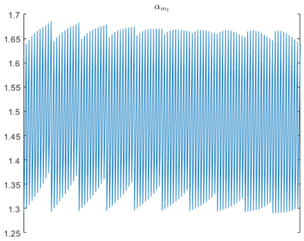

Figure 3.60: Simulation 2: α_{m_2} variation

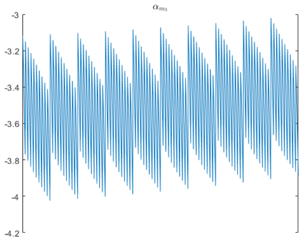

Figure 3.61: Simulation 2: α_{m_3} variation

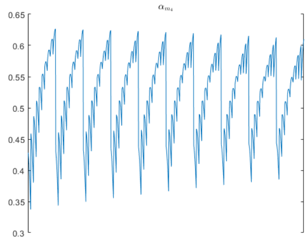

Figure 3.62: Simulation 2: α_{m_4} variation

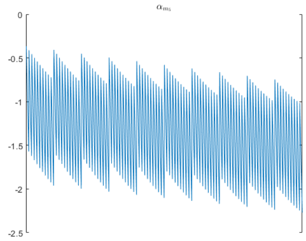

Figure 3.63: Simulation 2: α_{m_5} variation

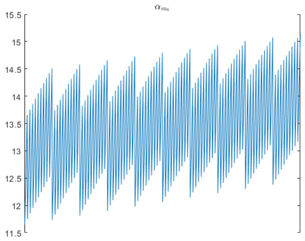

Figure 3.64: Simulation 2: α_{m_6} variation

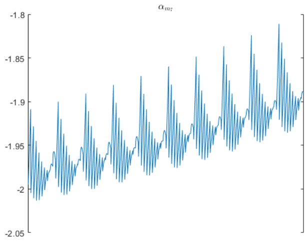

Figure 3.65: Simulation 2: α_{m_7} variation

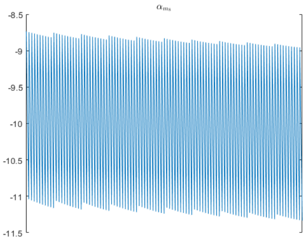

Figure 3.66: Simulation 2: α_{m_8} variation

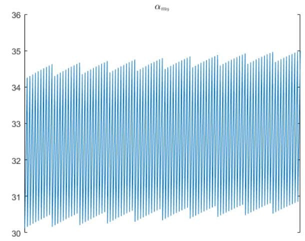

Figure 3.67: Simulation 2: α_{m_9} variation

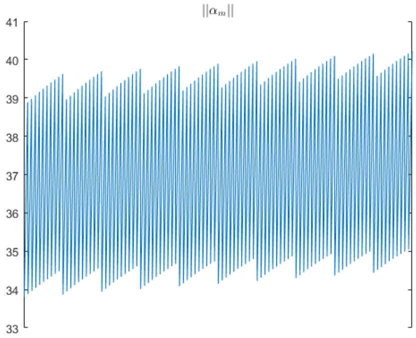

Figure 3.68: Simulation 2: norm(α_m) variation

Component	**Max**	**Min**	**Mean**
Alpha1	19.8578	18.5732	19.3705
Alpha2	1.6858	1.2902	1.4808
Alpha3	-3.0209	-4.0238	-3.5139
Alpha4	0.6263	0.3373	0.5224
Alpha5	-0.3654	-2.2708	-1.3382
Alpha6	15.1431	11.6480	13.4026
Alpha7	-1.8111	-2.0133	-1.9404
Alpha8	-8.7350	-11.3333	-10.0194
Alpha9	35.0060	30.1173	32.5093
Norm(α,2)	40.2166	33.7975	36.9449

Table 3.7: Simulation 2: α_m components

3^{rd} scenario

The third simulation is characterized by a target mass of 240 kg, more than double the spacecraft base mass (100 kg).

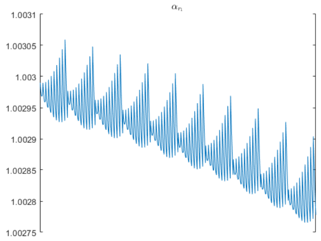

Figure 3.69: Simulation 3: α_{r_1} variation

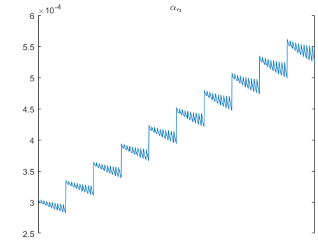

Figure 3.70: Simulation 3: α_{r_2} variation

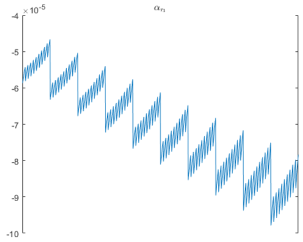

Figure 3.71: Simulation 3: α_{r_3} variation

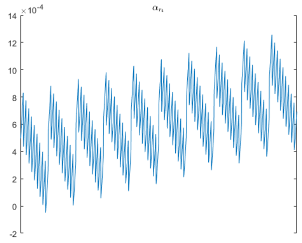

Figure 3.72: Simulation 3: α_{r_4} variation

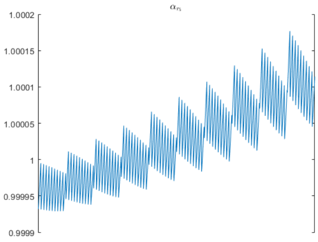

Figure 3.73: Simulation 3: α_{r_5} variation

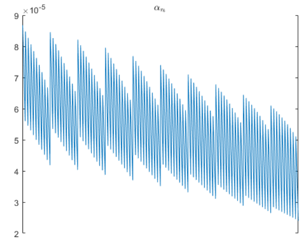

Figure 3.74: Simulation 3: α_{r_6} variation

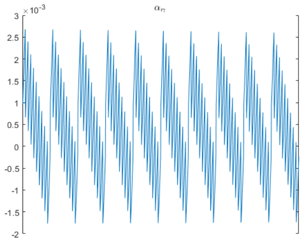

Figure 3.75: Simulation 3: α_{r_7} variation

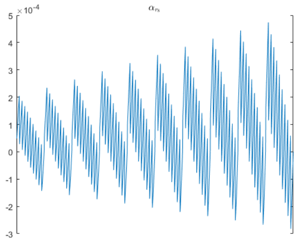

Figure 3.76: Simulation 3: α_{r_8} variation

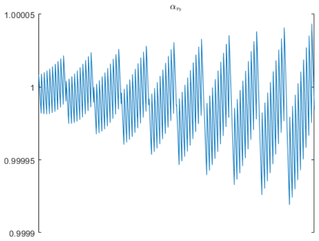

Figure 3.77: Simulation 3: α_{r_9} variation

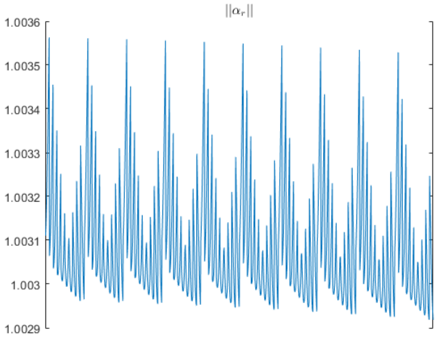

Figure 3.78: Simulation 3: norm(α_r) variation

Component	**Max**	**Min**	**Mean**
Alpha1	1.0031	1.0028	1.0029
Alpha2	5.6147e-04	2.8219e-04	4.2183e-04
Alpha3	-4.6666e-05	-9.7838e-05	-7.1047e-05
Alpha4	0.0013	-4.8781e-05	6.0501e-04
Alpha5	1.0002	0.9999	1.0000
Alpha6	8.6790e-05	2.4019e-05	5.3397e-05
Alpha7	0.0027	-0.0018	3.8716e-04
Alpha8	4.7359e-04	-2.8224e-04	5.8289e-05
Alpha9	1.0000	0.9999	1.0000
Norm(α,2)	1.0036	1.0029	1.0031

