



**Politecnico
di Torino**

Politecnico di Torino

**Mechatronic Engineering – Control Technologies for
Industry 4.0**

Master's Thesis

**Control methodology
for an inflatable robotic arm**

Supervisors

Prof. Stefano Mauro

Ing. Pierpaolo Palmieri

Ing. Matteo Gaidano

Candidate

Nicola Occhipinti

289648

December 2022

Abstract

Soft robotic arms, made up of flexible components, have emerged recently. These systems can improve safety in collaborative operations, however, they also suffer from decreased precision and low payload. The control problem is then challenging due to link flexibility.

The examined system is the so-called POP-UP robot, which is an anthropomorphic robotic arm with two inflatable links and three rigid joints designed and prototyped at DIMEAS. The system is composed by inflatable links and it is characterized by non-linearity. This structure is designed for aerospace applications and its main advantage regards the reduction of costs and storage, during the launch phase, with respect to a rigid equivalent manipulator and it is hard to derive a precise dynamic model suitable for control application.

As a consequence it is very difficult to find a robust control law aiming to reduce the oscillations that arise after grasping a payload.

This thesis aims to describe some methods to model an inflatable link and implement a vibration dumping control. A pseudo-rigid body model is defined and identified according to a set of experimental data. The same data are used to tune a FE model of the system.

The frequency response of the flexible link to input from the motoreducer is evaluated, considering different payloads.

This model is eventually used to develop a vibration control to the robot motion control. The obtained results are shown and discussed.

Contents

Introduction	1
Chapter 1- POPUP robot	6
Chapter 2- Modelling of inflatable link	6
2.1-Rigid link	6
2.1.1-Denavit-Hartenberg convention	7
2.1.2-Kinematics	13
2.1.3-Direct Kinematics	13
2.1.4-Inverse kinematics	15
2.1.1-Differential kinematics.....	19
2.2-Pseudo-rigid body model	21
2.2.1-Denavit-Hartenberg convention 7 dof.....	24
2.2.2-Link structure in Simscape modelled with PRBM	25
2.3-Finite element method (FEM).....	21
Chapter 3-Set membership identification	30
3.1-A priori information on the system	31
3.2-A priori information on the noise.....	33
3.3-Collected data	34
Chapter 4-Control system for robotic arms	37
4.1-Control in the operational space.....	37
4.2-Control in the joint space	37
4.2.1-Decentralized control	38
Chapter 5-Control system for POPUP	40
5.1-Decentralized control for one link – one motoreducer system	40
5.1.1-Vibration damping control for one link – one motoreducer system	44
5.2-Simulations	47
5.2.1-One link – one motoreducer system simulations.....	47
Chapter 6-Frequency analysis	48
6.1-Frequency analysis of the motoreducer	48

6.2-Frequency analysis of the link.....	53
Chapter 7-Vibration damping control system with a Low-Pass filter.....	59
7.1-Vibration damping control applied to the simplified system	60
7.1.1-Simulations without gravity	61
7.1.2-Simulations with gravity	68
7.2-Vibration damping control applied to the entire system	76
7.2.1-Simulations without gravity	77
7.2.2-Simulations with gravity	88
Chapter 8-Differential kinematic control with vibration damping control	94
8.1-Simulations	95
Conclusions.....	110
Appendix.....	111
Bibliography and webography	156

Introduction

The POPUP robot is an inflatable robotic anthropomorphic arm, which belongs to the class of soft and flexible robots. This kind of robots are becoming widespread in several fields such as industrial, medical, aerospace, etc. Soft robots can be classified into continuum [1] or articulated [2].

Soft robots are made of compliant and flexible or extensible materials like silicon rubbers. The choice of the material allows the structure to deform or absorb much of the energy arising from a collision [3]. This is the main advantage since soft robots are safer for humans. Nevertheless, there exist applications where a rigid robot is better than a soft one. For example, in a factory where manipulators are used to assemble something repeatedly in a building chain. In this case, the rigid robot is more reliable because it guarantees a certain precision, accuracy and high speed in the execution of the task since the environment is plentiful known and the structure is adapted to work in this context. Instead, if a robot is designed to accomplish tasks in a plenty or partially unknown environment, using soft robots can be an advantage due to increased safety. Soft robots are designed to interact with the environment and with the humans. “Better bend than break” says an old Scottish proverb. The idea of a soft robot was inspired by biological systems. Figure 1 shows a robotic octopus built by the University of Pisa with the aim of cleaning contaminated oceans from pollution and debris, or for delicate surgeries using a very small prototype [4].

Introduction



Figure 1 Robotic octopus

Another example of soft robot is the one built by Harvard University. This is an indestructible quadrupedal soft robot with the aim to save human lives in extreme environments. This robot doesn't need to be connected to any power source and it is made from a composite silicone rubber. For this reason, it is an extremely resilient robot that can handle extreme conditions, from fire and even a car running it over [5].

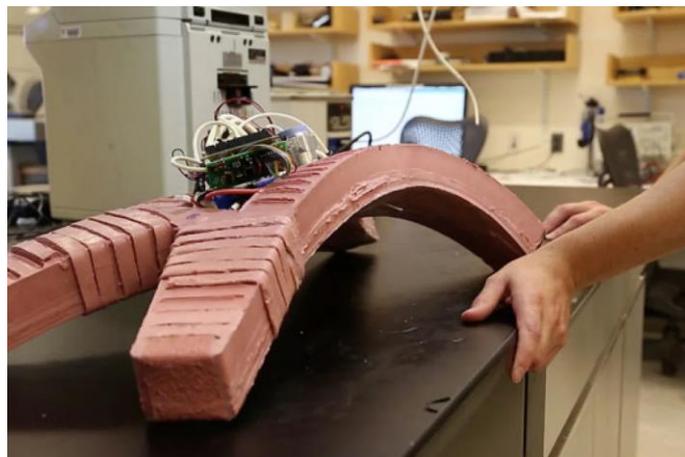


Figure 2 Indestructible robot

Other applications can be found in the medical field since soft robots have high performance potential where the kinematics of the system are not defined in

Introduction

advance, such as interacting with objects of unknown shape and size or coming into physical contact with living cell and human bodies. So, as a result of the latter, kinds of robots are used for rehabilitation applications, as the prototype of Figure 3, and clinical applications like in surgeries and endoscopies [6]. An example of soft robot used for endoscopies is shown in Figure 4.



Figure 3 Example of soft robot for rehabilitation



Figure 4 soft robot for endoscopy

Aerospace companies have decided to develop inflatable structures since these new structures bring several advantages such as:

- Low storage volume;
- Lightweight;
- Efficient packaging;

Introduction

- Cost reduction during the launch.

There are many applications in the aerospace field for these innovative structures. For example, NASA developed in 2014 an inflatable solar array [7] and this is shown in Figure 5. This structure is used for providing power for CubeSat, a very small satellite, or it can be used for human exploration in deep space.

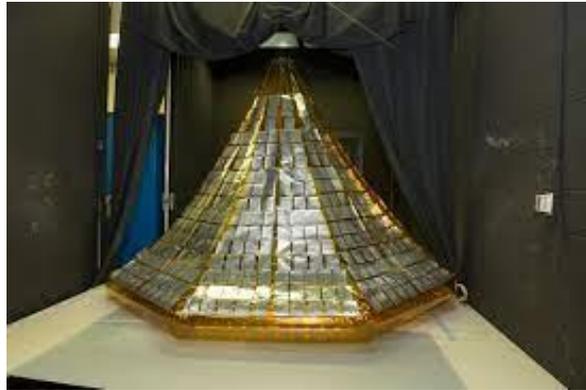


Figure 5 Inflatable solar array

Another example of inflatable structure can be found in the inflatable module designed for the International Space Station. This prototype is called BEAM (Bigelow Expandable Activity Module) [8]. This structure is depicted in Figure 6.

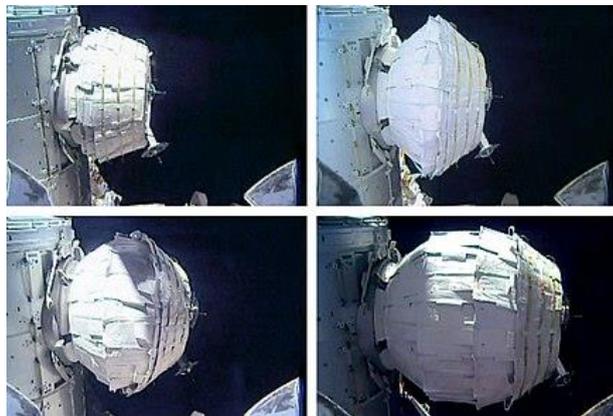


Figure 6 BEAM

Furthermore, Pneumocell studied a concept of inflatable Moon base, as depicted in Figure 7. These structures are designed for protecting astronauts from radiations and micrometeorites and consist of semi-buried inflatable habitats sited in regions

Introduction

where there is solar illumination, whilst the mirror system reflect the light inside the habitats [9].

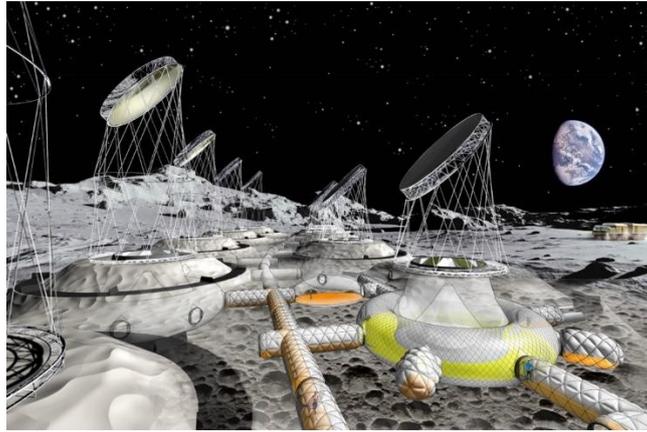


Figure 7 Inflatable Moon base

Other relevant applications in this field are inflatable wings [10], inflatable antennas [11] and inflatable aerodynamics decelerators [12].

Inflatable robots can be used in space application. As an example *King Louie robot* has been developed by Brigham Young University in collaboration with NASA. It is a humanoid robot with pneumatic actuation, as shown in Figure 8. Specifically, its arms are made up of pneumatic chambers (artificial muscles) whose functioning corresponds to the human muscles [13]. *Louie* will be useful in space missions as it is less bulky with respect to a droid made up of iron.



Figure 8 King Louie

Chapter 1

POPUP robot

The POPUP robot, shown in figure 9, is an anthropomorphic robotic arm, designed for aerospace applications, with two inflatable links and three rigid joints that are actuated by electric motors. In addition, a spherical wrist is going to be added to the structure in such a way to reach 6 degrees of freedom. This robotic arm is designed both for onboard applications aiming to collaborate with astronauts [14] and for open space missions [15]. The two structures will have different size, the first one is smaller with respect to the second and the one designed for open space mission will have similar dimensions of the Canadarm 3, shown in Figure 10, and it is needed for debris capture.

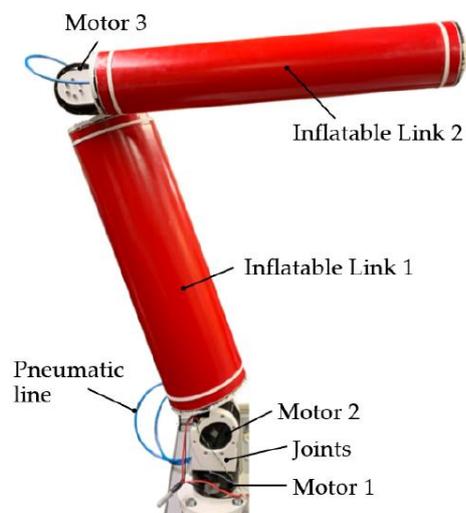


Figure 9 POPUP robot

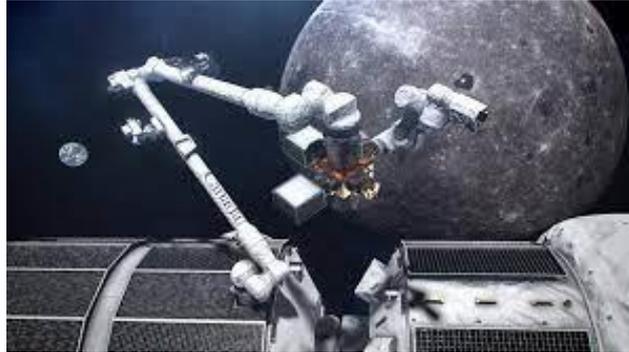


Figure 10 An artist's concept of Canadarm3

The POPUP robot is characterized by many advantages, such as:

- Low storage volume;
- It can be stored in a small package;
- Cost reduction during the launch phase;
- Increasing in safety for onboard applications;
- It can be deployed when required.

However, many issues are present, such as:

- It results complex to derive a precise dynamic model;
- Difficulty in the control due to non-linearities derived from deformations occurring along links;
- Low payload capacity;
- Low stiffness.

The equivalent rigid robotic arm without wrist has 3 degrees of freedom while the inflatable one can reach a greater number due to a low stiffness.

The robot could be connected to a pressure tank for the inflation phase through a valve that is switched off when the desired pressure value is reached since the pressure in this project isn't an actuation variable [16].

In Figure 11 the pneumatic line for the robot designed to be sent in space is shown. It is made up of:

- **Pressurized tank:** it can admit pressures up to of 30 MPa;
- **Reducing valve:** these valves are able to set the right values of pressure at the links;
- **Digital on-off valves:** these are ON only during inflation or deflation phases, whilst are OFF when the desired pressure is reached.

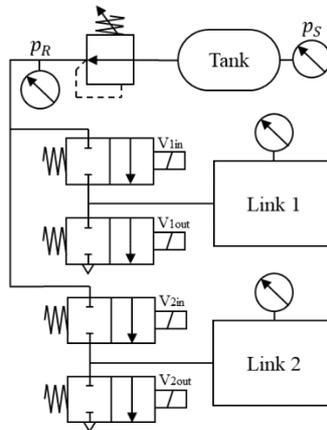


Figure 11 Pneumatic line

The POPUP could be sent in space in a very small package by following this deployment strategy: at the beginning it is deflated and the two links are wrapped around motor 2 and 3. Then, during the deployment phase, the two links are unwrapped and the inflation starts from the second one, as depicted in Figure 12. Then, during the withdrawal phase, the procedures are both the same however the links are deflated and wrapped.

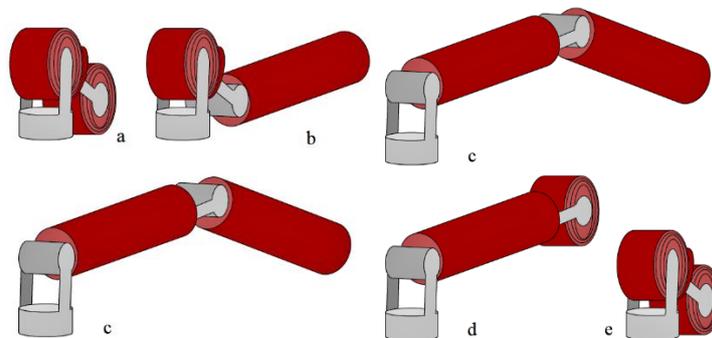


Figure 12 Deployment (on the left) and withdrawal (on the right)

Furthermore, a prototype for laboratory testing is built in order to study the control strategy. In this case, the presence of gravity affects the performance. The prototype for laboratory testing has the following characteristics: both links have a length $L = 600 \text{ mm}$, link 1 has radius $r_1 = 85 \text{ mm}$ and link 2 has radius $r_2 = 55 \text{ mm}$.

For what regards the sensors implemented in the POPUP prototype, these are:

- **Encoders:** this is a sensor incorporated inside the motoreducer and this provides angular position and angular velocity in feedback.
- **IMU:** Inertial Measurement Unit is an electronic device that measures the orientation of the body exploiting accelerometers, gyroscopes and sometimes magnetometers.



Figure 13 IMU sensor

- **Pressure sensor:** this sensor, as the name implies, is used to measure the pressure inside the links. This device exploits a membrane that is fitted with an electrical resistance which depends on the applied pressure.



Figure 14 Pressure sensor

Chapter 2

Modelling of inflatable link

There are several ways to model the inflatable behaviour of the link. In this thesis two models are discussed.

The methods are:

- 1) **Lumped parameter approach:** this method models the flexible links by some virtual joints carrying lumped masses. A linear torsional spring is considered on the virtual joints to model the flexibility. The mass model of such models is supposed to be diagonal or semi-diagonal [17].
- 2) **Finite element method (FEM):** this method consists of discretizing a continuum material into a finite number of parts. This procedure is used to solve partial differential equations in the space and it allows to calculate approximate solutions. The FEM is based on the concept of virtual works. This method is based on Euler-Bernoulli theory.

2.1 Rigid link

The inflatable links could be represented by means of the equivalent structure of a rigid robot. In this way the angular deformation occurring along the links are neglected. Every joint has its own coordinate system and it is convenient to follow the Denavit-Hartenberg convention in order to fix the reference frames. This

approximation can be used in simulation to derive the direct, inverse and differential kinematics.

2.1.1 Denavit-Hartenberg convention

This is a systematic procedure that simplifies the kinematics analysis of manipulators. We have to set four parameters:

- a_i is the link length;
- α_i the link twist;
- d_i is the link offset;
- ϑ_i is the joint angle.

After setting these parameters we can easily write the homogeneous transformation matrix A_i^{i-1} . Three of these parameters are constant while only one is variable, namely the joint variables. The joint variable changes between prismatic and revolute joint. In our case, since we only have revolute joints, our joint variable is the joint angle. The figure below shows the Denavit-Hartenberg parameters.

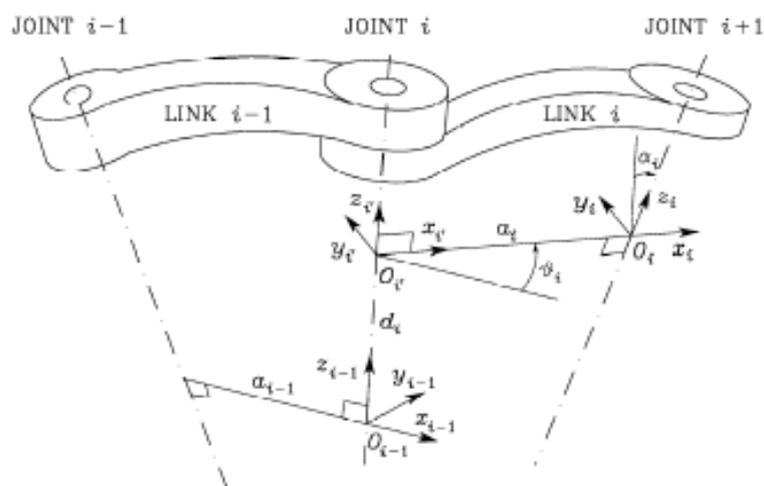


Figure 15 Denavit-Hartenberg convention

Following the DH convention, we obtain the local reference frames oriented in the following way.