Table 3.8: Simulation 3: α_r components

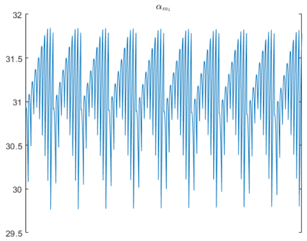

Figure 3.79: Simulation 3: α_{m_1} variation

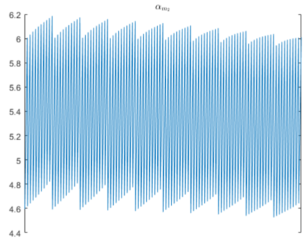

Figure 3.80: Simulation 3: α_{m_2} variation

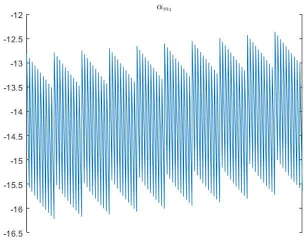

Figure 3.81: Simulation 3: α_{m_3} variation

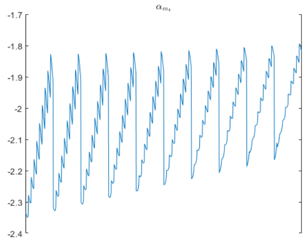

Figure 3.82: Simulation 3: α_{m_4} variation

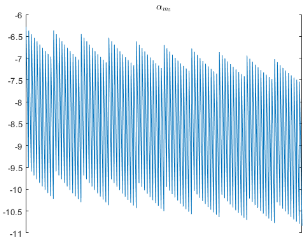

Figure 3.83: Simulation 3: α_{m_5} variation

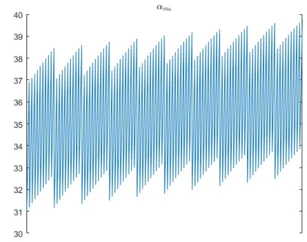

Figure 3.84: Simulation 3: α_{m_6} variation

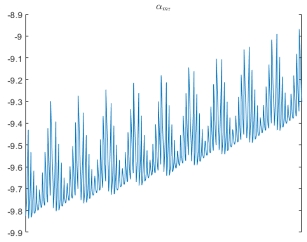

Figure 3.85: Simulation 3: α_{m_7} variation

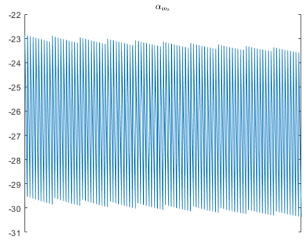

Figure 3.86: Simulation 3: α_{m_8} variation

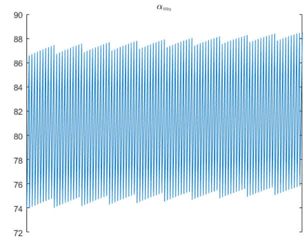

Figure 3.87: Simulation 3: α_{m_9} variation

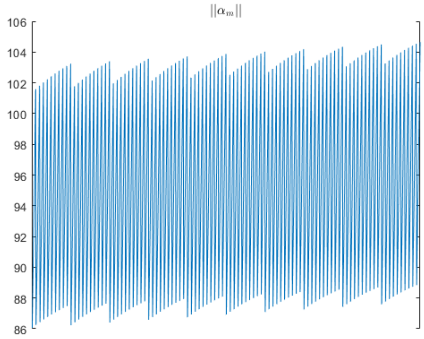

Figure 3.88: Simulation 3: norm(α_m) variation

Component	Max	Min	Mean
Alpha1	31.8447	29.7661	31.0914
Alpha2	6.1853	4.5263	5.3317
Alpha3	-12.3615	-16.2157	-14.2466
Alpha4	-1.7945	-2.3499	-2.0449
Alpha5	-6.2831	-10.8356	-8.5854
Alpha6	39.7426	30.9946	35.3466
Alpha7	-8.9685	-9.8370	-9.5142
Alpha8	-22.8412	-30.3640	-26.5602
Alpha9	88.5203	73.8966	81.0471
Norm(α,2)	104.6442	86.0617	95.1524

Table 3.9: Simulation 3: α_m components

Conclusions

It can be stated that α_r matrix shows a common behavior in all the 3 presented simulations: the diagonal terms oscillate around the unit value, while the non-diagonal ones are significantly lower. With a few exceptions, no major variations are observed in all α_r components throughout the configuration changes, probably due to the small motion taken as a use case. As expected, all the components tend to increase when the target mass is higher, since this implies a stronger perturbation on the system. However, this increase is not relevant, so that the norm of α_r is very close to the unit value even when the target mass is significantly higher than the one of the spacecraft. Overall, α_r can be basically approximated with the identity matrix, which means that the perturbation introduced by the target capture has

poor effects on reaction wheels inertial behavior.
On the other hand, α_m matrix presents a more interesting behavior. Once again, the diagonal terms are the most significant, taking on values well above unity. Non-diagonal terms are lower, but they can not be neglected, since most of them are still comparable with the diagonal ones. Actually, in some cases they are even higher (compare the diagonal term α_5 with the non-diagonal α_6). α_m components do not vary greatly throughout configuration changes, but they strongly increase when the target mass grows. This results in a significant increase in the norm of α_m, that rockets from 5 in the first simulation to more than 100 in the last one. That means that, unlike the reaction wheels, joints inertial behavior is importantly influenced by the magnitude of the perturbation, resulting in a strong variation of H matrix.
Since the perturbation of inertia matrix is compensated for by the MRAC, a stronger adaptive control is required on the joints, while just the NDI control could be sufficient for RWs. Consequently, a much lower adaptation rate is needed for MRAC components which act on them. That can be achieved by using different values of k (3.59) for joints and RWs. Another option is to specialize and differentiate the MRAC formulation for these two categories, so that control will adapt to their different needs.

3.4.2 $||\bar{\sigma}||$ and $||\dot{\bar{\sigma}}||$ estimation

Another workspace analysis has been conducted on the same robot (see Section 1.1) to identify the maximum values of $||\bar{\sigma}||$ and $||\dot{\bar{\sigma}}||$ within a prescribed workspace. A step motion as the one described in Section 2.5 was taken into account to determine the maximum allowed velocities and accelerations of the system.
The variables under analysis are computed assuming maximal actuator capability, with:

$$\tau_{NDI_{max}} \leq H^{\diamond}\ddot{q}_{max}, \tag{3.92}$$

The remaining term, $||\dot{\bar{\sigma}}||_{max}$, is approximated as:

$$||\dot{\bar{\sigma}}||_{max} \leq \left\| \begin{bmatrix} \ddot{q} & \dot{\tau}_{NDI} \end{bmatrix} \right\|_{max} \leq \left\| \begin{bmatrix} \ddot{q} & \dot{H}^{\diamond}\ddot{q} \end{bmatrix} \right\|_{max} \tag{3.93}$$

These quantities were evaluated for different joints configurations obtained through a nested loop simulation and their values are represented as a function of EE 3D position in Figures 3.89 and 3.90.

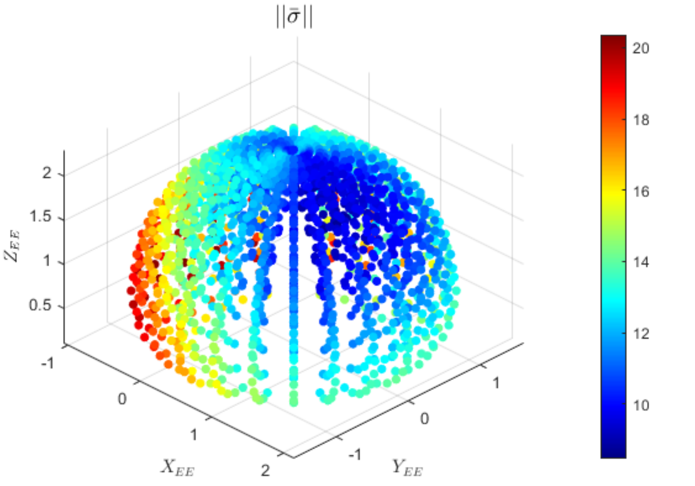

Figure 3.89: $||\bar{\sigma}||_{max}$ as a function of EE position

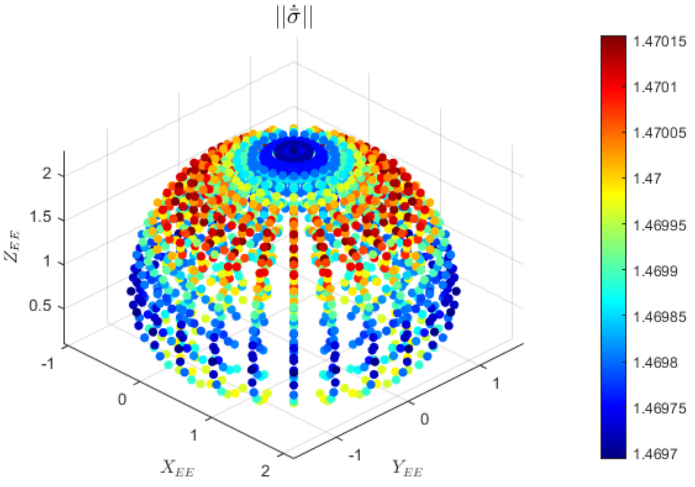

Figure 3.90: $||\dot{\bar{\sigma}}||_{max}$ as a function of EE position

While $||\bar{\sigma}||_{max}$ varies significantly within the considered workspace (Fig. 3.89), $||\dot{\bar{\sigma}}||_{max}$ remains nearly constant for slow motions typical of on-orbit operations (Fig. 3.90), so that a mean value can be considered for all the possible configurations without introducing significant errors.

3.5 High-frequency oscillations suppression

The high gains required within MRAC in order to have fast adaptation may induce high-frequency oscillations in the system dynamics. These oscillations may affect system stability and cause dangerous vibrations and mechanical stress. Consequently, it becomes fundamental to handle them properly.
As shown in [24], a low-pass filter acting on weights estimation update law can be implemented to address that problem. The filter is characterized by the following law:

$$\dot{\hat{W}}_f = \Gamma_f[\hat{W}(t) - \hat{W}_f(t)], \quad \hat{W}_f(0) = \hat{W}(0) \tag{3.94}$$

where Γ_f is a positive-definite gain matrix and $\hat{W}_f$ is the filtered version of the estimated weights matrix.

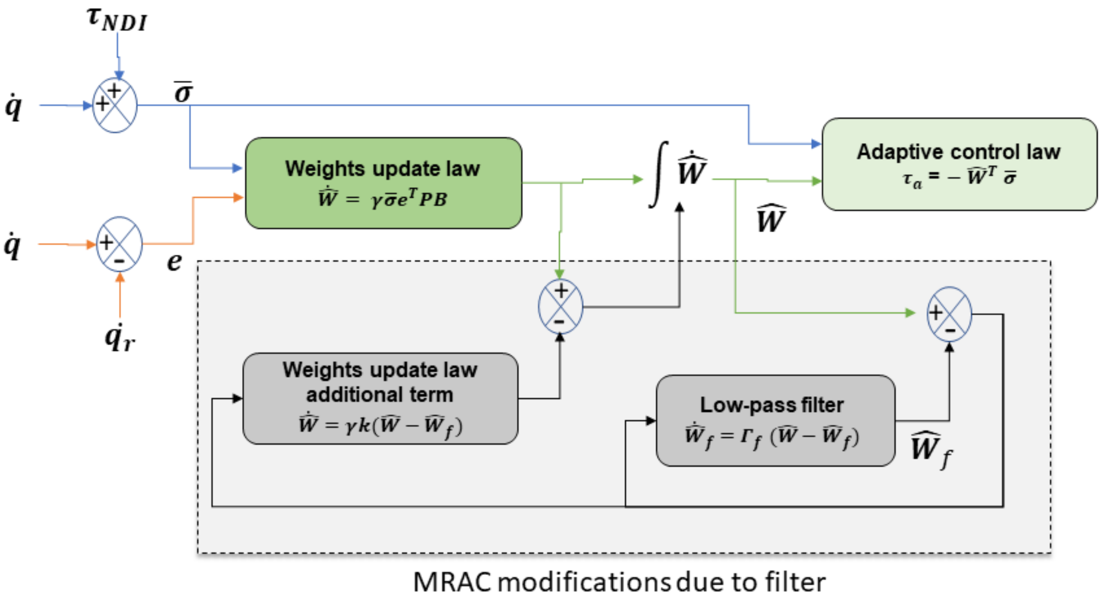
τ_{NDI}
$\dot{q}$
+
+
$\bar{\sigma}$
Weights update law
$\dot{\widehat{W}} = \gamma\bar{\sigma}e^T PB$
$\int \dot{\widehat{W}}$
Adaptive control law
$\tau_a = -\widehat{W}^T\bar{\sigma}$
$\widehat{W}$
$\dot{q}$
+
-
e
$\dot{q}_r$
+
-
+
-
Weights update law additional term
$\dot{\widehat{W}} = \gamma k(\widehat{W} - \widehat{W}_f)$
Low-pass filter
$\dot{\widehat{W}}_f = \Gamma_f(\widehat{W} - \widehat{W}_f)$
$\widehat{W}_f$
MRAC modifications due to filter

Figure 3.91: MRAC controller with filter modifications

Without losing generality, Γ_f can be considered as diagonal and, as a consequence, Equation (3.94) can be re-written and adapted to each MRAC channel:

$$\dot{\hat{W}}_{f_i} = \Gamma_{f_i}[\hat{W}_i(t) - \hat{W}_{f_i}(t)], \quad \hat{W}_{f_i}(0) = \hat{W}_i(0) \tag{3.95}$$

A new version of the weights matrix update law (3.19) is proposed as well, including a new term that takes into account the presence of the filter:

$$\dot{\hat{W}} = \gamma[\bar{\sigma}e^T PB - k_f(\hat{W} - \hat{W}_f)] \tag{3.96}$$

where $k_f \in \mathbf{R}$ is a positive modification gain.
Again, the law can adapted to each MRAC channel:

$$\dot{\hat{W}}_i = \gamma_i[\bar{\sigma} e^T PB(:,i) - k_f(\hat{W}_i - \hat{W}_{f_i})] \tag{3.97}$$

To prove the system stability, the following augmented Lyapunov's function represents a good candidate:

$$V = e^T Pe + tr(\tilde{W}\Lambda^{\frac{1}{2}})^T \gamma^{-1}(\tilde{W}\Lambda^{\frac{1}{2}}) + \bar{\sigma} tr(\tilde{W}_f^T \Gamma_f^{-1} \tilde{W}_f) \tag{3.98}$$

where $\tilde{W}_f = \hat{W}_f - W$. As proved in [24], this new update law ensures Lyapunov's stability and asymptotic convergence to zero of the error.
The effectiveness of the method can be seen in Figures 3.92 and 3.93, where a step reference motion is applied to the joints of the robot described in Section 1.1, and the results with and without filter are compared.

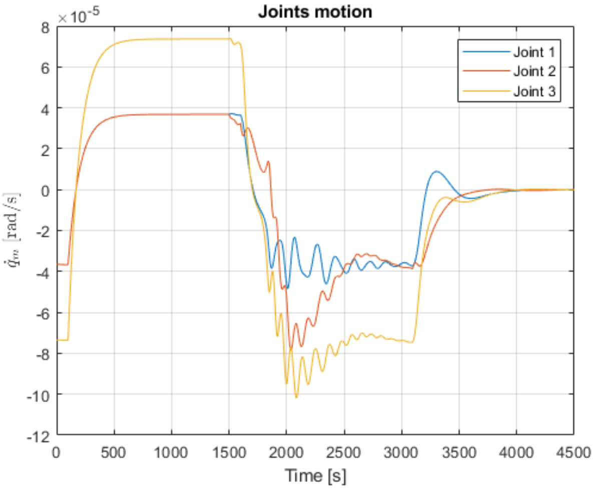

Figure 3.92: High frequency oscillations control without filter

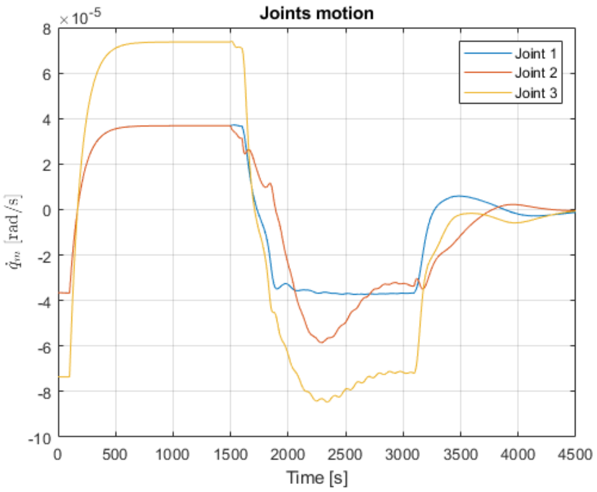

Figure 3.93: High frequency oscillations control with filter

It is clear how high-frequency oscillations deriving from high MRAC gains are strongly dampened through this method, at the expense of a slightly slower convergence to the reference model.