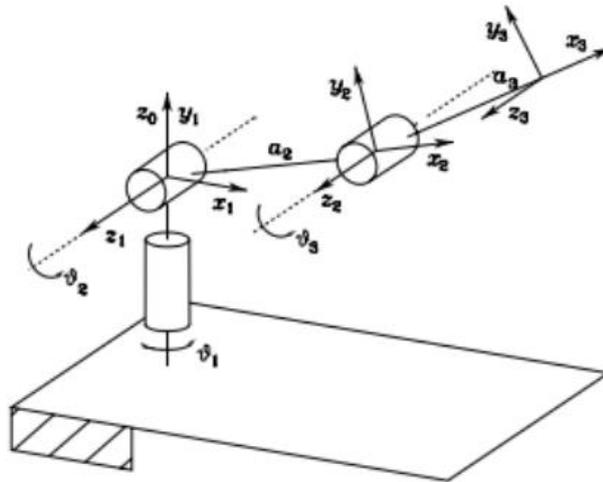


Figure 16 Denavit-Hartenberg convention applied to a 3 dof anthropomorphic robotic arm

After setting the Denavit-Hartenberg parameters it becomes very simple to obtain the transformation matrix between one local frame and the successive one exploiting the homogeneous transformation matrix 2.1.

$$A_i^{i-1}(q_i) = A_i^{i'} A_i^{i'} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

The DH parameters are written in the table below.

Link	θ_i	d_i	a_i	α_i
0	θ_1	0.15	0	$\pi/2$
1	θ_2	0.05	0.745	0
2	θ_3	-0.12	0.685	0

Table 1 Denavit-Hartenberg parameters

After setting these parameters we can exploit the robotics toolbox to build the robot structure and to study forward, inverse and differential kinematics. The equivalent rigid arm built in MATLAB is shown in the Figure 17.

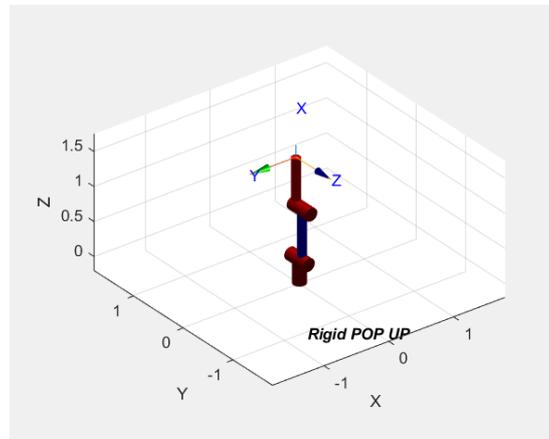


Figure 17 Equivalent rigid arm

2.1.2 Kinematics

Robot kinematics is the geometrical study of the arm motion with respect to a reference frame that could be centred on the base or could be referred to another point in the space. Let's study the robot kinematics by fixing the reference frame in the robot base.

2.1.3 Direct Kinematics

Direct kinematics is based on deriving the homogeneous transformation matrix that, given the joint variables, is able to provide the end effector position [18]. Since we

have three motorized links, the number of degrees of freedom is equal to three. This analysis exploits the Denavit-Hartenberg convention.

$$T_3^0(\theta_1\theta_2\theta_3) = A_1^0(\theta_1)A_2^1(\theta_2)A_3^2(\theta_3) \quad (2.2)$$

Exploiting the transformation matrix written above (2.1) we can write:

$$A_1^0 = \begin{bmatrix} c_{\theta_1} & 0 & s_{\theta_1} & 0 \\ s_{\theta_1} & 0 & -c_{\theta_1} & 0 \\ 0 & 1 & 0 & 0.15 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$A_2^1 = \begin{bmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & 0.745c_{\theta_2} \\ s_{\theta_2} & c_{\theta_2} & 0 & 0.745s_{\theta_2} \\ 0 & 0 & 1 & 0.05 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$A_3^2 = \begin{bmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & 0.685c_{\theta_3} \\ s_{\theta_3} & c_{\theta_3} & 0 & 0.685s_{\theta_3} \\ 0 & 0 & 1 & -0.12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$T_3^0 = \begin{bmatrix} c_1c_2c_3 - c_1s_2s_3 & -c_1c_2s_3 - c_1s_2c_3 & s_1 & 0.745(c_1c_2) - 0.07s_1 + 0.685(c_1c_2c_3) - 0.685(c_1s_2s_3) \\ s_1c_2c_3 - s_1s_2s_3 & -c_2s_1s_3 - c_3s_1s_2 & -c_1 & 0.745(s_1c_2) + 0.07c_1 - 0.685(s_1s_2s_3) + 0.685(c_2c_3s_1) \\ c_2s_3 - s_2c_3 & c_2c_3 - s_2s_3 & 0 & 0.745(s_2) + 0.685(c_2s_3) + 0.685(s_2c_3) + 0.15 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Where c_i is the cosine of the i -th joint angle and s_i is the sine of the i -th joint angle.

We can simplify the matrix exploiting the relationships between cosine and sine of the sum and difference between angles, knowing that:

$$\bullet \sin(\alpha \pm \beta) = \sin(\alpha)\cos(\beta) \pm \sin(\beta)\cos(\alpha) \quad (2.7)$$

$$\bullet \cos(\alpha \pm \beta) = \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta) \quad (2.8)$$

We can rewrite the homogenous matrix:

$$T_3^0 = \begin{bmatrix} c_1(c_{2+3}) & -c_1(s_{2+3}) & s_1 & 0.745(c_1c_2) - 0.07s_1 + 0.685(c_1(c_{2+3})) \\ s_1(c_{2+3}) & -s_1(s_{2+3}) & -c_1 & 0.745(s_1c_2) + 0.07c_1 + 0.685s_1(c_{2+3}) \\ s_{2+3} & c_{2+3} & 0 & 0.745(s_2) + 0.685(s_{2+3}) + 0.15 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

Where:

$$\bullet \quad c_{i+j} = \cos(q_i + q_j) \quad (2.10)$$

$$\bullet \quad s_{i+j} = \sin(q_i + q_j) \quad (2.11)$$

So, the last column of the homogenous matrix gives us the relationship between joint variables and end effector position, so we can write:

$$\bullet \quad p_x = 0.745(c_1c_2) - 0.07s_1 + 0.685(c_1(c_{2+3})) \quad (2.12)$$

$$\bullet \quad p_y = 0.745(s_1c_2) + 0.07c_1 + 0.685(s_1(c_{2+3})) \quad (2.13)$$

$$\bullet \quad p_z = 0.745(s_2) + 0.685(s_{2+3}) + 0.15 \quad (2.14)$$

2.1.4 Inverse kinematics

The inverse kinematics consists in computing the joint variable values starting from the position of the end effector. This is fundamental because all motion specifications are expressed in the workspace and thanks to the inverse kinematics it is possible to express these specifications in the joint space [18]. The problem is that, the direct kinematics is univocal while the inverse kinematics admits multiple solutions and in some cases it is not possible to obtain an analytical solution because the equations are non-linear.

The existence of solutions is guaranteed only if the end effector pose belongs to the workspace, this is the region described by the origin of the end-effector frame where all the manipulator joints execute all possible motions.

Let $\mathbf{p} = [p_x \ p_y \ p_z]$ be the position of the end point of the second link with respect to the base frame, where thanks to the direct kinematics we can write:

$$\bullet \quad p_x = 0.745(c_1c_2) - 0.07s_1 + 0.685(c_1(c_{2+3})) \quad (2.16)$$

$$\bullet \quad p_y = 0.745(s_1c_2) + 0.07c_1 + 0.685(s_1(c_{2+3})) \quad (2.17)$$

$$\bullet \quad p_z = 0.745(s_2) + 0.685(s_{2+3}) + 0.15 \quad (2.18)$$

Now, we can square and sum the three equation and we obtain:

$$\bullet \quad p_x^2 = 0.555c_1^2c_2^2 + 0.0049s_1^2 + 0.47c_1^2(c_{2+3})^2 - 0.1043c_1c_2s_1 + 1.02065c_1^2c_2(c_{2+3}) - 0.0959s_1c_1(c_{2+3}) \quad (2.19)$$

$$\bullet \quad p_y^2 = 0.555s_1^2c_2^2 + 0.0049c_1^2 + 0.47s_1^2(c_{2+3})^2 + 0.1043c_1c_2s_1 + 1.02065s_1^2c_2(c_{2+3}) + 0.0959s_1c_1(c_{2+3}) \quad (2.20)$$

$$\bullet \quad p_z^2 = 0.555s_2^2 + 0.47(s_{2+3})^2 + 0.0225 + 1.02065s_2(s_{2+3}) + 0.2235s_2 + 0.2055(s_{2+3}) \quad (2.21)$$

$$p_x^2 + p_y^2 + p_z^2 = 1.0965 + 1.02065c_3 + 0.2235s_2 + 0.2055(s_{2+3}) \quad (2.22)$$

With this system, it's very difficult to analytically obtain an exact inversion of the kinematics chain. So, we can simplify the system by setting the offset d_i equal to zero in such a way to obtain a model equal to the one discussed by Siciliano in [18].

The simplified model has the same Denavit-Hartenberg parameters as before except for the d_i .

Link	θ_i	d_i	a_i	α_i
0	θ_1	0	0	$\pi/2$
1	θ_2	0	0.745	0
2	θ_3	0	0.685	0

Table 2 Denavit-Hartenberg parameters without offset

- $p_x = 0.745(c_1c_2) + 0.685(c_1(c_{2+3})) = c_1(0.745(c_2) + 0.685(c_{2+3}))$
(2.23)

- $p_y = 0.745(s_1c_2) + 0.685(s_1(c_{2+3})) = s_1(0.745(c_2) + 0.685(c_{2+3}))$
(2.24)

- $p_z = 0.745(s_2) + 0.685(s_{2+3})$ (2.25)

Now, we can square and sum the three equations and we obtain:

- $p_x^2 = c_1^2(0.555c_2^2 + 0.47(c_{2+3})^2 + 1.02065c_2(c_{2+3}))$ (2.26)

- $p_y^2 = s_1^2(0.555c_2^2 + 0.47(c_{2+3})^2 + 1.02065c_2(c_{2+3}))$ (2.27)

- $p_z^2 = 0.555s_2^2 + 0.47(s_{2+3})^2 + 1.02065s_2(s_{2+3})$ (2.28)

$$p_x^2 + p_y^2 + p_z^2 = 1.024 + 1.02065c_3 \quad (2.29)$$

So,

$$c_3 = \frac{p_x^2 + p_y^2 + p_z^2 - 1.024}{1.02065} \quad (2.30)$$

The admissibility of the solution requires that $-1 \leq c_3 \leq 1$. From trigonometric relationship we can compute the sine of the third joint angle as follows:

$$s_3 = \pm\sqrt{1 - c_3^2} \quad (2.31)$$

And thus

$$\theta_3 = \text{Atan2}(s_3, c_3) \quad (2.32)$$

The last equation isn't univocal because it admits two solutions. This is due to the arctangent function. The first solution comes out if the angle belongs to $[-\pi, \pi]$,

instead the second one if the angle lays in the other two quadrants and it is equal to the opposite of the first angle.

After having determined θ_3 it is possible to compute the other two joint angles, we consider

$$p_x^2 + p_y^2 = (0.745(c_2) + 0.685(c_{2+3}))^2 \quad (2.33)$$

Now we take the square root

$$\sqrt{p_x^2 + p_y^2} = 0.745(c_2) + 0.685(c_2 c_3 - s_2 s_3) \quad (2.34)$$

$$c_2 = \frac{\pm \sqrt{p_x^2 + p_y^2} (0.745 + 0.685 c_3) + p_z 0.685 s_3}{0.555 + 0.469 + 1.02 c_3} \quad (2.35)$$

For the sine, we take the expression of p_z

$$p_z = 0.745(s_2) + 0.685(s_2 c_3 - c_2 s_3) = s_2(0.745 + 0.685 c_3) - 0.685 c_2 s_3 \quad (2.36)$$

$$\theta_2 = \text{Atan2}(s_2, c_2) \quad (2.37)$$

Since both sine and cosine of the second joint angle can assume two different values, this implies that the second joint angle can assume four different values.

For the first joint angle we start from the expression of p_x (2.23) and p_y (2.24) and the auxiliary expression obtained by squaring and summing the first two equations (2.33). We obtain

$$p_x = \pm c_1 \sqrt{p_x^2 + p_y^2} \quad (2.38)$$

$$p_y = \pm s_1 \sqrt{p_x^2 + p_y^2} \quad (2.39)$$

We can easily conclude that there are two possible solutions:

$$\theta_{11} = \text{Atan2}(p_y, p_x) \quad (2.40)$$

$$\theta_{12} = \text{Atan2}(-p_y, -p_x) \quad (2.41)$$

Overall, there exist four different configurations that express the same end effector position (in our case the final position of the second link).

The solution can be determined only if $p_x \neq 0$ and $p_y \neq 0$ otherwise there exist an infinite number of solutions where the value of the second and third joint angle don't depend on the value of the first one, which brings it to the singularity case.

2.1.5 Differential kinematics

The differential kinematics computes the relationship between the joint velocities and the corresponding end effector velocities both linear and angular. It is based on the Jacobian matrix that is different for every configuration.

$$J = \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} \quad (2.42)$$

Where z_{i-1} is the third column of the rotation matrix R_{i-1}^0 , $z_0 = [0; 0; 1]$ and p_e is given by the first three elements of the last column of the complete homogeneous transformation matrix.

The computations of the Jacobian are done in [7].

$$J = \begin{bmatrix} -s_1(0.745c_2 + 0.685c_{2+3}) & -c_1(0.745s_2 + 0.685s_{2+3}) & -0.685c_1s_{2+3} \\ c_1(0.745c_2 + 0.685c_{2+3}) & -s_1(0.745s_2 + 0.685s_{2+3}) & -0.685s_1s_{2+3} \\ 0 & 0.745c_2 + 0.685c_{2+3} & 0.685c_{2+3} \\ 0 & s_1 & s_1 \\ 0 & -c_1 & -c_1 \\ 1 & 0 & 0 \end{bmatrix} \quad (2.43)$$

The relationship between joint velocities and the corresponding end effector velocities is the following one:

$$v_e = J(q)\dot{q} \quad (2.44)$$

Since the last three rows are linearly dependent, we can consider only the firsts three. So, let's consider the 3x3 matrix characterized by the first three rows and let's compute the determinant of this matrix in order to determine the singularities.

$$\det(J) = -0.581s_3(0.745c_2 + 0.685c_{2+3}) \quad (2.45)$$

There are three cases of singularities:

If $s_3 = 0$ we obtain the first two singularities:

- $\theta_3 = 0$

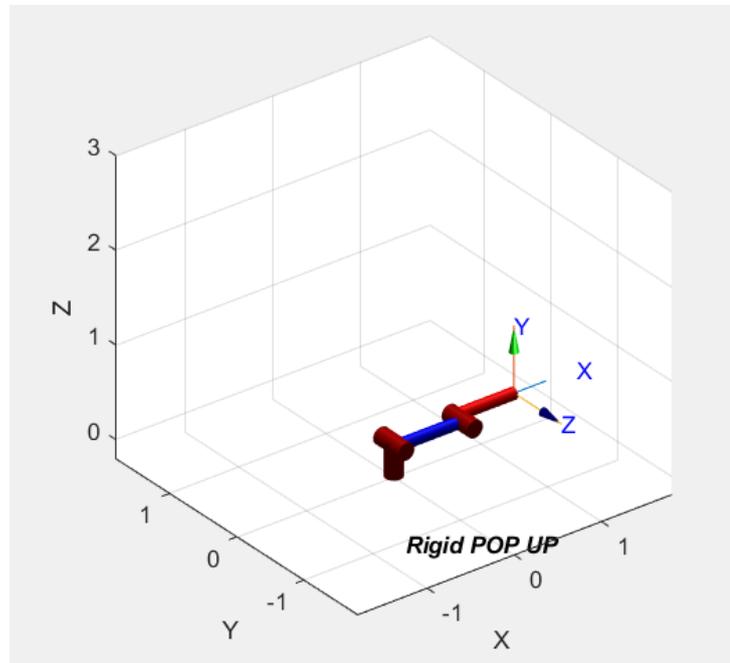


Figure 18 First elbow singularity

- $\theta_3 = \pi$

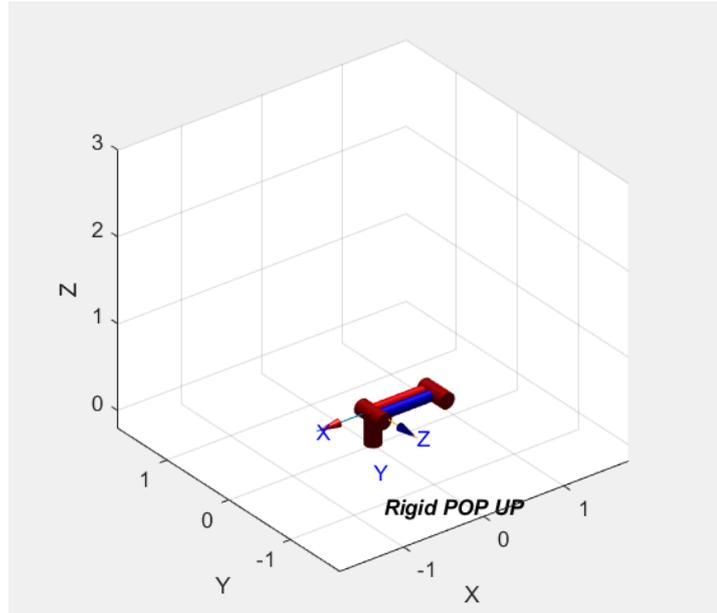


Figure 19 Second elbow singularity

- The last case of singularity occurs when $(0.745c_2 + 0.685c_{2+3}) = 0$
(2.46)

The first two cases are the elbow singularities while the last one is the shoulder singularity.