3.6 Study case

The study case chosen to analyze the effectiveness of the strategy is the same one described in 2.5. This choice derives from the possibility of comparing the results of the two different control strategies, investigating analogies and differences.

3.6.1 Simulation results

As shown in Fig.3.94, the servicer base is successfully stabilized, since its rotational velocities are dampened to almost zero values by the end of the simulations. That means that the momentum transferred by the target to the base during the capture is effectively compensated by the reaction wheels.

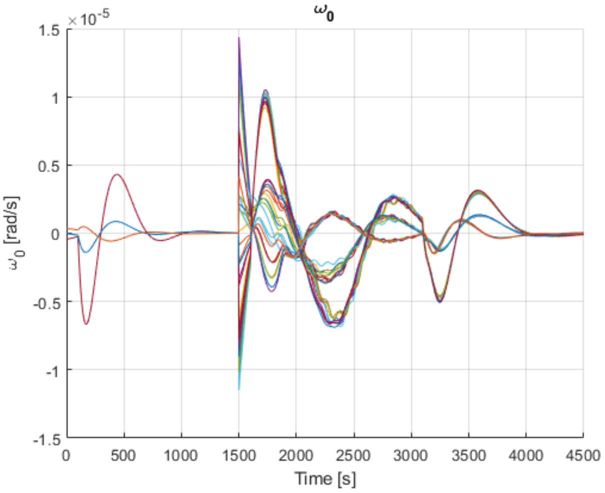

Figure 3.94: Base angular velocities for each run

Fig.3.95, Fig.3.96 and Fig.3.97 show the comparison between the real joints motion and the desired one. Fig.3.95 is characterized by only NDI control: the absence of MRAC results in a less accurate motion, with a slower convergence. In Fig.3.96, although the joints follow quite well the reference profile, big oscillations are observed due to high MRAC gains, since the filter described in (3.94) is not implemented. Its effectiveness is shown in Fig.3.97, where the oscillations are suppressed, at the expense of a slightly slower convergence.

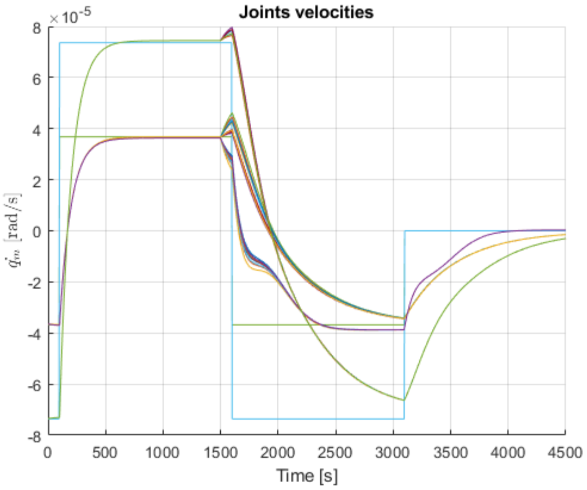

Figure 3.95: Joints velocities with NDI for each run

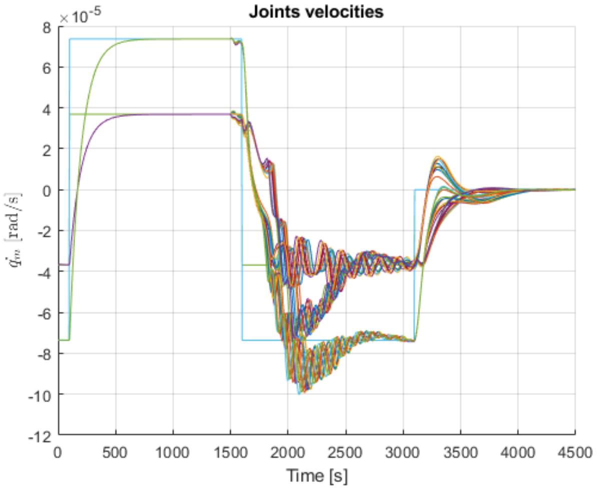

Figure 3.96: Joints velocities with NDI-MRAC for each run

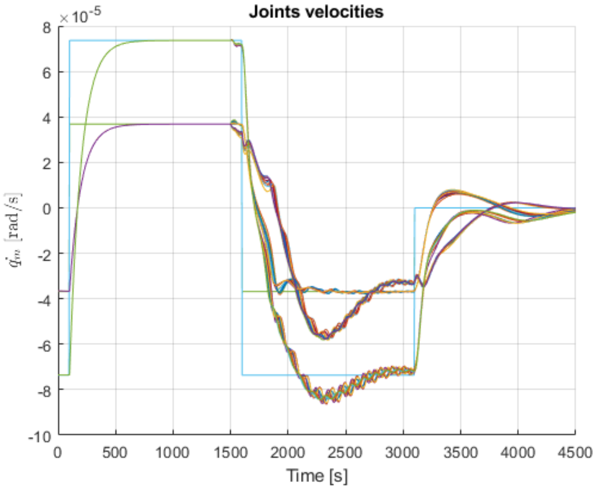

Figure 3.97: Joints velocities with NDI-MRAC+filter for each run

Looking at Fig.3.98, it is clear how the convective term is totally negligible with respect to the inertial one within NDI formulation. Consequently, this serves as proof for the choice of a simplified NDI control (Equation 3.3).

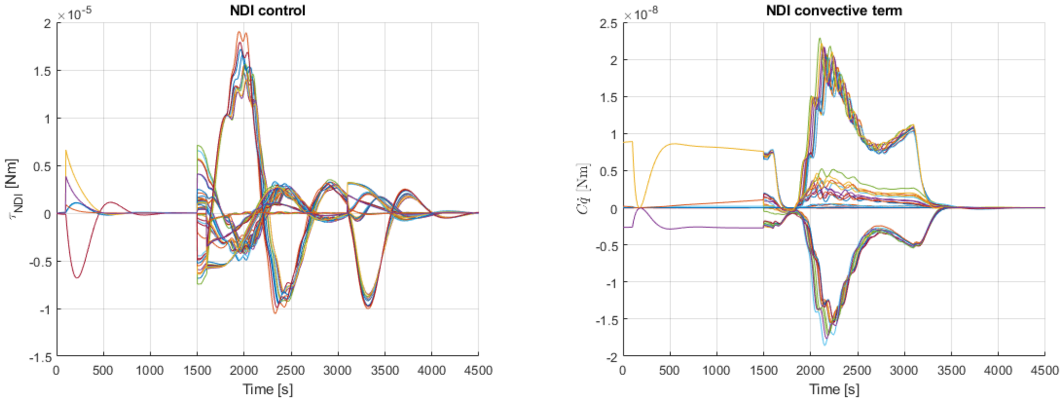

Figure 3.98: Comparison between simplified NDI and convective term

Fig. 3.99 shows the values of MRAC torques for all the simulations. Overall, they result to be approximately one order of magnitude lower than NDI ones. MRAC contribution to the global control is totally negligible before capture, when the perturbations are basically absent. On the contrary, it becomes relevant after capture, with the goal of compensating for the unknown disturbances introduced by the target.

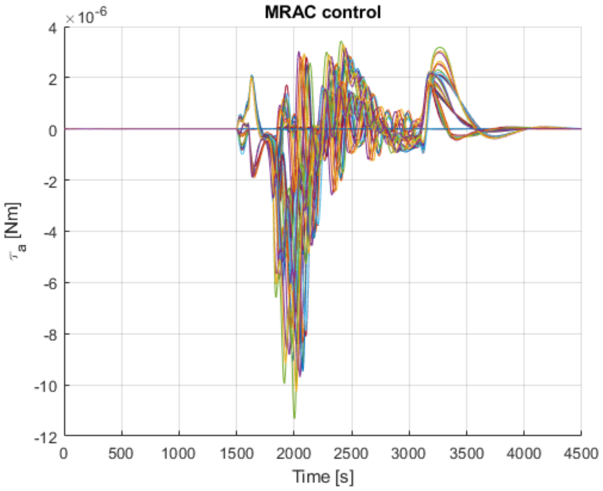

Figure 3.99: MRAC torques without filter for each run

Comparing Figures 3.99 and 3.100, it is clear how the filter allows to reduce the significant oscillations that characterize MRAC torques, resulting in a more stable dynamics and a more precise movement.

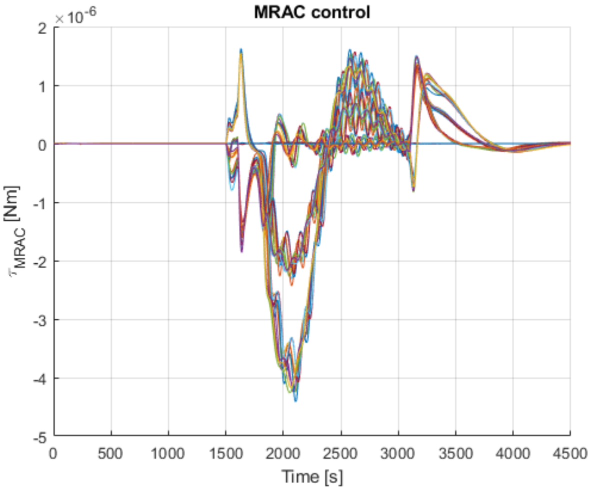

Figure 3.100: MRAC torques with filter for each run

Finally, Figures 3.101-3.103 show the evolution of momenta throughout the capture maneuver, proving the conservation of momentum for the global system. In particular, as previously assumed, the servicer momenta are zero before capture, they start varying at the capture moment and they finally stabilize to constant values. On the other hand, target momenta are constant before capture, they start varying at the capture moment and they finally stabilize to zero. The global system momenta are given by the sum of servicer and target momenta, and they remain constant during the whole capture, since the applied forces and torques are internal to the system.

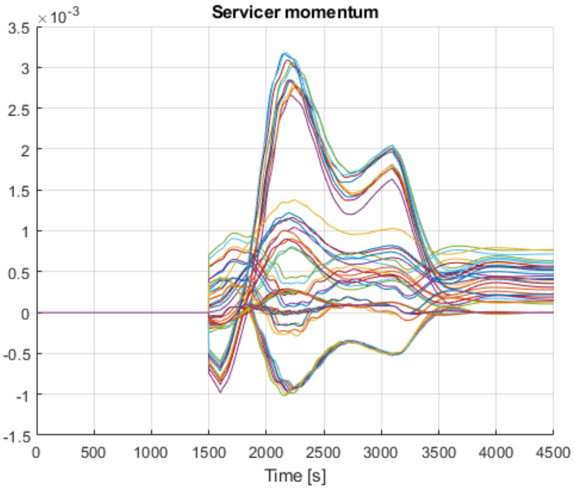

Figure 3.101: Servicer momenta evolution for each run

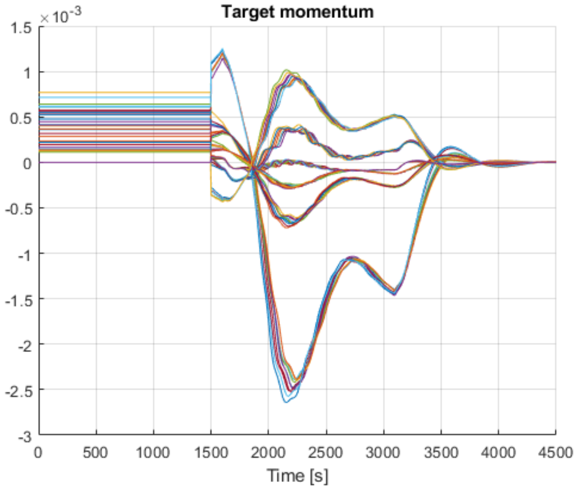

Figure 3.102: Target momenta evolution for each run

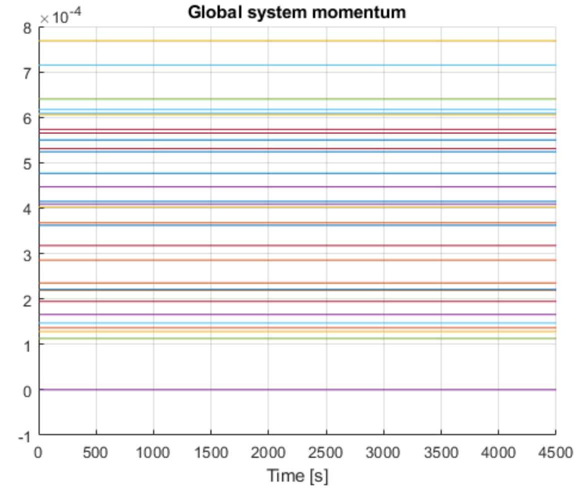

Figure 3.103: System momenta evolution for each run

3.6.2 Conclusions

In conclusion, MRAC results to successfully perform its task: it counteracts effectively the perturbations due to capture, leading to enhanced performances and faster convergence. As already done in Chapter 2, the MSEs related to joints velocities can be used as representative indicators of the effectiveness of the control strategy. Tab 3.10 compares the errors of NDI, NDI-MRAC and NDI-MRAC+filter.

Simulation ID	No MRAC	MRAC	MRAC + filter
Sim 1	1.0034e-09	2.2959e-10	2.1346e-10
Sim 2	9.5732e-10	2.2368e-10	2.0694e-10
Sim 3	9.8275e-10	2.1783e-10	2.0897e-10
Sim 4	1.0017e-09	2.2312e-10	2.1132e-10
Sim 5	1.0157e-09	2.3605e-10	2.1841e-10
Sim 6	1.0230e-09	2.3470e-10	2.2076e-10
Sim 7	9.8285e-10	2.2950e-10	2.1733e-10
Sim 8	9.5120e-10	2.2512e-10	1.9811e-10
Sim 9	9.5172e-10	2.1323e-10	2.0035e-10
Sim 10	9.6251e-10	2.1753e-10	1.9920e-10

Table 3.10: Mean squared errors in joints velocities with MRAC

Once again, the overall errors result to be very low for all the three strategies, as proof of the quality of the control. However, MRAC allows to reduce the error of each simulation by a factor between 4 and 5, demonstrating its effectiveness in compensating for the disturbance. The implementation of the low-pass filter inside the MRAC logic allows to further reduce the errors, thanks to its ability of dampening the MRAC high-frequency oscillations, that cause a deviation of joints motion from the reference, increasing the error.

3.6.3 Comparison with NDO

Finally, a comparison between NDI-NDO and NDI-MRAC control strategies will be presented in this section. Although both the methods can be used to compensate for unknown perturbations and disturbances, NDO and MRAC have important theoretical differences:

- NDO estimates and compensates unknown disturbances or model uncertainties affecting the system by designing an observer that reconstructs the disturbance and subtracting it from the control input.
- MRAC aims to make the plant output follow a reference model with unknown parameters by adapting controller parameters online to make the closed-loop behave like the reference.

Basically, NDO estimates and cancels the disturbance, while MRAC learns and adapts the parameters to match a reference model.
As shown in Tab. 3.11 and Fig. 3.104, MRAC strategy results to be clearly the best one, with errors that are half the ones with NDO, which is still better than NDI alone.