2.2 Pseudo-rigid body model

Exploiting the Lumped-parameter approach it is possible to derive a pseudo-rigid body model (PRBM). In this model every link is considered as two rigid bodies with length l_1 and l_2 so that their sum is equal to the total length of the link. The two parts of the link are connected by a hinge with torsional spring and a damper, as shown in Figure 21. The damper is present since it accounts for the velocity of the deformation occurring along the link in such a way to have a more precise dynamics. In Figure 21, θ is the angular deflection, v is the extremity displacement

that represents the displacement of the link extremity position between the POPUP or every flexible link manipulator and the equivalent rigid one.

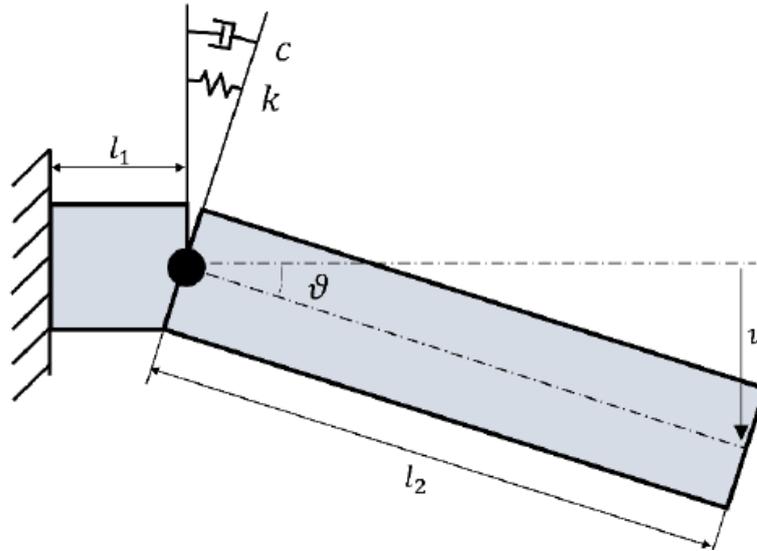


Figure 20 Link modelled as a PRBM

The model seen until now is a planar model, however it is possible to extend this model to three dimensions considering the deflection and displacement occurring along two orthogonal axes, so we have two angular deflections along x and y axes. The same holds for the extremity displacement because in a three-dimensional space, we have the extremity displacement along two axes.

The link motion can be described by the following equation:

$$I\ddot{\theta} + c(p)\dot{\theta} + k(p)\theta = \tau \quad (2.47)$$

Where:

- I is the moment of inertia;
- p is the internal relative pressure;
- τ is the external torque;
- c is the damping coefficient and it depends on the pressure;
- k is the stiffness coefficient and it depends on the pressure.

It is possible to define the wrinkling moment, it represents the bending load for which the first wrinkles appear on the link and theoretically it is the half of the collapse moment that can be defined as the bending load for which an increase in deflection does not correspond to an increase in reaction force [17] and it can be computed in the following way:

$$M_c = \frac{\pi}{4} \pi p r^3 \quad (2.48)$$

The stiffness characteristic below the theoretical wrinkling moment can be approximated as a linear characteristic as shown in the Figure 21 [16]. For this reason, in order to obtain a controllable system it is required to work in a condition behind the wrinkling moment.

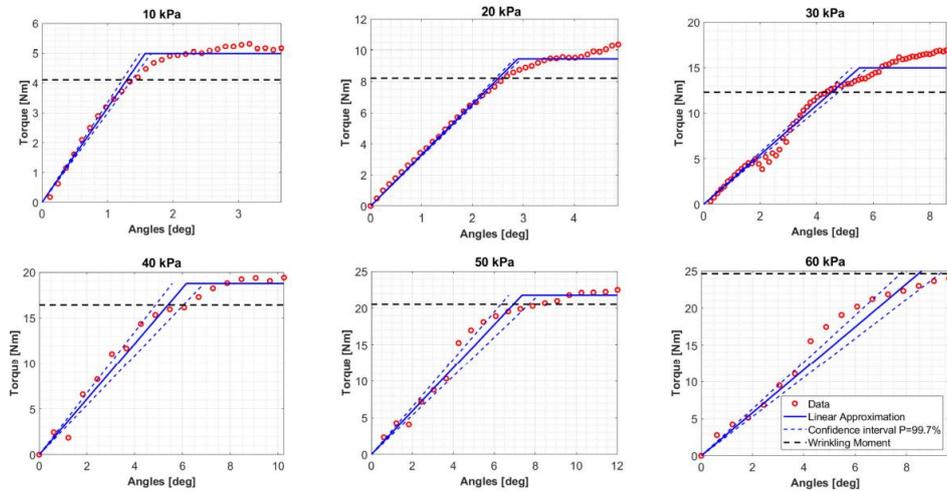


Figure 21 Stiffness characteristic of inflatable link varying pressure

The experimental test validates the theoretically wrinkling moment as a limit value. So, working without overpass this value, the geometric dimensions such as radius and length but also the pressure level can be chosen according to the equation (2.48).

It's possible to estimate the stiffness coefficient and the damping coefficient respectively with a static and dynamic test [16]. These parameters are used in the simulation environment in order to simulate the behaviour of the system modelled with PRBM.

The table below shows the estimated parameters.

P [KPa]	f_s [Hz]	c [Nms/rad]	k[Nm/rad]
10	15.04	0.557	321
20	15.63	0.547	347
30	15.94	0.463	360
40	16.17	0.450	370
50	16.44	0.433	382
60	16.56	0.429	388
70	16.65	0.412	392
80	16.83	0.407	401
90	16.93	0.406	405

Table 3 Estimated parameters

2.2.1 Denavit-Hartenberg convention 7dof

The pseudo-rigid body model, seen in the previous paragraph, adds 2 virtual joints for each link since, only bending deformations are considered along two orthogonal axes while torsional deformations are neglected in order to deal with a simplified model. The virtual joints are called $\delta_{i,j}$ where i indicates the link number while j the orientation (along x or y axes). The robotic arm now can be represented with the scheme shown in Figure 22, where the white cylinders are the rigid joints while the grey ones are the virtual joints.

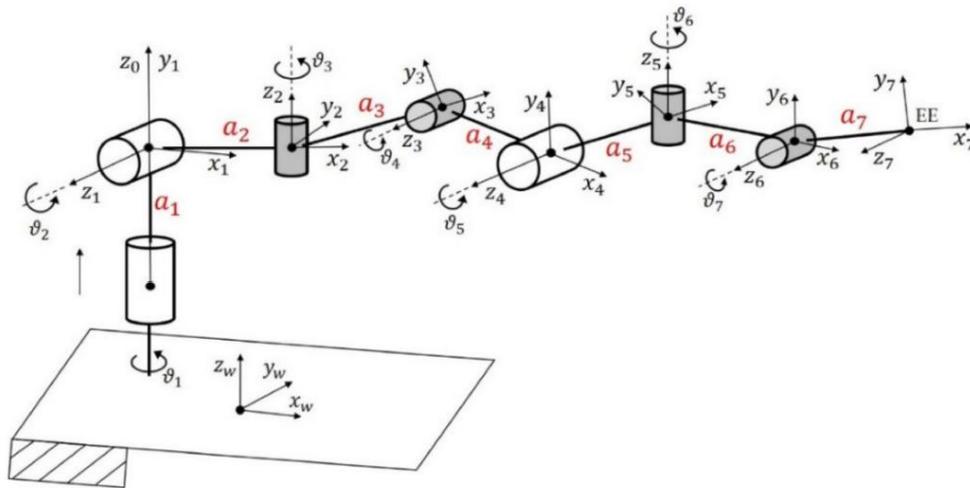


Figure 22 Representation of the 7 dof representation

Now, it's possible to set the D-H parameters of the pseudo-rigid model.

Link	θ_i (rad)	d_i (m)	a_i (m)	α_i (rad)
0	θ_1	0	0	$\pi/2$
1	θ_2	0	0.03	$-\pi/2$
1.1	$\delta_{1,1}$	0	0	$\pi/2$
1.2	$\delta_{1,2}$	0	0.57	0
2	θ_3	0	0.03	$-\pi/2$
2.1	$\delta_{2,1}$	0	0	$\pi/2$
2.2	$\delta_{2,2}$	0	0.57	0

Table 4 Denavit-Hartenberg parameters pseudo-rigid model

2.2.2 Link structure in Simscape modelled with PRBM

Simscape is a software developed by Mathworks that is used to rapidly create models of physical systems within the Simulink environment. The Simscape model can be ran with the daessc solver that is a solver providing algorithms specifically designed to simulate differential algebraic equations arising from physical systems. In addition, with this solver it's possible to open a window called 'Mechanical explorer' that shows how the system behaves during the simulation.

Since, every link is modelled as a pseudo-rigid body with a virtual joint, in Simscape the structure of every link is shown in the Figure 23.

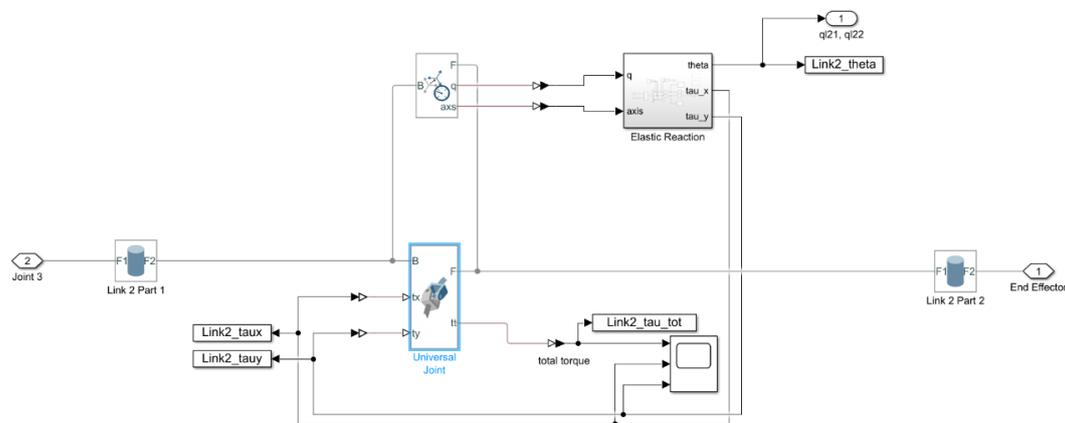


Figure 23 Pseudo-rigid link in Simscape

The universal joint block in the middle represents the virtual joints and it is a joint with two rotational degrees of freedom, while the link is split between part 1 (on the left) and part 2 (on the right). The first part of the link has length $L_1 = 30 \text{ mm}$ (this is the length where the first wrinkle appear) and the second one has length $L_2 = 570 \text{ mm}$. At the top, it's possible to distinguish a sensor between the 2 parts of the link and this is used to measure the virtual joint variables along axes x and y and these are exported in the workspace through the 'To Workspace' block 'Link2_theta'. Furthermore, the 'Elastic Reaction' block is used to compute the

bending moment along both axes and furthermore, it is able to write an alarm condition in the command window if a wrinkling moment occurs.

2.3 Finite element method (FEM)

The PRBM parameters are derived experimentally, as seen before, through a static and dynamic test. In order to find appropriate parameters to obtain an equivalent FEM structure it is necessary to perform a fitting test comparing the response of the PRBM link with respect to the FEM one. In Simulink a FEM link is represented using Flexible Cylindrical beam block. This block exploits the Euler-Bernoulli beam theory and the flexible elements can have elastic bending, axial and torsional deformations. This block requires as parameter the geometrical ones that are the same used of the PRBM, the Young Modulus, the density, the Poisson's Ratio and for what regards damping coefficient are required two coefficients. The first damping coefficient is proportional to the mass matrix and the second one to the stiffness matrix following the Rayleigh damping definition. The mass damping coefficient is neglected since the inflatable link is a compliant structure and for this reason only the stiffness damping coefficient is estimated. The Rayleigh damping is a viscous damping [20] which is proportional to a linear combination of mass and stiffness following the following formula:

$$\mathbf{C} = \mu\mathbf{M} + \lambda\mathbf{K} \quad (2.49)$$

In the fitting experiment a dataset is collected that is then exploited as reference output. Then, the Parameter Estimator toolbox is used and the stiffness damping coefficient and Young's Modulus are set as variable parameters with initial value 0.002 s and 0.4 GPa, respectively. The thickness is fixed and chosen equal to 1.2 mm and the payload is equal to 1 kg. After some iterations, the following values are obtained:

- $E = 0.55741$ GPa;
- $\lambda = 3.896 * 10^4$ s.

In figure 25, the red curve represents the output of the PRB link with 2kg of payload in the presence of gravity. It is chosen the vertical displacement as output. This system is underdamped and the fitting experiment aims to find appropriate parameters required by Flexible Cylindrical beam block. The blue curve is the output of the FEM link with the parameters estimated by the toolbox. The two curves aren't perfectly overlapped and there is a steady state error.

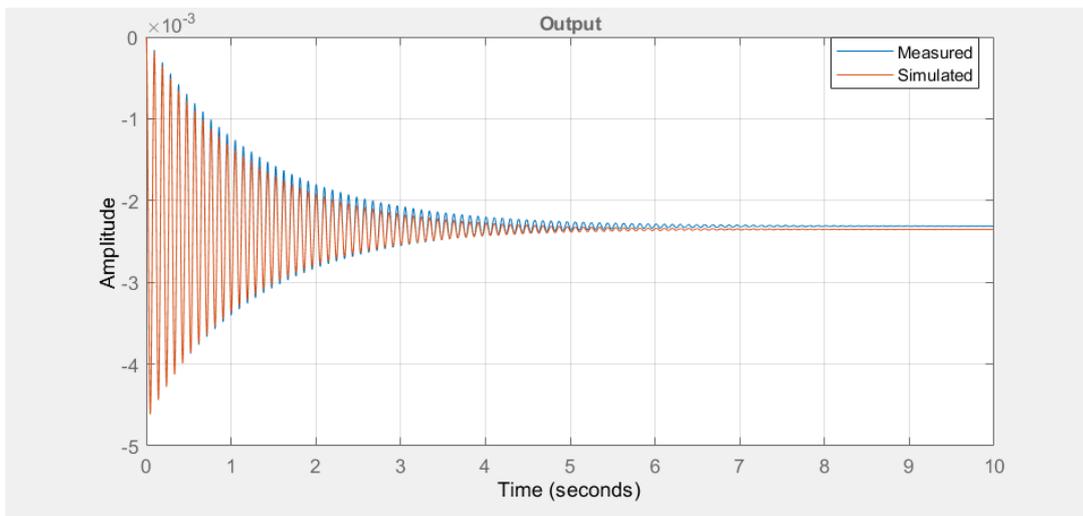


Figure 24 First fitting experiment

Another fitting experiment is done with the same procedure as before but with a different thickness and payload. Now the thickness is set equal to 0.8 mm and the payload is still equal to 2kg. In this experiment the fitting is more accurate than the previous one because the two responses converge and they are perfectly overlapped. The values that fit better the two curves are:

- $E = 794.9 \text{ MPa}$;
- $\mu = 3.4919 * 10^4 \text{ s}$.

With these values the red curve in Figure 25 is obtained and it's possible to observe that this curve follows the blue one that is the reference output.

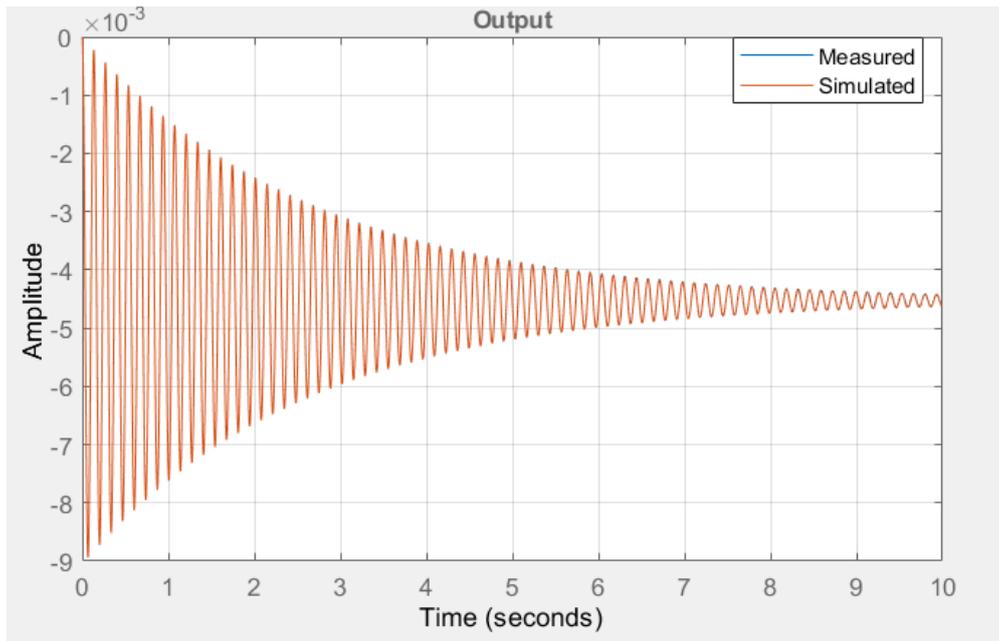


Figure 25 Second fitting experiment

Chapter 3

Set membership identification

A trial to identify the entire system in the frequency domain is done exploiting the set membership theory. The aim of this experiment is to identify the black box shown in figure 27 that is a MIMO system whose entries are the joint position and velocity (for now the first motoreducer is neglected in order to have a simplified system) whereas the outputs are the cartesian positions of the end effector with respect to the base frame.



Figure 26 Representation of the system as a black box

Before anything, we first have to formulate the problem in the set membership theory. Set membership theory is based on three fundamental ingredients: a priori information on the system, a priori information on the noise and a set of collected data.