Simulation ID	NDO	MRAC
Sim 1	5.1862e-10	2.2959e-10
Sim 2	5.0352e-10	2.2368e-10
Sim 3	5.1492e-10	2.1783e-10
Sim 4	5.2036e-10	2.2312e-10
Sim 5	5.2079e-10	2.3605e-10
Sim 6	5.2634e-10	2.3470e-10
Sim 7	5.1330e-10	2.2950e-10
Sim 8	5.0267e-10	2.2512e-10
Sim 9	5.0389e-10	2.1323e-10
Sim 10	5.0696e-10	2.1753e-10

Table 3.11: Mean squared errors in joints velocities: NDO vs MRAC

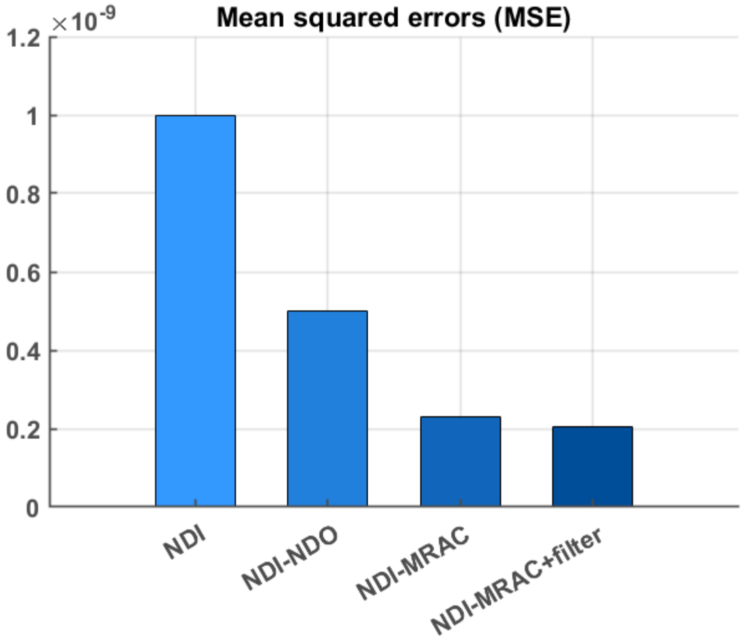

Figure 3.104: MSE in joints velocities

Chapter 4

Conclusions and future developments

In this final chapter, some possible further developments of the MRAC control strategy will be proposed and the overall conclusions of the thesis will be presented.

4.1 Adaptation of the strategy to more complex robots

One of the possible next steps could be the adaptation of MRAC control strategy to more elaborated robot models, like the one with the VISPA arm presented in Section 2.1. As seen in Chapter 2, NDI-NDO control strategy is not able to guarantee a fast convergence when applied to this model. Consequently, it could be interesting to analyze the performance of MRAC when applied to the same robot. Some slight modifications will need to be performed in the control logic before running the simulations.

Overall, MRAC control results to be easily adaptable to different robot models and operational scenarios, so that multiple simulations can be performed to test and compare its performances under several real working conditions. Some correlations might be found between MRAC effectiveness and simulations features, such as target mass, robot configuration, reference velocities profile, and so on.

4.2 Set-theoretic MRAC

An interesting further development of MRAC control strategy is implementing a set-theoretic MRAC: instead of estimating the unknown parameters as precise values, it makes them vary within estimated ranges, in order to guarantee that

the error between the system and the reference model will stay lower than the one obtained in an a-priori worst case scenario. The adaptive law and control input are designed to ensure that the system's state and parameters always remain inside pre-defined admissible sets, despite uncertainties and adaptation.
A projection operator needs to be defined [23]:

$$\text{Proj}(\theta, y) \triangleq \begin{cases} y, & \text{if } \psi(\theta) < 0, \\ y, & \text{if } \psi(\theta) \geq 0 \text{ and } \psi'(\theta)y \leq 0, \\ y - \dfrac{\psi'^{\ T}(\theta)\psi'(\theta)y}{\psi'(\theta)\psi'^{T}(\theta)}\,\psi(\theta), & \text{if } \psi(\theta) \geq 0 \text{ and } \psi'(\theta)y > 0 \end{cases} \tag{4.1}$$

where $\psi : \mathbf{R}^n \rightarrow \mathbf{R}$ is a continously differentiable convex function defined as:

$$\psi(\theta) \triangleq \frac{(\epsilon_\theta + 1)\theta^T\theta - \theta_{max}^2}{\epsilon_\theta\ \theta_{max}^2} \tag{4.2}$$

$\theta_{max} \in \mathbf{R}$ represents a projection bound imposed on the norm of $\theta \in \mathbf{R}^n$ and ϵ_θ is a projection tolerance bound.
A generalized restricted potential function must be defined as well. A possible candidate is the following:

$$\phi(||e(t)||_P) = \frac{||e(t)||_P^2}{\epsilon^2 - ||e(t)||_P^2} \tag{4.3}$$

where $||e(t)||_P = \sqrt{e^T(t)Pe(t)}$.
The derivative of this restricted potential function is:

$$\phi_d(||e(t)||_P) = \frac{\epsilon^2}{(\ \epsilon^2 - ||e(t)||_P^2\ )^2} \tag{4.4}$$

Finally, a new weights update law with guaranteed boundedness can be formulated:

$$\dot{\hat{W}} = \gamma\ Proj(\phi_d(||e(t)||_P)\ \bar{\sigma}\ e^T\ P\ B) \tag{4.5}$$

This is in the same form of Equation 3.19, but with the introduction of the projection operator to bound the error.
Implementing this new approach within the proposed MRAC control strategy will allow to guarantee at least a pre-defined performance level as the simulation conditions vary, resulting in a more reliable and robust control. However, the problem shifts to the definition of boundaries and projection tolerances, that need to be properly estimated through specific workspace analysis.

4.3 Implementation of flexible appendages

Another significant MRAC application could be the compensation for the unknown dynamics and disturbances due to the addition of flexible appendages to the robot model. In fact, strong dynamics coupling and vibration effects can arise because of their characteristic geometry and light weight. NDI-MRAC control strategy can be used to counteract these perturbations, achieving dynamic decoupling and robustness without requiring an accurate model of the flexible dynamics.
The dynamics of this kind of systems is expressed by the following equation [13]:

$$\underbrace{\begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{0q} & \mathbf{H}_{0\eta} \\ \star & \mathbf{H}_q & \mathbf{H}_{q\eta} \\ \star & \star & \mathbf{H}_\eta \end{bmatrix}}_{\mathbf{H}(\mathbf{q}_0,\mathbf{q})} \begin{bmatrix} \ddot{\mathbf{q}}_0 \\ \ddot{\mathbf{q}} \\ \ddot{\boldsymbol{\eta}} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{0q} & \mathbf{C}_{0\eta} \\ \mathbf{C}_{q0} & \mathbf{C}_q & \mathbf{C}_{q\eta} \\ \mathbf{C}_{\eta 0} & \mathbf{C}_{\eta q} & \mathbf{C}_\eta \end{bmatrix}}_{\mathbf{C}(\mathbf{q}_0,\mathbf{q},\dot{\mathbf{q}}_0,\dot{\mathbf{q}},\dot{\boldsymbol{\eta}})} \begin{bmatrix} \dot{\mathbf{q}}_0 \\ \dot{\mathbf{q}} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{6\times 1} \\ \mathbf{0}_{n_q\times 1} \\ \mathbf{K}_{\boldsymbol{\eta}}\boldsymbol{\eta} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{6\times 1} \\ \boldsymbol{\tau}_q \\ \mathbf{0}_{n_\eta\times 1} \end{bmatrix} \tag{4.6}$$

where $\boldsymbol{\eta} \in \mathbb{R}^{n_\eta \times 1}$ is the generalized flexible modal displacement vector, gathering all modal displacement among the different flexible bodies, and $\mathbf{K}_\eta$ is a diagonal and positive stiffness matrix.
Manipulating Equation (4.6), the system dynamics can be expressed in the following joint space equivalent form:

$$\mathbf{H}^\diamond(\mathbf{q},\mathbf{q}_0)\ddot{\mathbf{q}} + \mathbf{C}^\diamond(\mathbf{q},\mathbf{q}_0,\dot{\mathbf{q}},\dot{\mathbf{q}}_0,\dot{\boldsymbol{\eta}})\dot{\mathbf{q}} + \boldsymbol{\tau}_\eta = \boldsymbol{\tau}_q \tag{4.7}$$

where the equivalent inertia and convective matrices are defined as:

$$\mathbf{H}^\diamond = \mathbf{H}_q - \begin{bmatrix} \mathbf{H}_{0q}^T & \mathbf{H}_{q\eta} \end{bmatrix} \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{0\eta} \\ \mathbf{H}_{0\eta}^T & \mathbf{H}_\eta \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{H}_{0q} \\ \mathbf{H}_{q\eta}^T \end{bmatrix}$$

$$\mathbf{C}^\diamond = \mathbf{C}_q - \begin{bmatrix} \mathbf{H}_{0q}^T & \mathbf{H}_{q\eta} \end{bmatrix} \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{0\eta} \\ \mathbf{H}_{0\eta}^T & \mathbf{H}_\eta \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C}_{0q} \\ \mathbf{C}_{\eta q} \end{bmatrix}$$

All the effects of flexible modes are collected in the fictitious torque τ_n:

$$\begin{aligned} \boldsymbol{\tau}_\eta = &\left(\begin{bmatrix} \mathbf{C}_{q0} & \mathbf{C}_{q\eta} \end{bmatrix} - \begin{bmatrix} \mathbf{H}_{0q}^T & \mathbf{H}_{q\eta} \end{bmatrix} \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{0\eta} \\ \mathbf{H}_{0\eta}^T & \mathbf{H}_\eta \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{0\eta} \\ \mathbf{C}_{\eta 0} & \mathbf{C}_\eta \end{bmatrix} \right) \begin{bmatrix} \dot{\mathbf{q}}_0 \\ \dot{\boldsymbol{\eta}} \end{bmatrix} \\ &- \begin{bmatrix} \mathbf{H}_{0q}^T & \mathbf{H}_{q\eta} \end{bmatrix} \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{0\eta} \\ \mathbf{H}_{0\eta}^T & \mathbf{H}_\eta \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0}_{6\times n_\eta} \\ \mathbf{K}_{\boldsymbol{\eta}} \end{bmatrix} \boldsymbol{\eta} = \mathbf{D}^\diamond \begin{bmatrix} \dot{\mathbf{q}}_0 \\ \dot{\boldsymbol{\eta}} \\ \boldsymbol{\eta} \end{bmatrix} \end{aligned} \tag{4.8}$$

Therefore, the goal of MRAC is to adapt the control parameters in order to compensate for this additional unknown disturbance τ_n, allowing the NDI to work on a classical simpler dynamics, similar to the one defined earlier in Chapter 1.

This will allow to extend the scope of application of MRAC to a multitude of new systems and operating scenarios.
Some preliminary simulations were conducted on the robot described in Section 2.1. Fig.4.1 shows the comparison between the desired joints motion profile and the real one with NDI-MRAC control technique. The evolution of MRAC and reference errors are also diplayed.

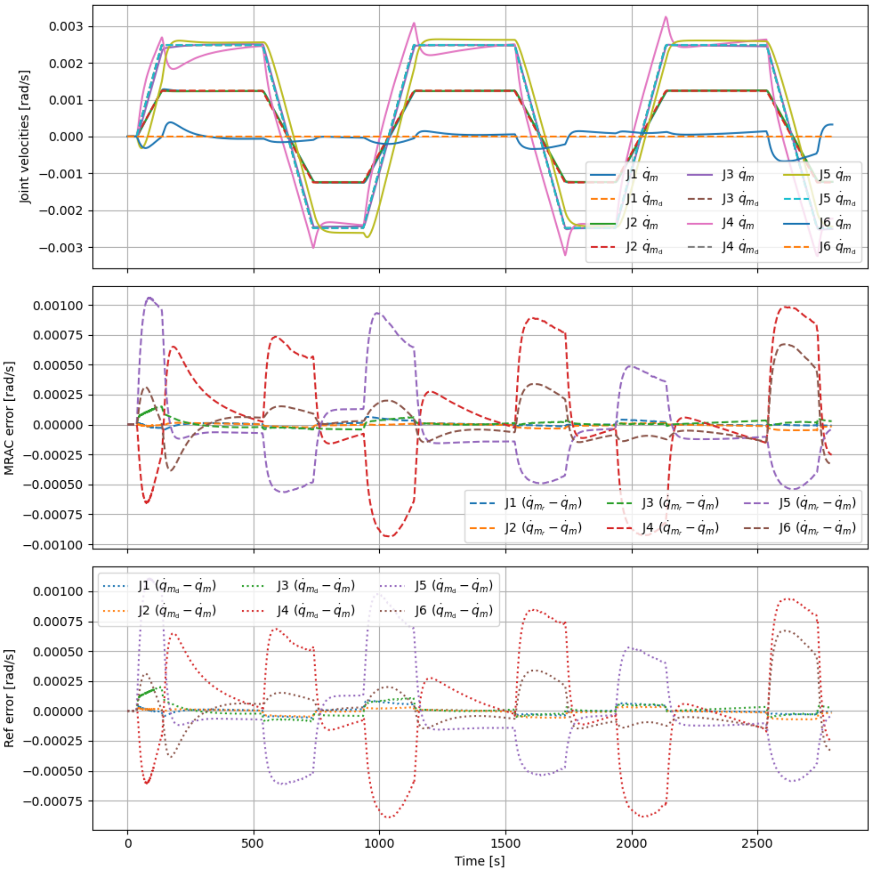

Figure 4.1: MRAC tracking performance with flexibility appendages

The obtained behavior is acceptable, but it is still the result of preliminary analyses. Additional work is needed to refine the mathematical model and improve the results.

4.4 Final conclusions

This thesis presented the development and validation of two robust control strategies for space manipulator systems operating in uncertain post-capture conditions: an NDI–NDO scheme and an NDI–MRAC approach. Both methods were thoroughly designed, implemented, and evaluated through an extensive set of simulations covering a wide range of scenarios. These included variations in target mass and inertia, external disturbances, and uncertainties in the manipulator and base dynamics. The results confirmed the effectiveness and robustness of both control architectures in compensating for the perturbations induced by the target capture and ensuring stable post-capture behavior.

Among the two, the NDI–MRAC strategy demonstrated superior performance in terms of adaptability, convergence speed, tracking accuracy, and disturbance rejection. The inclusion of a low-pass filter proved particularly beneficial, as it effectively mitigated the effects of high-frequency oscillations commonly associated with high-gain adaptive control schemes. This improvement led to smoother transient responses and enhanced overall system reliability. These findings emphasize the strong potential of adaptive control techniques in the field of space robotics, suggesting that MRAC-based methods could play a pivotal role in future Active Debris Removal (ADR) and On-Orbit Servicing (OOS) missions, where robustness and adaptability are of critical importance.

In addition to its technical contributions, this research opened several promising directions for future investigation. However, they could not be fully explored within the time frame of my internship. Pursuing these directions would further enhance the autonomy, resilience, and operational safety of space manipulator systems, bringing us closer to fully autonomous robotic servicing capabilities in orbit.

This work was conducted during an internship at ONERA – The French Aerospace Lab, whose continuous support, expertise, and scientific excellence were essential to the success of this project. The guidance and encouragement from ONERA researchers provided not only a strong foundation for tackling complex control challenges but also an inspiring research environment that fostered curiosity, creativity, and scientific rigor. I am deeply grateful for the opportunity to contribute to ONERA's ongoing efforts in advancing space robotics. Beyond the technical outcomes, this experience has been profoundly formative, offering invaluable exposure to the research process within a world-class institution and strengthening both my theoretical understanding and practical skills in advanced control systems.

Bibliography

[1] Evangelos Papadopoulos, Farhad Aghili, Ou Ma, and Roberto Lampariello. “Robotic manipulation and capture in space”. In: *Frontiers in Robotics and AI* 9 (2022), p. 849288 (cit. on p. 1).

[2] Mohammad Alizadeh and Zheng H Zhu. “A Comprehensive Survey of Space Robotic Manipulators for On-Orbit Servicing”. In: *Frontiers in Robotics and AI* 11 (2024), p. 1470950 (cit. on p. 1).

[3] Shuang Li and Yuchen She. “Recent advances in contact dynamics and post-capture control for combined spacecraft”. In: *Progress in Aerospace Sciences* 120 (2021), p. 100678 (cit. on p. 1).