3.1 A priori information on the system

The system to be identified is a multiple-input and multiple-output (MIMO) system since it is characterized by four inputs and two outputs. As a result, the shape adopts the following structure:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} G_1 & G_2 & G_3 & G_4 \\ G_5 & G_6 & G_7 & G_8 \\ G_9 & G_{10} & G_{11} & G_{12} \end{pmatrix} \begin{pmatrix} q_2 \\ \dot{q}_2 \\ q_3 \\ \dot{q}_3 \end{pmatrix} \quad (3.1)$$

It is possible to separate this MIMO identification problem into a three multiple-input and single-output (MISO) problems:

$$1) \quad x = G_1 q_2 + G_2 \dot{q}_2 + G_3 q_3 + G_4 \dot{q}_3 \quad (3.2)$$

$$2) \quad y = G_5 q_2 + G_6 \dot{q}_2 + G_7 q_3 + G_8 \dot{q}_3 \quad (3.3)$$

$$3) \quad z = G_9 q_2 + G_{10} \dot{q}_2 + G_{11} q_3 + G_{12} \dot{q}_3 \quad (3.4)$$

Let's start considering only the x coordinates, then the SPARSEPOP code for the y and z coordinates is the same, obviously with a different dataset.

Let's assume that every transfer function G_i is a monic and a proper second order transfer function. We have to estimate the coefficients of every transfer function (since these are monic for each, we have to estimate 5 coefficients: 3 for the numerator and 2 for the denominator).

$$G_1 = \frac{b_1 z^2 + b_2 z + b_3}{z^2 + a_1 z + a_2} \quad (3.5)$$

$$G_2 = \frac{b_4 z^2 + b_5 z + b_6}{z^2 + a_3 z + a_4} \quad (3.6)$$

$$G_3 = \frac{b_7 z^2 + b_8 z + b_9}{z^2 + a_5 z + a_6} \quad (3.7)$$

$$G_4 = \frac{b_{10} z^2 + b_{11} z + b_{12}}{z^2 + a_7 z + a_8} \quad (3.8)$$

Where b_i and a_i are the parameters to be estimated, so overall we have twenty unknown parameters.

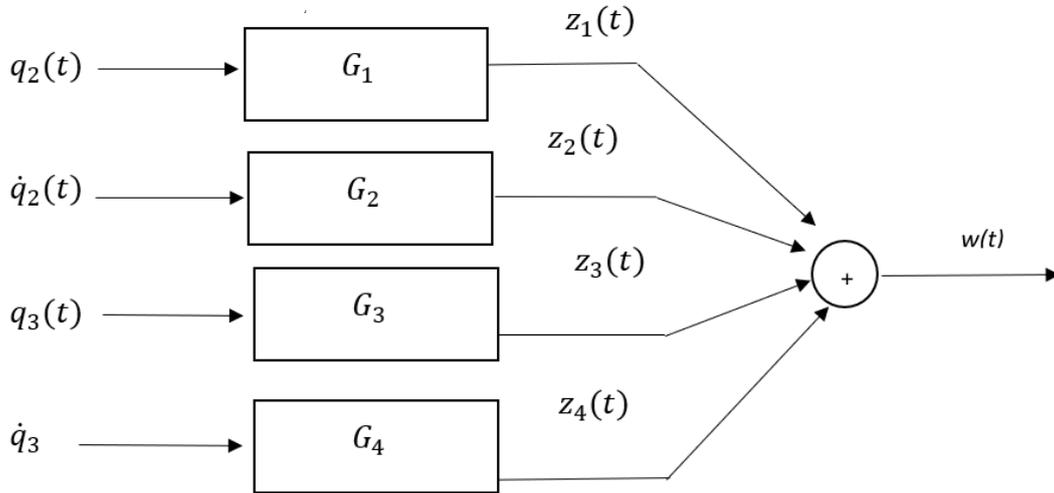


Figure 27 Block diagram of the system to be identified

The output of this system is given by four contributions called partial outputs z_i which are unknowns.

$$w(t) = z_1(t) + z_2(t) + z_3(t) + z_4(t) \quad (3.9)$$

But,

$$z_i = G_i u_i \quad (3.10)$$

where u_i can be or the joint position or the joint velocity. Now, let's introduce the backward shift operator that is the equivalent of the z-transform but in the time domain.

$$z^{-1} = q^{-1} \quad (3.11)$$

$$q^{-n} s(k) = s(k - n) \quad (3.12)$$

Now, we can express the previous expression of the transfer functions in function of time exploiting the definition of the backward shift operator.

$$G_1 = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{b_1 + b_2 q^{-1} + b_3 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} \quad (3.13)$$

$$G_2 = \frac{b_4 + b_5 z^{-1} + b_6 z^{-2}}{1 + a_3 z^{-1} + a_4 z^{-2}} = \frac{b_4 + b_5 q^{-1} + b_6 q^{-2}}{1 + a_3 q^{-1} + a_4 q^{-2}} \quad (3.14)$$

$$G_3 = \frac{b_7 + b_8 z^{-1} + b_9 z^{-2}}{1 + a_5 z^{-1} + a_6 z^{-2}} = \frac{b_7 + b_8 q^{-1} + b_9 q^{-2}}{1 + a_5 q^{-1} + a_6 q^{-2}} \quad (3.15)$$

$$G_4 = \frac{b_{10} + b_{11} z^{-1} + b_{12} z^{-2}}{1 + a_7 z^{-1} + a_8 z^{-2}} = \frac{b_{10} + b_{11} q^{-1} + b_{12} q^{-2}}{1 + a_7 q^{-1} + a_8 q^{-2}} \quad (3.16)$$

Now, let's replace:

$$q_2 = \theta_1 \quad (3.17)$$

$$\dot{q}_2 = \theta_2 \quad (3.18)$$

$$q_3 = \theta_3 \quad (3.19)$$

$$\dot{q}_3 = \theta_4 \quad (3.20)$$

So, we can write

$$w(t) = z_1(t) + z_2(t) + z_3(t) + z_4(t) = G_1 \theta_1 + G_2 \theta_2 + G_3 \theta_3 + G_4 \theta_4 \quad (3.21)$$

3.2 A priori information on the noise

The noise is entering the problem as an OE structure (output error structure), which means that the output is affected by a bounded additive noise and the scheme of this noise structure is shown in figure 28.

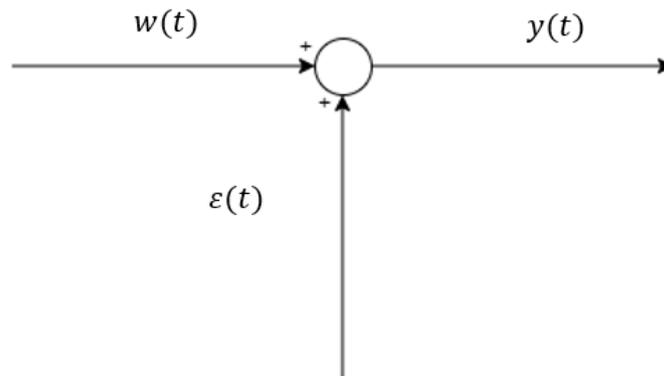


Figure 28 OE structure

$$y(t) = w(t) + \varepsilon(t) \quad (3.22)$$

Since, the output error is bounded of a certain quantity, let's suppose that the bounded value is equal to 5mm:

$$|\varepsilon(t)| \leq 5mm \quad \forall t=1, \dots, N \quad (3.23)$$

3.3 Collected data

A set of N (where different values of N were tested but with the same results) input-output data have been collected using as model the pseudo-rigid body model. All the information can be used to define the feasible parameter set, D_θ , that is the set of all parameter values that are consistent with the a priori information on the system and noise, as well as with the collected data:

$$D_\theta = \{ \theta \in R^{20}: \text{eq. 3.21 substituting } G_i \text{ and merge with eq. 3.22,} \\ \text{ineq. 3.23} \} \quad (3.24)$$

In this case, the equations describing D_θ depend on the other unknown variables $\varepsilon(\cdot)$, so we have to define a higher dimensional space, called the extended feasible parameter set (EFPS):

$$D_{\theta,\varepsilon} = \{\theta \in R^{2^0}, \bar{\varepsilon} = \begin{bmatrix} \varepsilon(1) \\ \dots \\ \varepsilon(N) \end{bmatrix} : \text{same equations as before}\} \quad (3.25)$$

The EFPS has a very complex shape, so we can define the parameter uncertainty interval (PUI) for each parameter θ_i :

$$PUI_{\theta_i} = [\underline{\theta}_i, \overline{\theta}_i] \quad (3.26)$$

To find these two extremes, we have to solve two optimization problems having the following form:

$$\underline{\theta}_i = \min \theta_i \quad (3.27)$$

$$\overline{\theta}_i = \max \theta_i = \min(-\theta_i) \quad (3.28)$$

Subjected to the constraints inside the EFPS.

Since first constraints are bilinear, the computation of the previous extremes requires the solution of the two polynomial optimization problems, which are non-convex problems. Standard mathematical algorithms fail to compute the global optima with such kinds of problems. That's why we use the convex-relaxation approach, which consists of relaxing the non-convex constraints in the original problem. We will find a set of convex problems that depend on the order of relaxation δ . As this parameter increases, the convex approximation converges to the convex hull of the EFPS. The starting value for this parameter is:

$$\delta_{min} = \frac{\max degree}{2} = 1 \quad (3.29)$$

In, order to solve the problem through SparsePOP we have to write the problem in the 'SparsePOP language' by following the guide in [21]. In particular all the equation must be of the form:

$$f(x) = 0 \quad (3.30)$$

And the inequality must be of the form:

$$g(x) \geq 0 \quad (3.31)$$

Unfortunately, with this method, though it was considered a simplified system without the first actuator and only using position and velocity without acceleration, and also a linear approximation the result was the same for several simulations with different number of samples, with different orders of relaxation. The obtained result after several hours of simulation for each simulation is the following error message: “*Out of memory*”. This means that the RAM memory available isn’t enough to solve this optimization problem. This happens when a code operates large amounts of data.

Since the simplified estimation was unable to compute the parameters of the transfer functions mentioned above, the complete version of the system is very complex to be estimated because it is computationally more demanding: it is similar to the previous one with an additional motoreducer and also with a non-linear map that accounts for the non-linearity of the system under investigation. The entire code used to estimate the simplified system is reported in the Appendix.

Chapter 4

Control system for robotic arms

4.1 Control in the operational space

The scheme of control in the operational space is shown in figure 29. The reference is provided in the operational space and this enters directly in the feedback control scheme. The drawback of this control scheme is that it is more complex from a computational point of view since it requires the knowledge of the robot mathematical model.

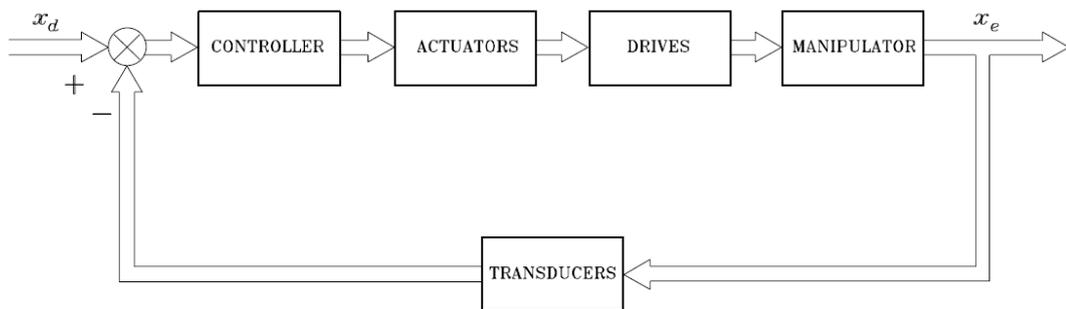


Figure 29 Control scheme in the operational space

4.2 Control in the joint space

The control scheme in the joint space is shown in figure 30. The reference is provided in the operational space and it must be translated to the joint space task specification through the inverse kinematics block. Then, there is a feedback chain between the desired joint angle and the measured one. The aim of this control scheme is to determine the generalized torques to be sent in the revolute joints in such a way that the joint angular position signals are equal to the desired ones.

The disadvantage of this control scheme is that x_d and x_e are in open loop so the precise position of the end effector is not known. However, the end effector position could be computed through the direct kinematics.

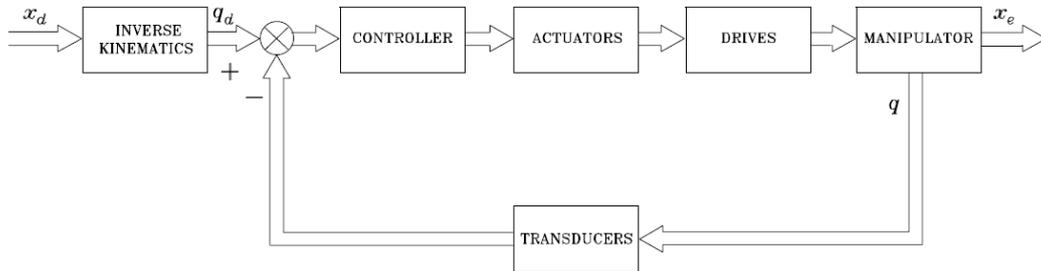


Figure 30 Control scheme in the joint space

4.2.1 Decentralized control

The decentralized control treats each joint independently from the others and the coupling effects are considered as disturbances. The control scheme is shown in Figure 31.

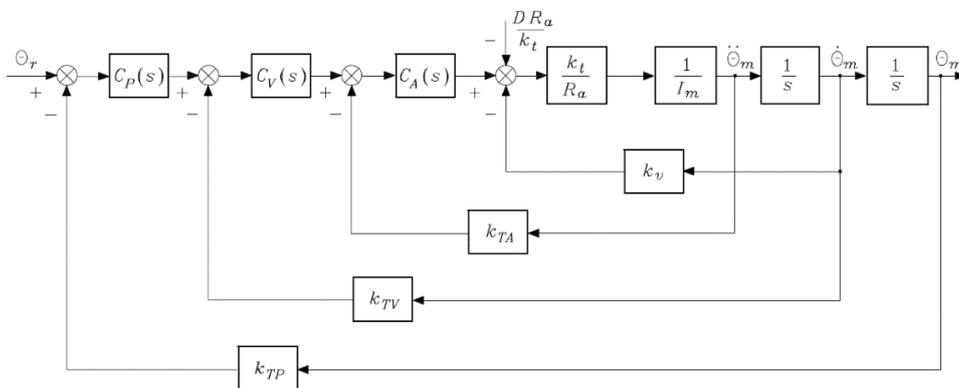


Figure 31 Decentralized control scheme

There are the three feedbacks respectively for the joint position, joint velocity and joint acceleration. These three feedbacks allow to have a more robust control because the entire dynamics of actuator is taken into account.

In this control scheme starting from the left there is the signal generator block that provides the reference joint position signal. Then, there are two feedbacks: position and velocity respectively with their PI controllers in order to achieve good performance. After that, another feedback current chain is present between the motoreducer and its correspondent driver and at on the right part there is the robot structure composed by links and joints. In addition, on the lower part the encoder and bending estimator are present which estimate the angular position, velocity and the angular deformations occurring along the links respectively.

At the beginning, in order to tune the PI gains, a rigid link is considered since it is simpler to control. The results are shown in the figures below. As input a step with amplitude equal to $\pi/2$ is provided to the system. The PI gains are:

- Decentralized Position Control: $K_p = 20$ and $K_i = 0.25$;
- Decentralized Velocity Control: $K_p = 5000$ and $K_i = 100$.

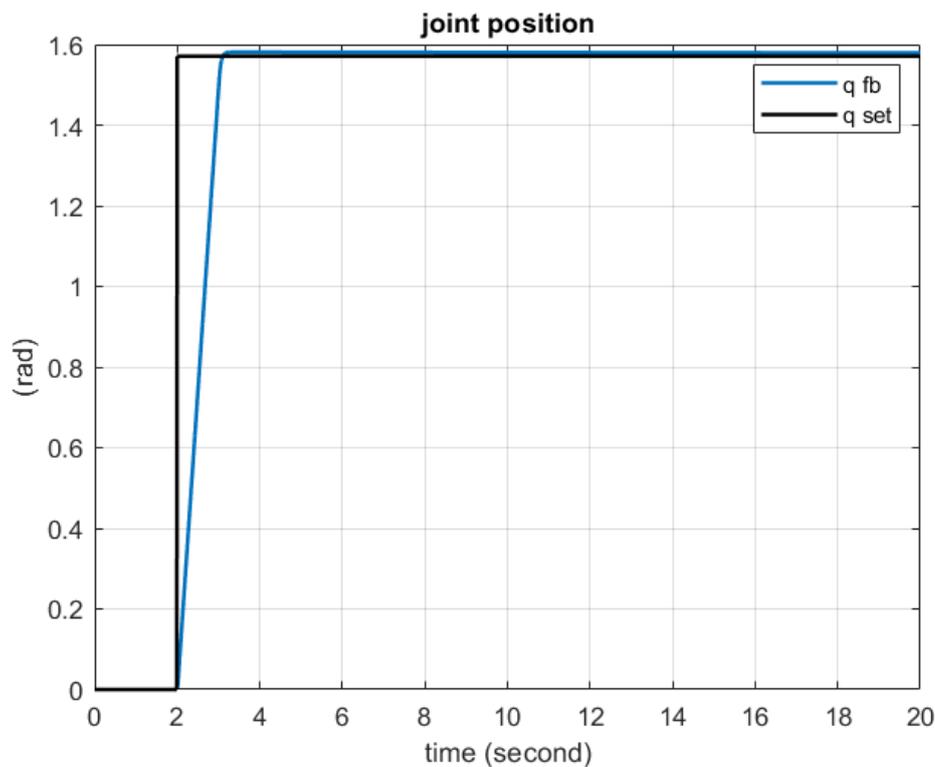


Figure 33 Reference and feedback joint angular position rigid robot

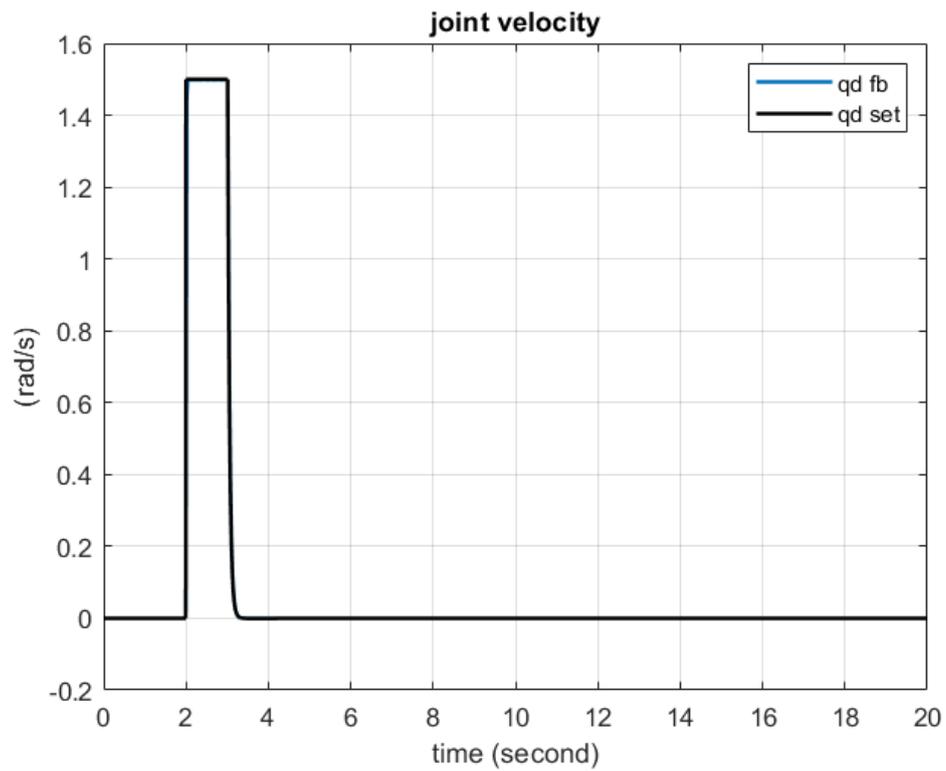


Figure 34 Reference and feedback joint angular velocity rigid robot

After that, the rigid link is replaced by the pseudo-rigid one by using the same gains of the rigid link the feedback velocity and position are shown in Figures 35 and 36 respectively.

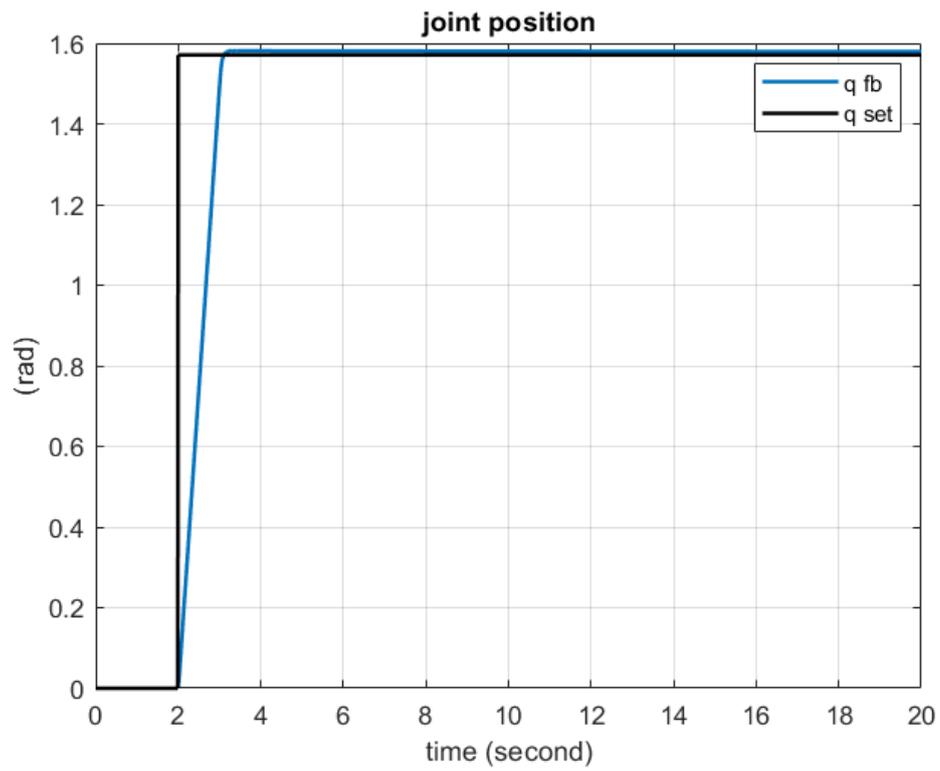


Figure 35 Reference and feedback angular position PRBM

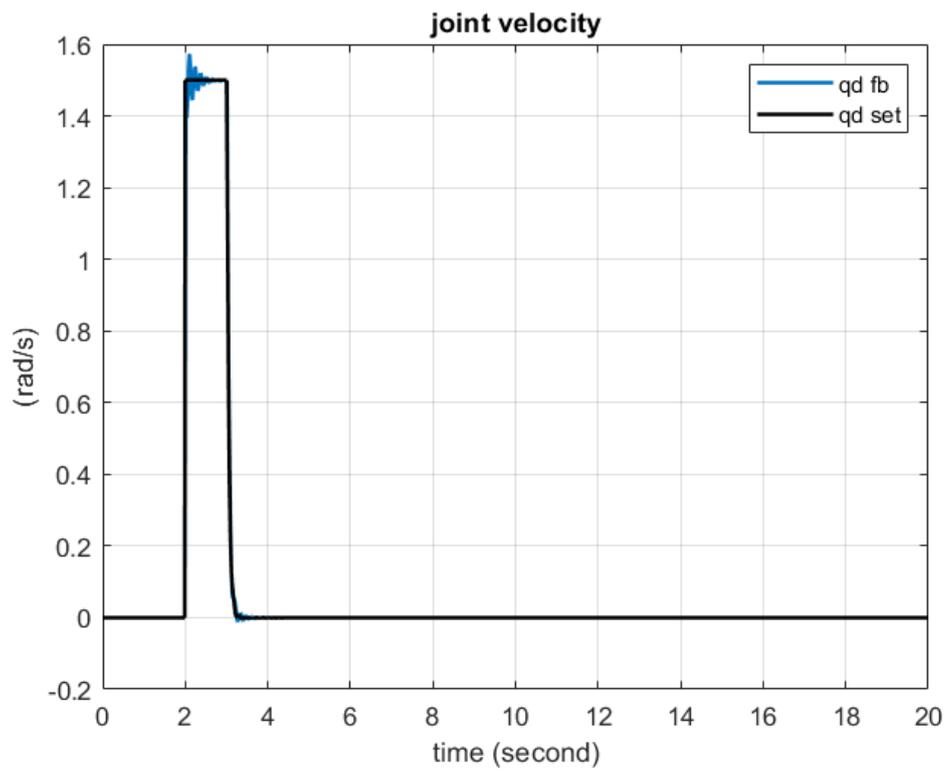


Figure 36 Reference and feedback angular velocity PRBM

The joint angular position is almost the same, while the feedback angular velocity now has an oscillatory behaviour and this is due to the flexibility of the link.

5.1.1 Vibration damping control for one link – one motoreducer system

After setting the PI parameters of the decentralized control system a vibration damping control is applied to the system. In particular a couple of PID controllers for each motoreducer. This choice is done since these controllers allow robustness performance in a wide range of operating conditions and the difficulty consist of tuning their gains.

The control scheme is similar to the one developed by Mahamood M. and Pedro O. in [22]. In this system the control input is given by the sum between the control action of the PI controller and the PID controller. The PI controller is used for tracking motion while the PID controller is designed for the control of end point relative acceleration of each link, and it aims to reduce vibrations. The control system for POPUP is shown in figure 37.

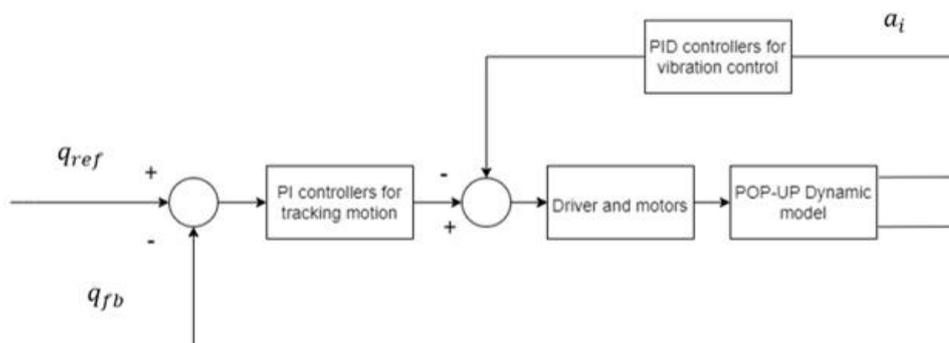


Figure 37 POPUP control system

The PI controllers receive as input the joint position and velocity tracking error since in order to have higher stability and a more precise dynamics two feedback chains are considered: one for angular position and one for angular velocity.

Instead, the PID control action has the following shape:

$$u_{PID_i} = k_{p_i}e_i + k_{I_i} \int e_i dt + k_{D_i}\dot{e}_i \quad (5.2)$$

Where: $e_i = \alpha_{i_d} - \alpha_i$ are the relative tip linear acceleration of the i-th link. In particular, for the second and third motoreducers the relative linear acceleration along x-axis of the local reference system shown in Figure 39 is considered while for the first one the difference of the two relative linear accelerations of both link along y-axis of the reference system shown in Figure 39 is considered since these are oriented in an opposite way. In order to reduce the oscillations, we want to maintain the desired acceleration in the neighbourhood of zero. The total control input is the following one:

$$u_{tot_i} = u_{PI_i} + u_{PID_i} \quad (5.3)$$

Figure 38 shows the POPUP feedback control system. The PI controllers are inside both the Decentralized Position Control and the Decentralized Velocity Control blocks (on the left) while the other ones are located inside the PID controllers block (on the right). The additional control action given by PID controllers are subtracted in the Decentralized Velocity control block since the velocity error is an acceleration because it is a difference between two velocities in a certain step size.

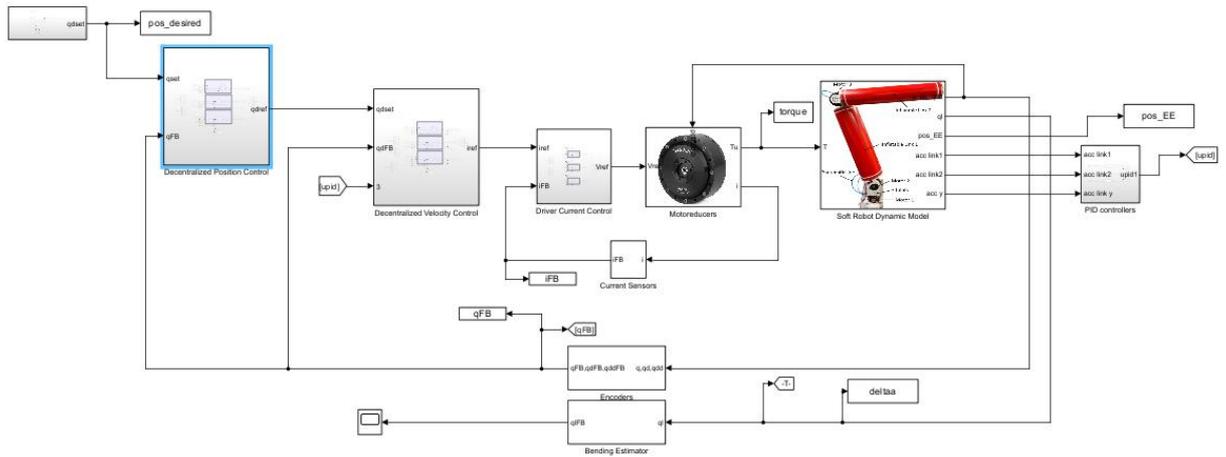


Figure 38 POPUP control scheme in Simulink

We have to attenuate the oscillations occurring in the x-axis with motor 2 and 3, since the local reference frame of the second link is oriented in the following way (the same holds for the first link).

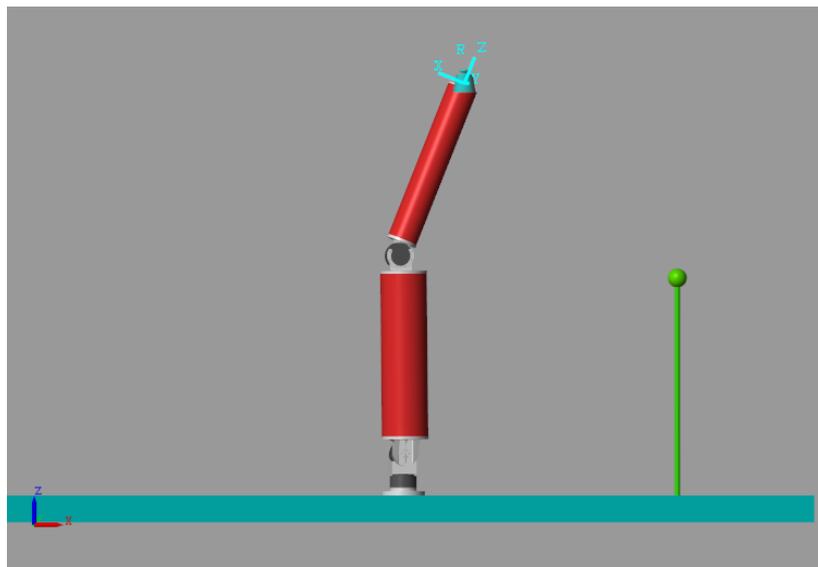


Figure 39 Local reference frame second link

The aim is to compensate oscillations occurring along the x-axis with the correspondent motoreducer (2 for the first link while 3 for the second link) and the ones occurring along the y axis could be compensated thanks to the first actuator.

5.2 Simulations

5.2.1 One Link – one motoreducer system simulations

Firstly, a simplified version of the whole system is taken into account. In particular the first step is to control one link and one motoreducer system as shown in Figure 40. The dimensions of the inflatable link are chosen equal to the second link of the POPUP.

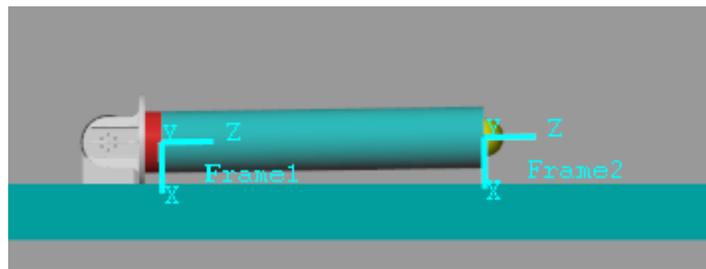


Figure 40 One link-one motoreducer system

The following simulations are done with these parameters:

- $q_0 = 0^\circ$;
- $p = 30\text{kPa}$;
- $\text{payload} = 2\text{kg}$.

With these parameters the arm can move a payload of 2kg without reaching wrinkling moment for the appropriate choice of input velocity.

$$M_w = \frac{\pi}{4} \pi p r^3 = 12.32 \text{ (Nm)}$$

This means that: $M_w = \text{payload} * g * L = \text{payload} * 9.81 * 0.6 \text{ (Nm)}$

$$\text{payload}_{max} \cong 2 \text{ kg}$$

Several gains are tried with the trial-and-error procedure but unfortunately no significant results are obtained. For this reason, a frequency analysis is performed for the simplified one link-one motoreducer system in order to add a low-pass filter with the lower cut-off frequency.

Chapter 6

Frequency analysis

The frequency response function (FRF) represents the behaviour of the system output in the frequency domain. In particular, the Bode plot is a particular kind of frequency response and it is made up of two graphs. The first graph expresses the magnitude of the frequency response in decibels while the second one represents the phase shift. In order to obtain Bode plot a sine wave with different frequencies has to be sent to the system. It is chosen the response of the system to a sine wave because exploiting the Fourier theory it is known that every signal could be expressed as a sum of sinusoidal waves. These graphs are fundamental to derive some important parameters for the control theory, like cut-off frequency, resonance frequency or gain and phase margins.

In the POPUP robot it is possible to perform two principal analyses in order to analyse the behaviour of the motoreducer and of the system composed by link and joint. Moreover, in this case let's start for the simplified case exploiting the one link and one joint system, considering the dimensions of the second link of the POPUP.

6.1 Frequency analysis of the motoreducer

In order to obtain the FRF of the driver and motoreducer, the reference joint velocity and the feedback velocity are chosen as input and output, respectively. The first step consists of simulating the driver – motoreducer system in different configurations. The tables below shows the cut-off frequencies of the driver - motoreducer system loaded with the link, which is represented in the three different ways seen in the previous chapter. A different amplitude of the input amplitude signal is considered in such a way to test the response of the system with a different input amplitude velocity. Only the velocity feedback chain is considered.

In Figures 41, 42 and 43 the FRF of the motoreducer loaded with an attached link rigid, FEM and pseudo-rigid respectively are shown. A payload of 2 kg is added at the extremity of the link. By increasing the input amplitude the cut-off frequency decreases and also the bandwidth of the system as a consequence. These simulations are done with a PI in the decentralized velocity control with proportional gain of 5000 and integral gain of 100.