[4] VV Svotina. “Space debris removal–Review of technologies and techniques. Rigid coupling between space debris and service spacecraft”. In: *Journal of Space Safety Engineering* (2024) (cit. on p. 1).

[5] Markus Wilde, Stephen Kwok Choon, Alessio Grompone, and Marcello Romano. “Equations of motion of free-floating spacecraft-manipulator systems: an engineer’s tutorial”. In: *Frontiers in Robotics and AI* 5 (2018), p. 41 (cit. on pp. 1, 5).

[6] Mathieu Rognant, Sofiane Kraiem, and Jurek Sasiadek. “Kinematic indices of rotation-floating space robots for on-orbit servicing”. In: *IFToMM World Congress on Mechanism and Machine Science*. Springer. 2019, pp. 3107–3116 (cit. on p. 1).

[7] Roberto Lampariello. “On grasping a tumbling debris object with a free-flying robot”. In: *IFAC Proceedings Volumes* 46.19 (2013), pp. 161–166 (cit. on p. 1).

[8] Farhad Aghili. “Optimal trajectories and robot control for detumbling a non-cooperative satellite”. In: *Journal of Guidance, Control, and Dynamics* 43.5 (2020), pp. 981–988 (cit. on p. 1).

[9] Kazuya Yoshida, Dimitar Dimitrov, and Hiroki Nakanishi. "On the capture of tumbling satellite by a space robot". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2006, pp. 4127–4132 (cit. on p. 1).

[10] Ria Vijayan, Marco De Stefano, and Christian Ott. "Control of an Orbital Manipulator with Reaction Wheels for On-orbit Servicing". In: *IFAC-PapersOnLine* 55.38 (2022), pp. 26–32 (cit. on pp. 1, 2).

[11] Borna Monazzah Moghaddam and Robin Chhabra. "On the guidance, navigation and control of in-orbit space robotic missions: A survey and prospective vision". In: *Acta Astronautica* 184 (2021), pp. 70–100 (cit. on p. 1).

[12] Pablo Colmenarejo et al. "Results of the comrade project: combined control for robotic space-craft and manipulator in servicing missions: active debris removal and re-fuelling". In: *11th International ESA Conference on Guidance, Navigation & Control Systems*. 2020, pp. 10–23 (cit. on p. 1).

[13] Sofiane Kraıem, Mathieu Rognant, Jean-Marc Biannic, and Yves Brière. "Dynamics and robust control of a space manipulator with flexible appendages for on-orbit servicing". In: *CEAS Space Journal* 15.5 (2023), pp. 681–700 (cit. on pp. 2, 19, 91).

[14] Omer Burak Iskender, NG Wee Meng, Keck-Voon Ling, Luca Simonini, Vincent Dubanchet, and Jose Maria. "Detumbling of small satellites with model predictive control". In: *70the International Astronautical Congress, Washington D.C., United States, 21-25 October 2019.* (2019) (cit. on p. 2).

[15] Iskender Omer Burak. "Model predictive control for spacecraft rendezvous and docking with uncooperative targets". PhD thesis. Ph. D. dissertation, Nanyang Technological University, Singapore, 2020 . . ., 2020 (cit. on p. 2).

[16] Caroline Buckner and Roberto Lampariello. "Tube-based model predictive control for the approach maneuver of a spacecraft to a free-tumbling target satellite". In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018, pp. 5690–5697 (cit. on p. 2).

[17] Dan Zhang and Bin Wei. "A review on model reference adaptive control of robotic manipulators". In: *Annual Reviews in Control* 43 (2017), pp. 188–198 (cit. on p. 2).

[18] Tansel Yucelen. "Model reference adaptive control". In: *Wiley encyclopedia of electrical and electronics engineering* (2019), pp. 1–13 (cit. on pp. 2, 32, 34, 35).

[19] K Merve Dogan, Tansel Yucelen, Benjamin C Gruenwald, and Jonathan A Muse. “On model reference adaptive control for uncertain dynamical systems with unmodeled dynamics”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 377–382 (cit. on p. 2).

[20] Jesse Jaramillo, Tansel Yucelen, and Kevin Wilcher. “Scalability in model reference adaptive control”. In: *AIAA Scitech 2020 Forum*. 2020, p. 1119 (cit. on pp. 2, 57, 64, 65).

[21] Anton I Glushchenko et al. “I-DREM MRAC with time-varying adaptation rate & no a priori knowledge of control input matrix sign to relax PE condition”. In: *2021 European Control Conference (ECC)*. IEEE. 2021, pp. 2175–2180 (cit. on p. 2).

[22] Jesse Jaramillo and Tansel Yucelen. “Model reference adaptive control for time-varying command following: A gradient descent approach”. In: *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE. 2024, pp. 4297–4302 (cit. on p. 2).

[23] Ehsan Arabi, Benjamin C Gruenwald, Tansel Yucelen, and Nhan T Nguyen. “A set-theoretic model reference adaptive control architecture for disturbance rejection and uncertainty suppression with strict performance guarantees”. In: *International Journal of Control* 91.5 (2018), pp. 1195–1208 (cit. on pp. 3, 90).

[24] Tansel Yucelen and Wassim M Haddad. “Low-frequency learning and fast adaptation in model reference adaptive control”. In: *IEEE Transactions on Automatic Control* 58.4 (2012), pp. 1080–1085 (cit. on pp. 3, 79, 80).

[25] Alireza Mohammadi, Mahdi Tavakoli, Horacio J Marquez, and Farzad Hashemzadeh. “Nonlinear disturbance observer design for robotic manipulators”. In: *Control Engineering Practice* 21.3 (2013), pp. 253–267 (cit. on p. 19).

[26] Simon P Schatz, Tansel Yucelen, Benjamin C Gruenwald, and Florian Holzapfel. “Application of a novel scalability notion in adaptive control to various adaptive control frameworks”. In: *AIAA Guidance, Navigation, and Control Conference*. 2015, p. 0608 (cit. on p. 55).

[27] Anton I Glushchenko, Konstantin A Lastochkin, and Vladislav A Petrov. “Normalization of regressor excitation in the dynamic extension and mixing procedure”. In: *Automation and Remote Control* 83.1 (2022), pp. 17–31 (cit. on p. 55).

[28] Jiwei Wang, Simone Baldi, and Henk J van Waarde. “Bridging Model Reference Adaptive Control and Data Informativity”. In: *arXiv preprint arXiv:2502.21091* (2025) (cit. on p. 56).

[29] Stanislav Aranovskiy, Alexey Bobtsov, Romeo Ortega, and Anton Pyrkin. "Parameters estimation via dynamic regressor extension and mixing". In: *2016 American Control Conference (ACC)*. IEEE. 2016, pp. 6971–6976 (cit. on p. 56).

[30] Francesco Sena, Hrishik Mishra, Ria Vijayan, and Marco De Stefano. *A Control Strategy for an Orbital Manipulator Equipped with an External Actuator at the End-Effector*. 2025.

[31] Sofiane Kraıem, Mathieu Rognant, and Sérgio Waitman. "Simulation platform to design and validate control laws for a space manipulator system performing on-orbit servicing". In: *2024 International Conference on Space Robotics (iSpaRo)*. IEEE. 2024, pp. 98–103.

[32] Stephen Boyd et al. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[33] Flores-Abad et al. "A review of space robotics technologies for on-orbit servicing". In: *Progress in Aerospace Sciences* 68 (2014).

[34] Christian Ott, Alexander Dietrich, and Alin Albu-Schäffer. "Prioritized multi-task compliance control of redundant manipulators". In: *Automatica* 53 (2015), pp. 416–423.

[35] Mark Shilton, Martin Garland, Chris Hackett, Elie Allouis, Matt Lisle, Connor Meringolo, Davide Paganini, and Alexander Hall. "VISPA URDF". In: (2024). URL: `https://github.com/AirbusDefenceAndSpace/vispa`.

[36] Wendel K Belvin, William R Doggett, Judith J Watson, John T Dorsey, Jay E Warren, Thomas C Jones, Erik E Komendera, Troy Mann, and Lynn M Bowman. "In-space structural assembly: Applications and technology". In: *3rd AIAA Spacecraft Structures Conference*. 2016, p. 2163.