1) With gravity, initial position 0° , 2 kg of payload

Cut-off frequency	Rigid	FEM	Pseudo-rigid
3 rad/s	12.9 rad/s	49.2 rad/s	50 rad/s
1 rad/s	38.3 rad/s	151.3 rad/s	151 rad/s
0.1 rad/s	376 rad/s	707 rad/s	707 rad/s

Table 5 Cut-off frequency for the motoreducer and driver simulation 1

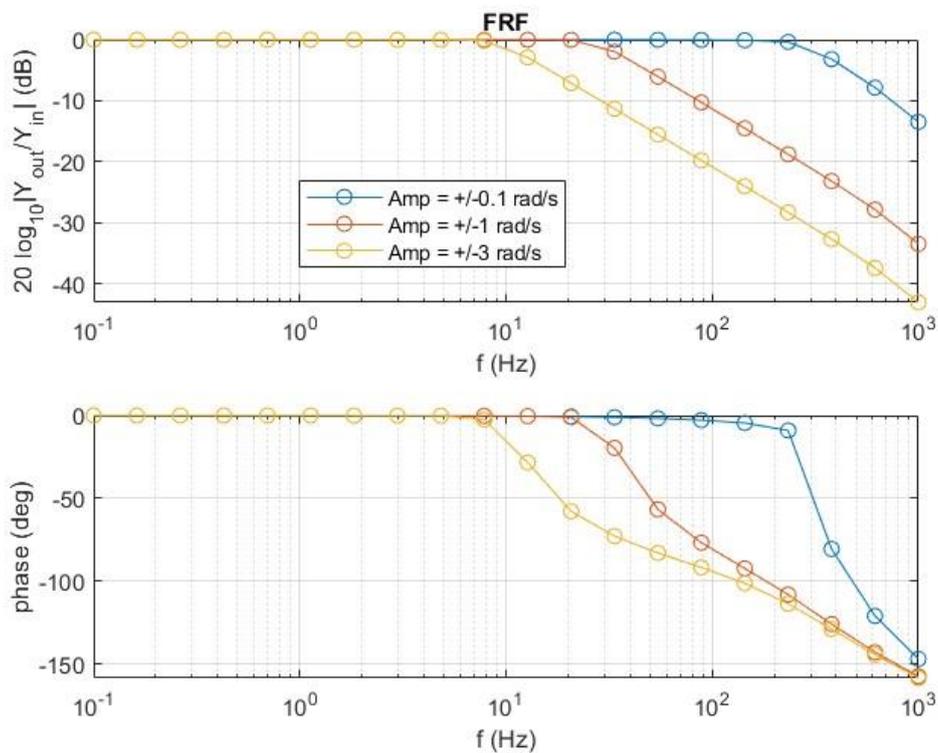


Figure 41 FRF motoreducer with an attached rigid link

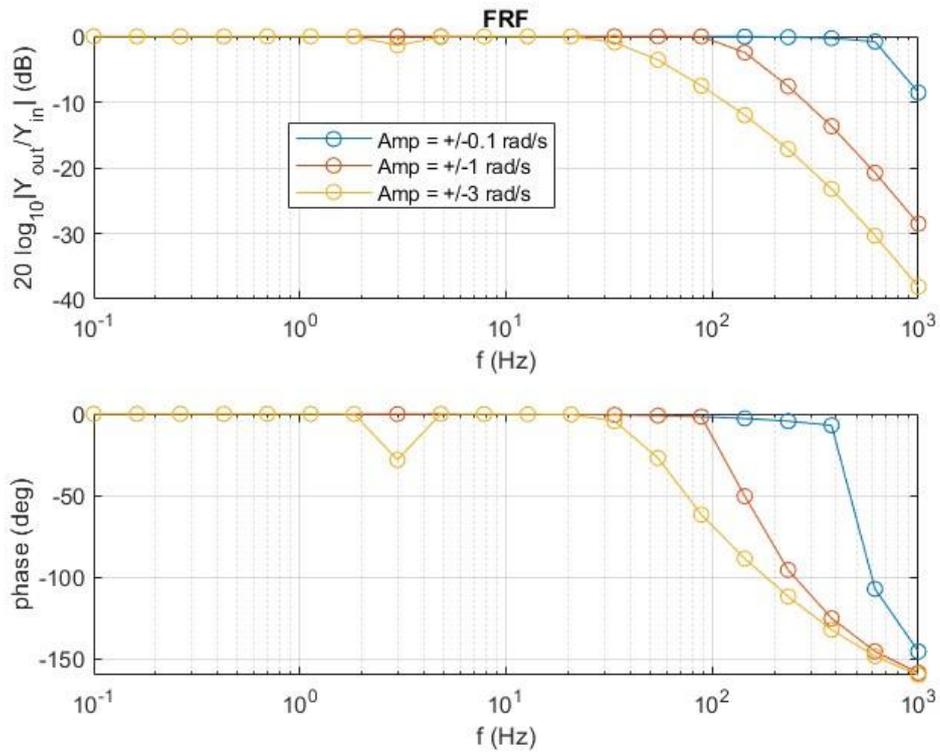


Figure 42 FRF motoreducer with an attached FEM link

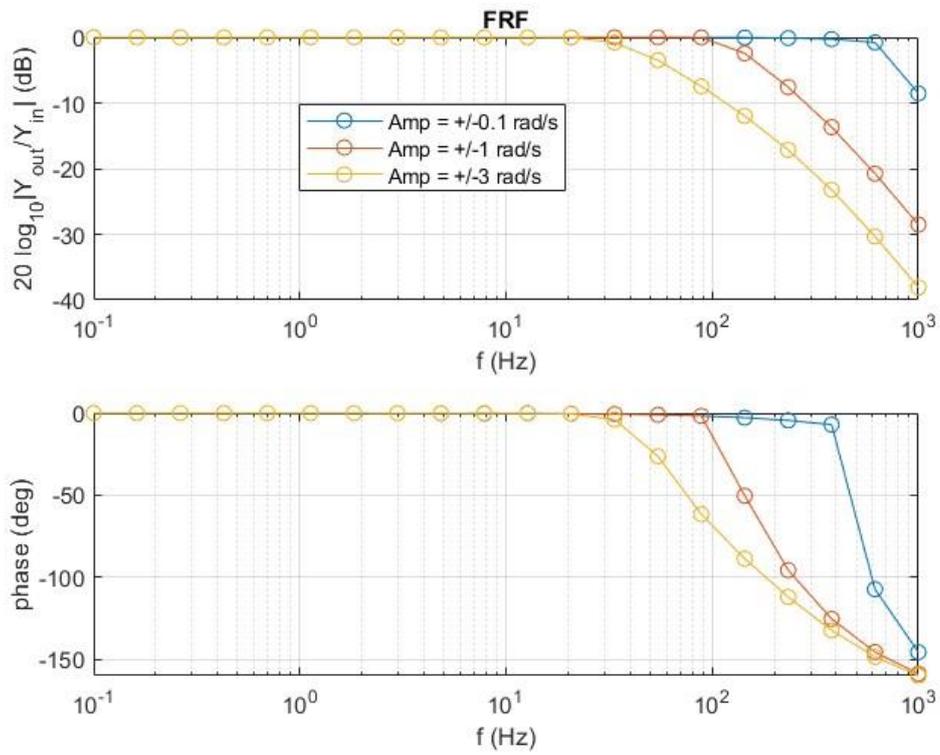


Figure 43 FRF motoreducer with an attached PRB link

2) With gravity, initial position 0° , 1 kg of payload

Cut-off frequency	Rigid	FEM	Pseudo-rigid
3 rad/s	25 rad/s	49.5 rad/s	50 rad/s
1 rad/s	77 rad/s	151.8 rad/s	165 rad/s
0.1 rad/s	580 rad/s	709 rad/s	710 rad/s

Table 6 Cut-off frequency for the motoreducer and driver simulation 2

3) With gravity, initial position 0° , 0 kg of payload

Cut-off frequency	Rigid	FEM	Pseudo-rigid
3 rad/s	25 rad/s	49.6 rad/s	47.8 rad/s
1 rad/s	75 rad/s	151.8 rad/s	152 rad/s
0.1 rad/s	580 rad/s	709.7 rad/s	710 rad/s

Table 7 Cut-off frequency for the motoreducer and driver simulation 3

Without gravity, the results are almost the same. This system behaves as a low-pass filter.

Now, let's derive the motoreducer FRF without any load. In order to do that, both payload and attached link are removed.

4) Only motoreducer without link

Cut-off frequency	
3 rad/s	49.1 rad/s
1 rad/s	151.5 rad/s
0.1 rad/s	707 rad/s

Table 8 Cut-off frequency for the motoreducer and driver simulation 4

By looking at the table 8 values, it can be observed that the load doesn't affect the behaviour of the system and for this reason the motoreducer are very performant for this system. The only parameters that changes considerably the cut-off frequency is the amplitude of the input velocity signal. The Figure 44 shows the FRF of the motoreducer without any attached link.

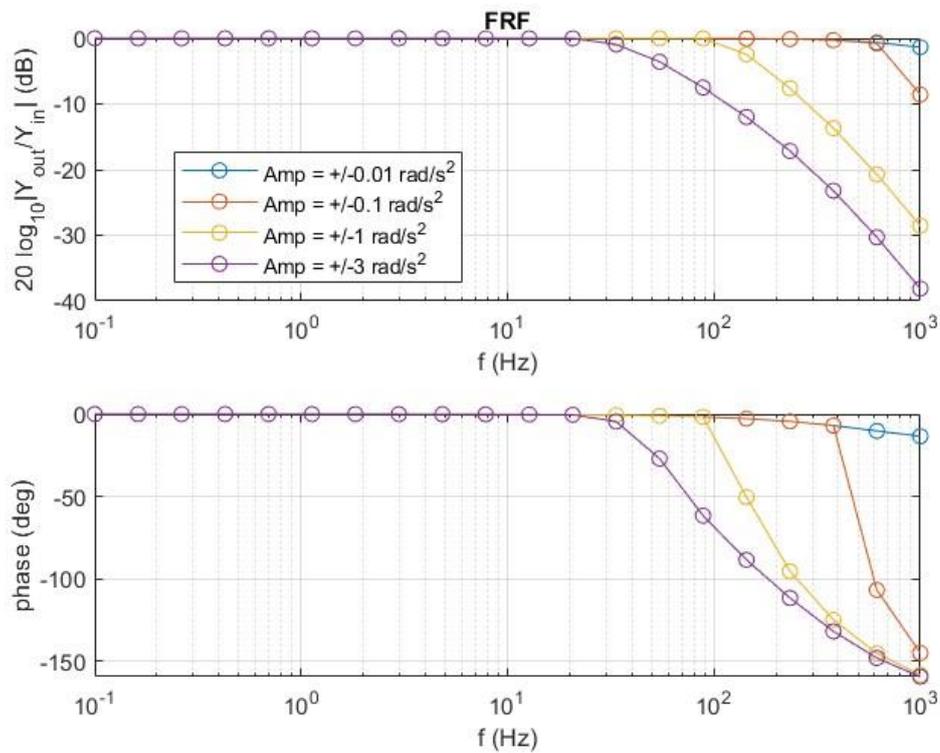


Figure 44 FRF motoreducer without any attached link

Then, a FRF of the motoreducer without any load and variable proportional gain is performed. In this analysis the input amplitude signal is constant and its value is 1 rad/s.

- 5) Only motoreducer without link and with different proportional gain and amplitude of sine wave equal to 1.

Cut-off frequency	
$K_p = 1$	0.49 rad/s
$K_p = 10$	5.6 rad/s
$K_p = 100$	56 rad/s
$K_p = 1000$	152 rad/s
$K_p = 5000$	152 rad/s

Table 9 Cut-off frequency for the motoreducer and driver simulation 5

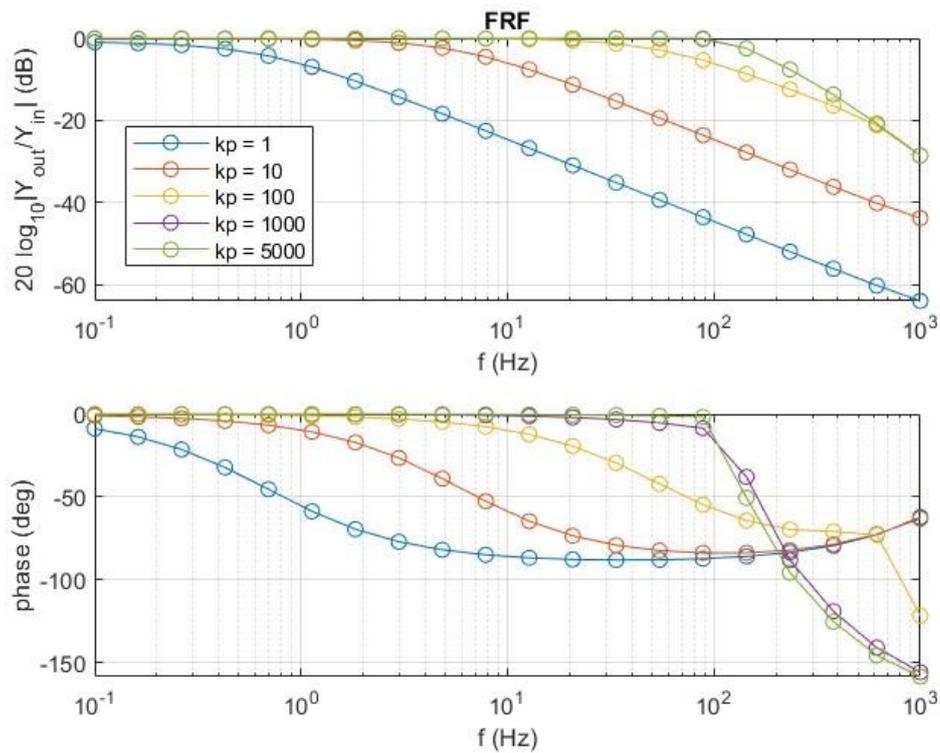


Figure 45 FRF motoreducer without any attached link with variable proportional gain

For what regards the proportional gain in the decentralized velocity control a value that guarantees good performances is higher than 100.

6.2 Frequency analysis of the link

In order to obtain the FRF of the driver and motoreducer the feedback joint velocity is chosen as input and the tip angular velocity along y-axis is chosen as output.

Firstly, a check on the output signal is done and in particular the right output of the rigid link must correspond to the input of the system because the ratio between output and input in the rigid link is equal to one.

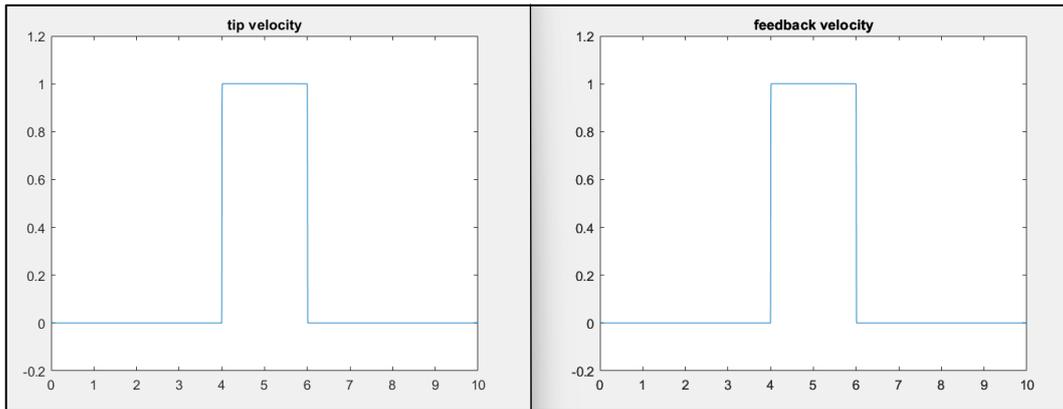


Figure 46 Input and output for this frequency analysis in a rigid link

Now, several simulations are done for rigid, FEM and pseudo-rigid link with and without gravity and with a payload of 2 kg, 1 kg and without payload.

- 1) With gravity, 2 kg of payload, initial position 0°

	Rigid	FEM	Pseudo-rigid
Cut-off frequency	/	430 rad/s	13.5 rad/s
Resonance peak	/	23.5 dB	21 dB
Resonance frequency	/	7.8 rad/s	7.85 rad/s

Table 10 Cut-off frequency for the link simulation1

The FRF of the rigid link is shown in Figure 47 and it is a flat response that means that the input signal is transferred to the output without any attenuation or amplification since the body is a rigid one. Instead, the Figure 48 represents the FRF of the FEM link and it is characterized by two resonance peaks where the first one presents an higher resonance peak. Figure 49 shows the FRF of the pseudo-rigid link and it is characterized by only one resonance peak.

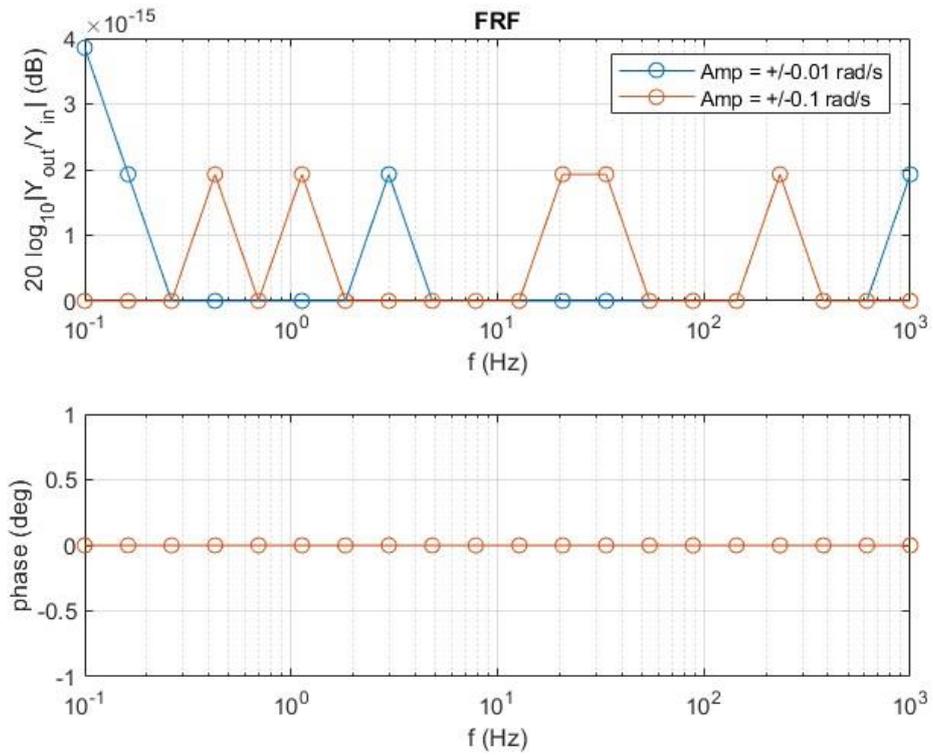


Figure 47 FRF rigid link

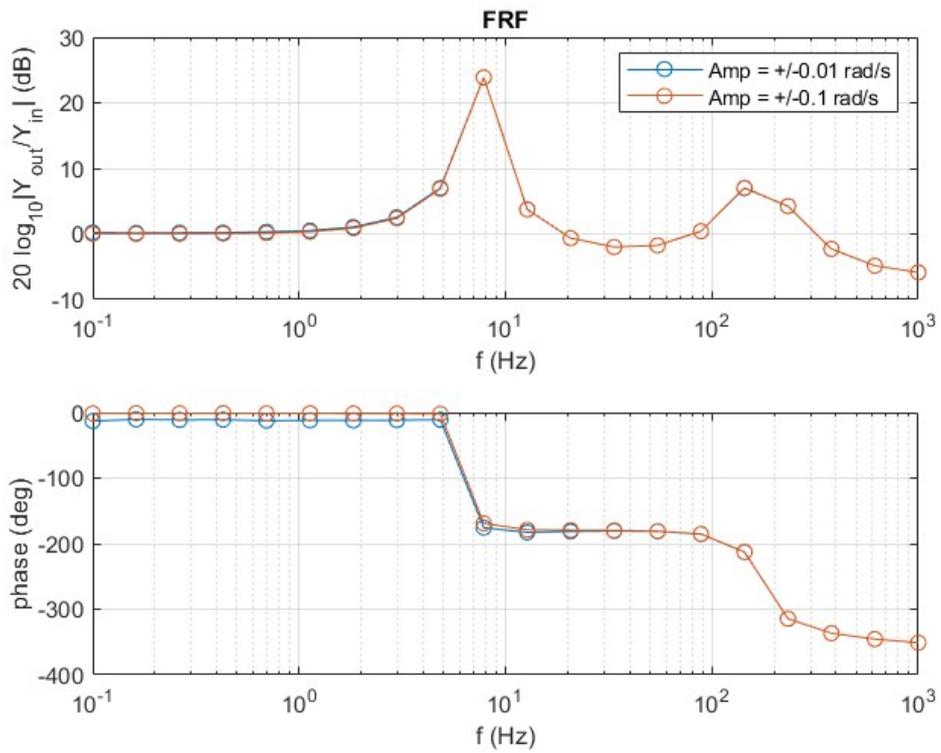


Figure 48 FRF FEM link

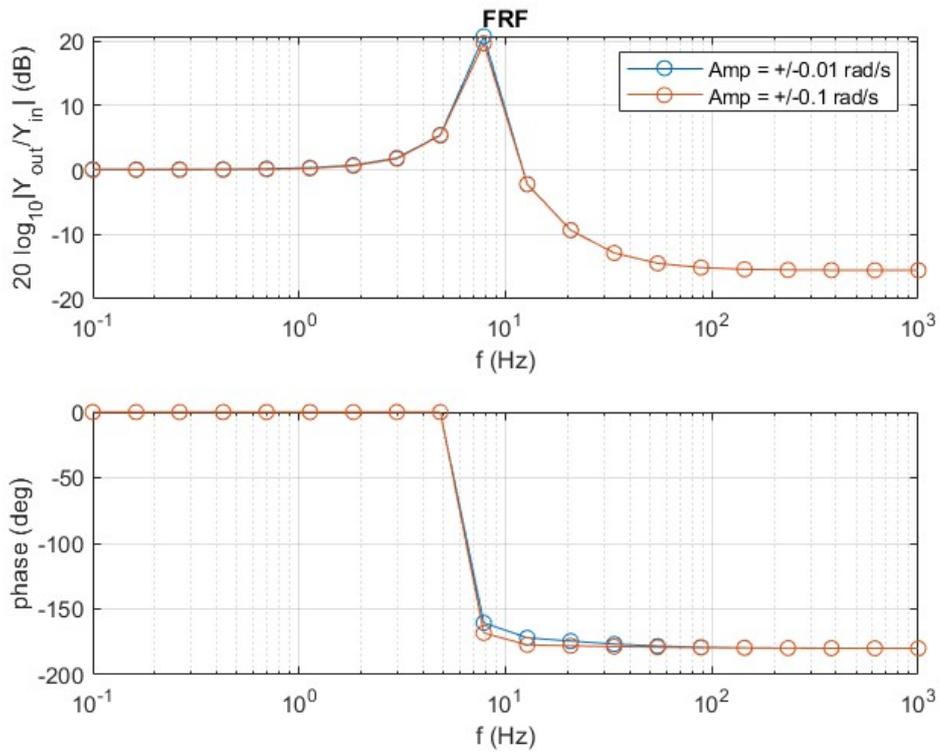


Figure 49 50 FRF PRB link

2) With gravity, 1 kg of payload, initial position 0°

	Rigid	FEM	Pseudo-rigid
Cut-off frequency	/	$>1000 \text{ rad/s}$	19.5 rad/s
Resonance peak	/	12 dB	8 dB
Resonance frequency	/	12.5 rad/s	11 rad/s

Table 11 Cut-off frequency for the link simulation2

3) With gravity, 0 kg of payload, initial position 0°

	Rigid	FEM	Pseudo-rigid
Cut-off frequency	/	> 1000 rad/s	147 rad/s
Resonance peak	/	13.5 dB	10 dB
Resonance frequency	/	55 rad/s	54.5 rad/s

Table 12 Cut-off frequency for the link simulation3

4) No gravity, 2 kg of payload, initial position 0°

	Rigid	FEM	Pseudo-rigid
Cut-off frequency	/	430 rad/s	13.5 rad/s
Resonance peak	/	24 dB	20 dB
Resonance frequency	/	7.85 rad/s	7.85 rad/s

Table 13 Cut-off frequency for the link simulation4

5) No gravity, 1 kg of payload, initial position 0°

	Rigid	FEM	Pseudo-rigid
Cut-off frequency	/	>1000 rad/s	19.2 rad/s
Resonance peak	/	12 dB	10 dB
Resonance frequency	/	12.7 rad/s	7.8 rad/s

Table 14 Cut-off frequency for the link simulation5

6) No gravity, 0 kg of payload, initial position 0°

	Rigid	FEM	Pseudo-rigid
Cut-off frequency	/	> 1000 rad/s	145 rad/s
Resonance peak	/	13.5 dB	9.7 dB
Resonance frequency	/	54.5 rad/s	145 rad/s

Table 15 Cut-off frequency for the link simulation6

Looking at the Bode plot of the PRB link, it's possible to say that the transfer function of the link contains two complex poles. Furthermore, the magnitude plot

of the PRBM link has one resonance peak while the FEM one has two resonance peaks. In addition, also in this case the cut-off frequency of the system decreases with an higher payload.

Chapter 7

Vibration damping control system with a Low-Pass filter

Thanks to the frequency analysis performed in the previous chapter, it is obtained the cut-off frequency of the link and the motoreducer. In particular, the lowest cut-off frequency of the PRBM link is taken in order to add a low-pass filter with this frequency before the PID controllers aiming to vibration control. In this way, the relative acceleration signal is filtered in such a way to attenuate the part of the signal with a higher frequency with respect to the cut-off one.

First of all, this control system is applied in the simplified one link – one motoreducer system as done before. The system is shown in Figure 50.

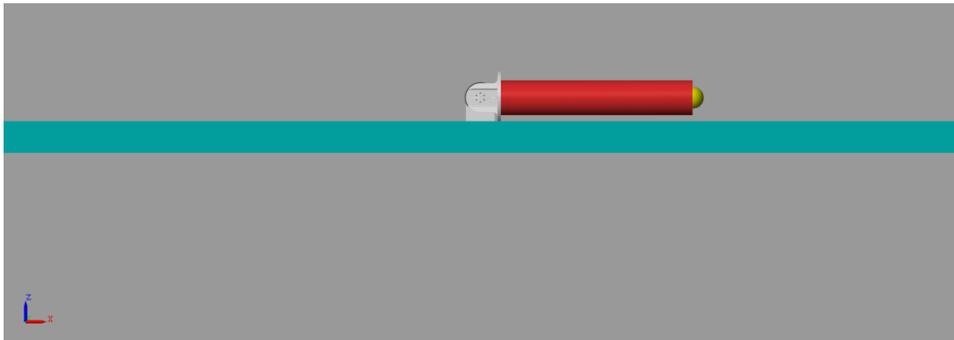


Figure 50 Simplified system

signal is sent to the PID controllers aiming to vibration suppression. Then, this control action is subtracted to the PI ones in the Decentralized Velocity subsystem.

The following simulations are done in Simulink environment and the pseudo-rigid body link is used.

7.1.1 Simulations without gravity

The first simulation is done with these parameters:

- The input reference position is a step with a delay of 2 seconds and $\pi/2$ of amplitude;
- Cut-off frequency of the filter equals to 13.5 rad/s;
- PID controller: $k_p = 1$, $k_i = 0.01$, $k_d = 0.01$;
- 2 kg of payload;
- No gravity;
- $q_0 = 0^\circ$;
- PI parameters Decentralized Position Control: $k_p = 30$, $k_i = 0.25$.
- PI parameters Decentralized Velocity Control: $k_p = 5000$, $k_i = 100$.

The results show how the additional control action aiming to vibration suppression improves the performance of the overall system. In particular, the deformations along x-axis in the virtual joint (Figure 52), the torque sent to the link (Figure 53), the end effector accelerations (Figure 54) are decreased. These improvements imply higher damping in the end effector position dynamics, resulting in fewer oscillations as shown in Figures 55 and 56

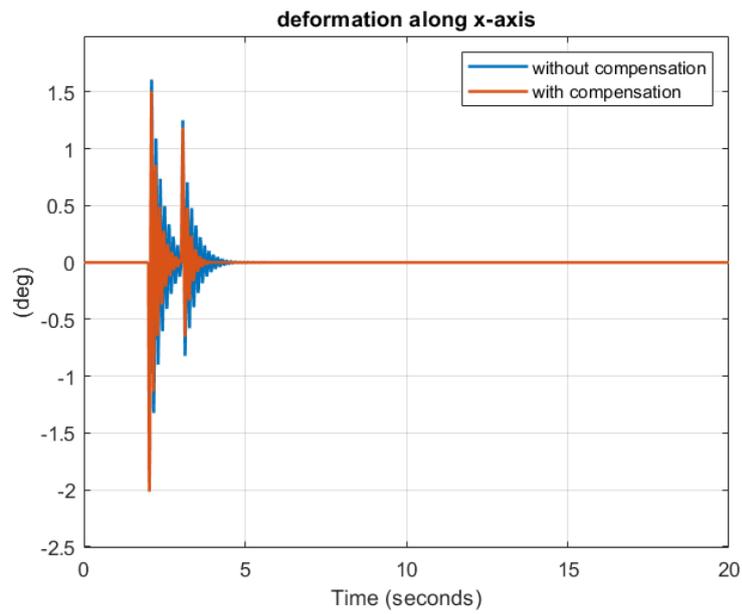


Figure 52 Deformation along x-axis with and without compensation

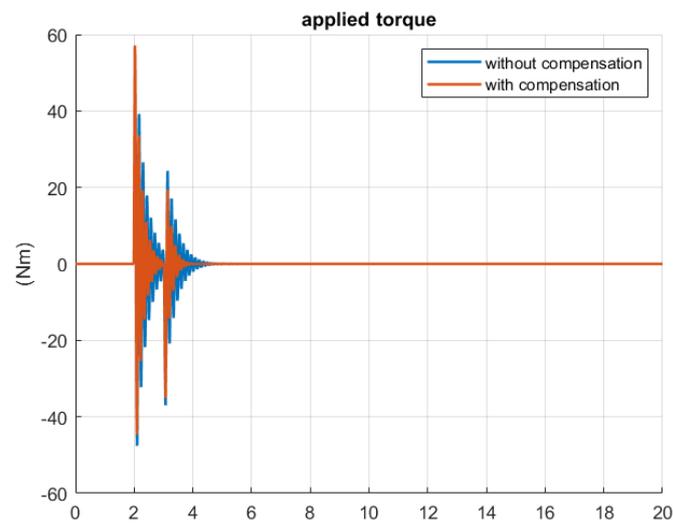


Figure 53 Torque applied to the joint with and without compensation

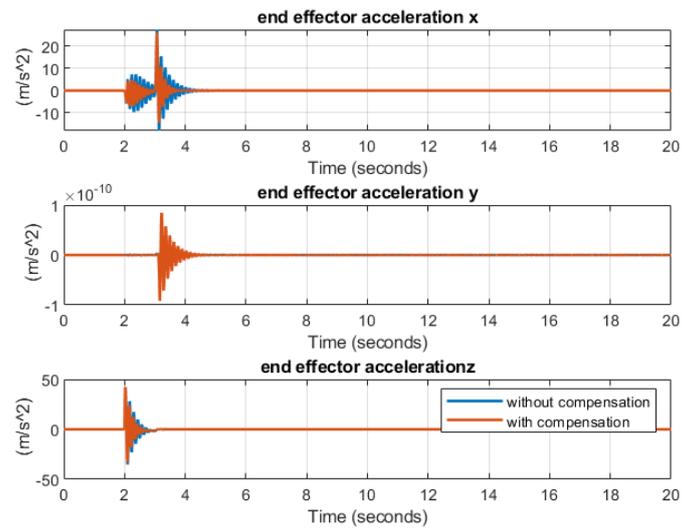


Figure 54 End effector accelerations with and without compensation

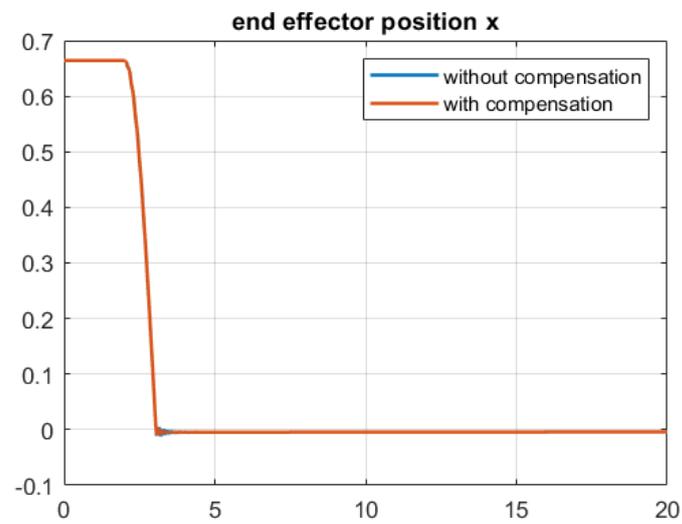


Figure 55 End effector position along x-axis with and without compensation

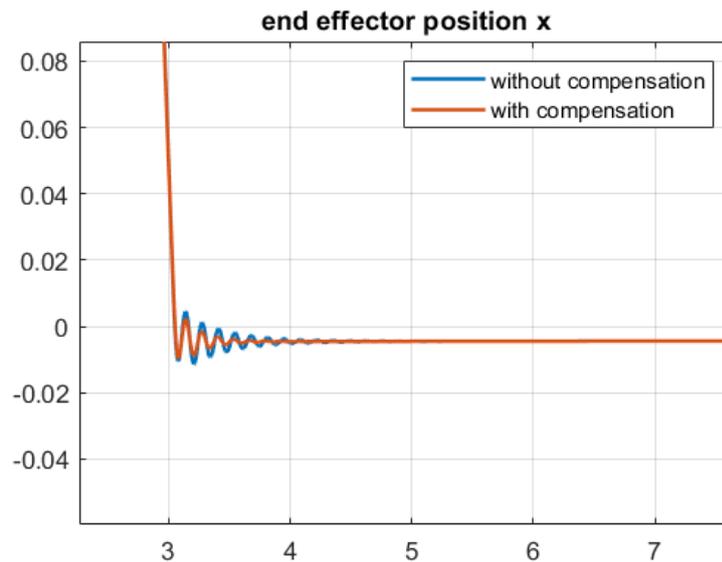


Figure 56 Zoom of end effector position along x-axis

Other quantities such as deformation along y-axis of the virtual joint and end effector position along y-axis are not shown since this is a planar robotic arm and these quantities are very smaller.

The second simulation is done with these parameters:

- The input reference position is a square wave;
- Cut-off frequency of the filter equals to 13.5 rad/s;
- PID controller: $k_p = 1$, $k_i = 0.01$, $k_d = 0.01$;
- 2 kg of payload;
- No gravity;
- $q_0 = 0^\circ$;
- PI parameters Decentralized Position Control: $k_p = 30$, $k_i = 0.25$.
- PI parameters Decentralized Velocity Control: $k_p = 5000$, $k_i = 100$.

The results of this simulation are similar to the previous one. In this simulation is used a square wave in order to see the effect of this control both for rise and fall phase of the angular position input.

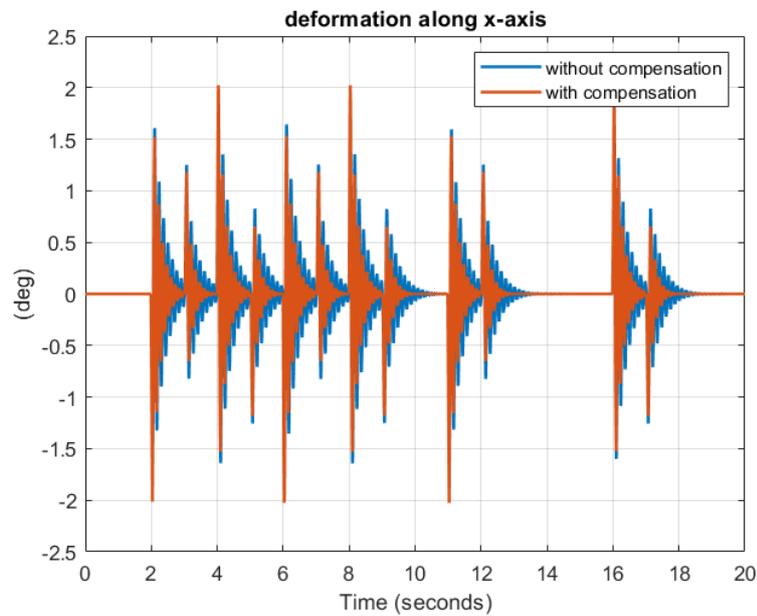


Figure 57 Deformation along x-axis with and without compensation

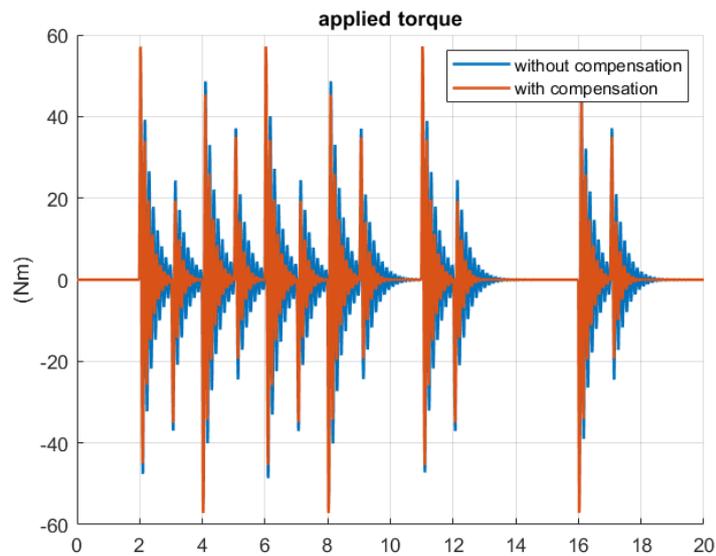


Figure 58 Torque applied to the joint with and without compensation

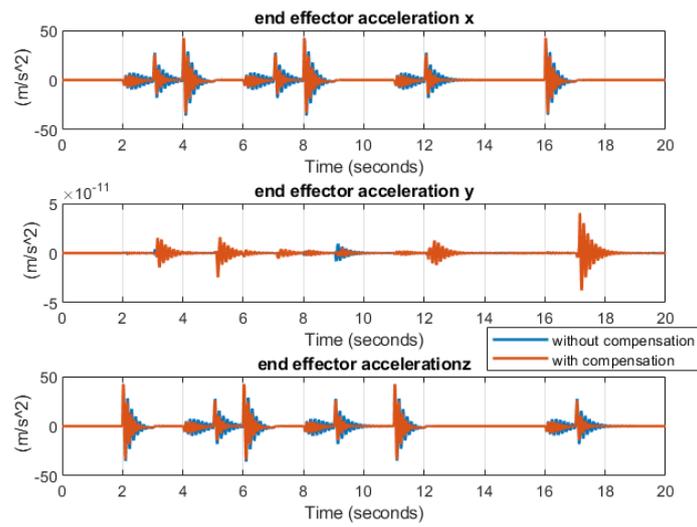


Figure 59 End effector accelerations with and without compensation

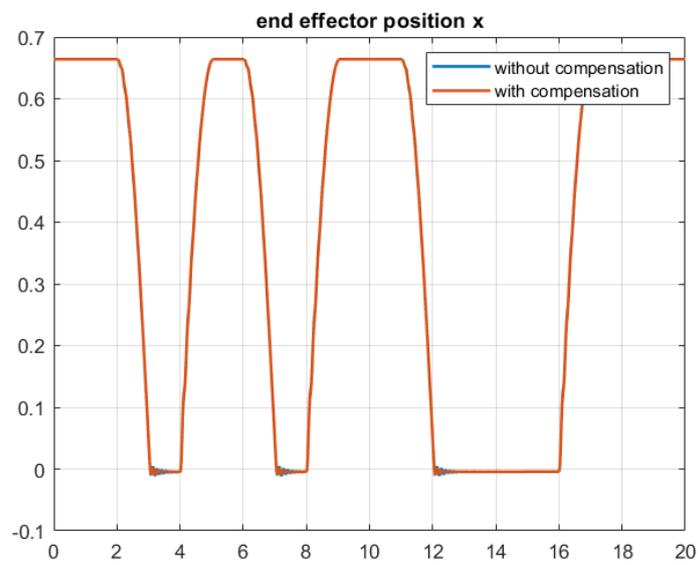


Figure 60 End effector position along x-axis with and without compensation

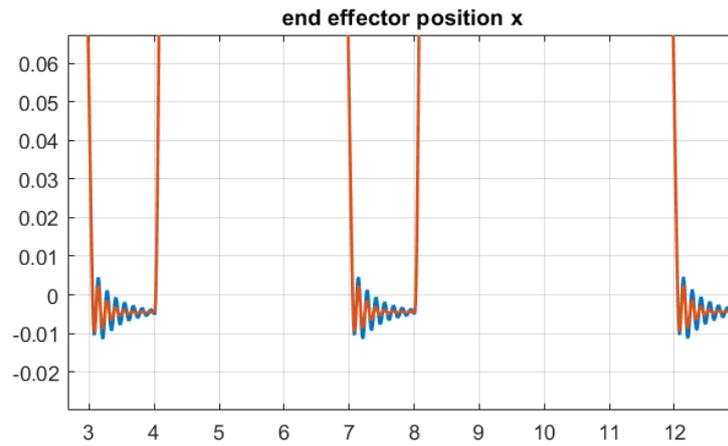


Figure 61 Zoom of end effector position along x-axis

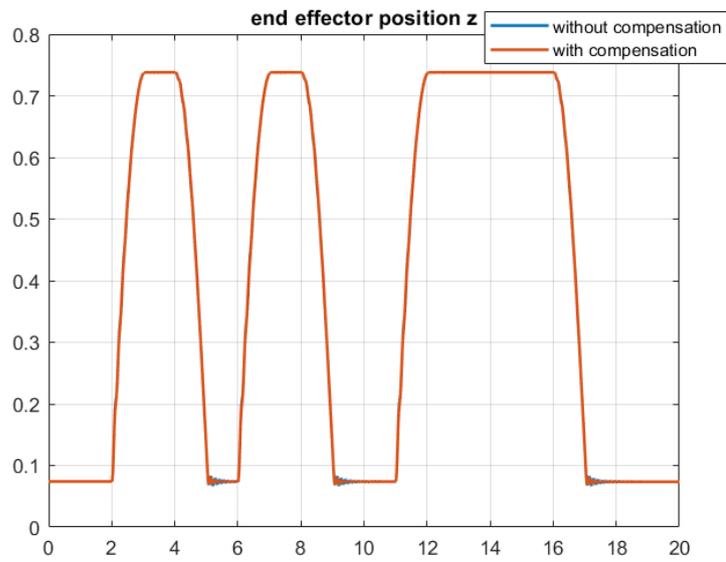


Figure 62 End effector position along z-axis with and without compensation

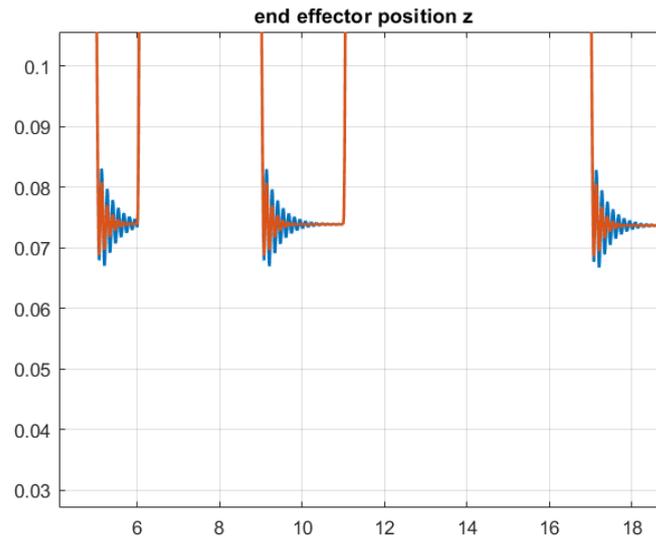


Figure 63 Zoom of end effector position along z-axis with and without compensation

7.1.2 Simulations with gravity

Now, let's add the gravity in order to test this controlled scheme also in the presence of gravity and the same parameters of the previous simulations are used.

The third simulation is done with these parameters:

- The input reference position is a step with a delay of 2 seconds and $\pi/2$ of amplitude;
- Cut-off frequency of the filter equals to 13.5 rad/s;
- PID controller: $k_p = 70$, $k_i = 5$, $k_d = 0.1$;
- 2 kg of payload;
- With gravity;
- $q_0 = 0^\circ$;
- PI parameters Decentralized Position Control: $k_p = 30$, $k_i = 0.25$.
- PI parameters Decentralized Velocity Control: $k_p = 5000$, $k_i = 100$.

Furthermore an equivalent system made up of an equivalent rigid link (with the same parameters) is compared with the pseudo-rigid body link.

The deformation occurring along x-axis of the virtual joint are more damped but this value at steady state is different from zero since there is the gravity that affects the performance (Figure 64). As a consequence of this, the end effector position along z-axis is different with respect to a rigid equivalent system since there are deformations resulting in an offset (Figures 68 and 69). Furthermore, with the additional control input aiming to vibration damping control the end effector position presents less oscillations (Figures 66,67 68 and 69) and the end effector accelerations are also more damped.

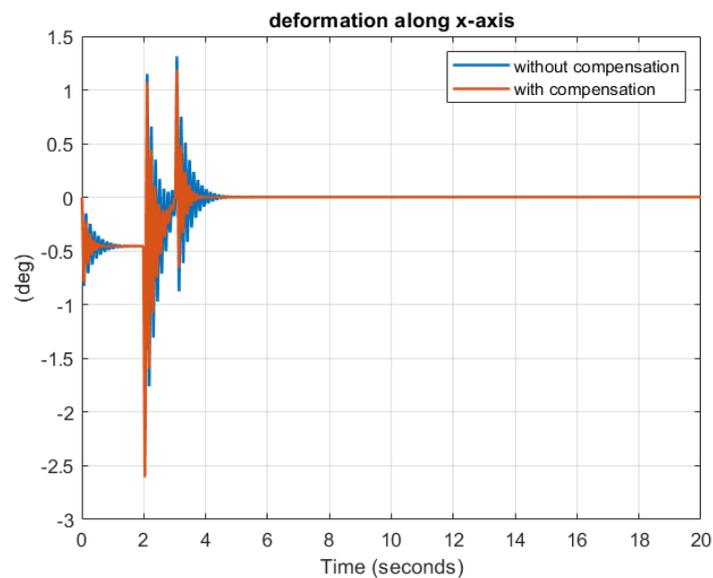


Figure 64 Deformation along x-axis with and without compensation

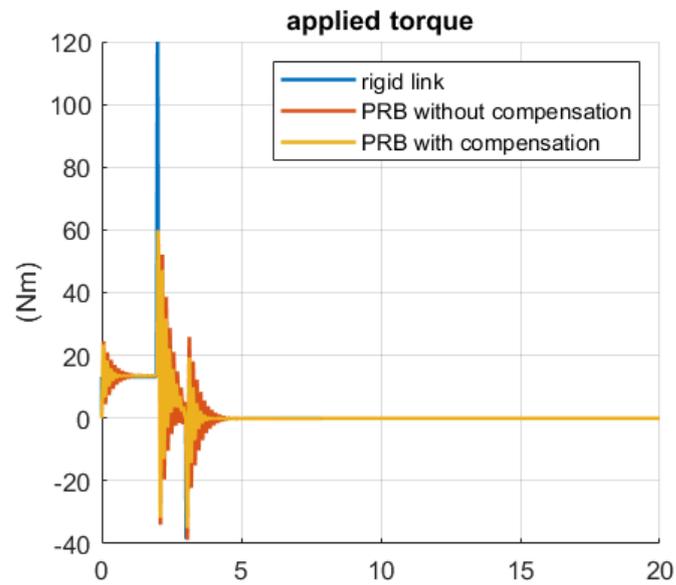


Figure 65 Torque applied to the joint with and without compensation PRB and rigid robot

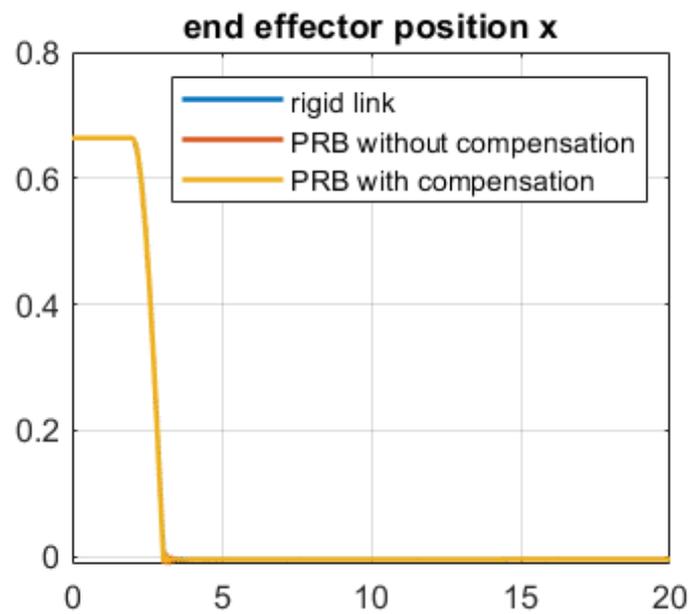


Figure 66 End effector position along x-axis with and without compensation PRB and rigid robot

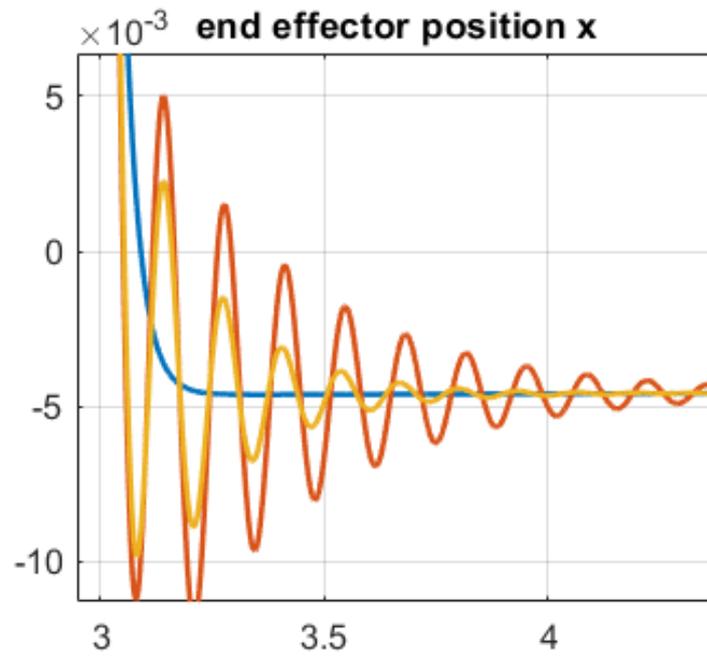


Figure 67 Zoom of end effector position along x-axis

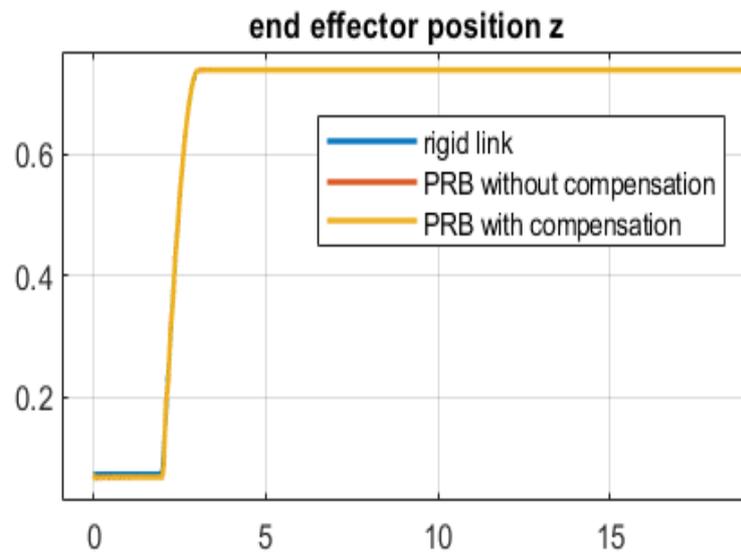


Figure 68 End effector position along z-axis with and without compensation and rigid robot

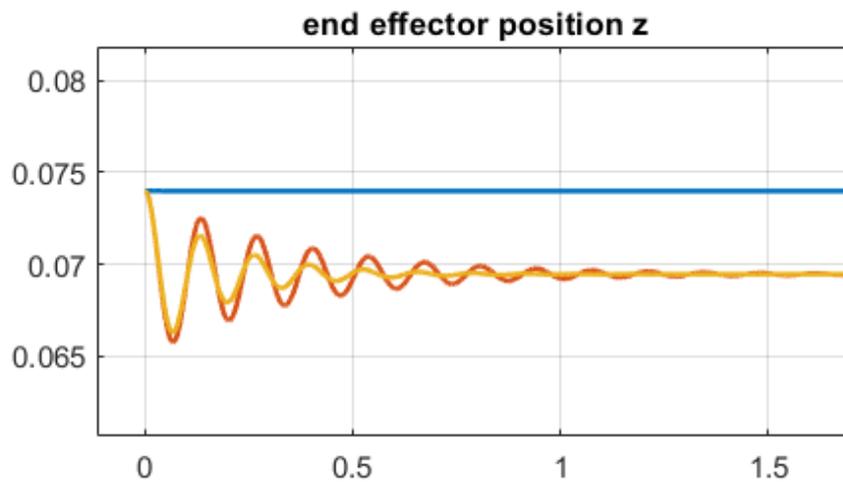


Figure 69 Zoom of end effector position along z-axis

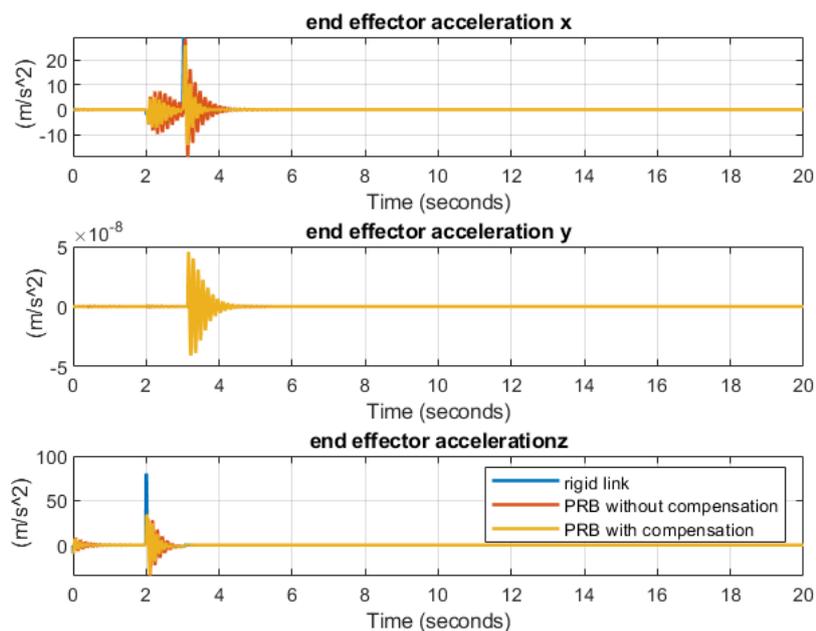


Figure 70 End effector accelerations with and without compensation PRBM and rigid robot

The fourth simulation is done with these parameters:

- The input reference position is a square wave;
- Cut-off frequency of the filter equals to 13.5 rad/s;
- PID controller: $k_p = 70$, $k_i = 5$, $k_d = 0.1$;

- 2 kg of payload;
- With gravity;
- $q_0 = 0^\circ$;
- PI parameters Decentralized Position Control: $k_p = 30$, $k_i = 0.25$.
- PI parameters Decentralized Velocity Control: $k_p = 5000$, $k_i = 100$.

Furthermore an equivalent system made up of an equivalent rigid link (with the same parameters) is compared with the pseudo-rigid body link.

In this case the results are equal to the previous simulation.

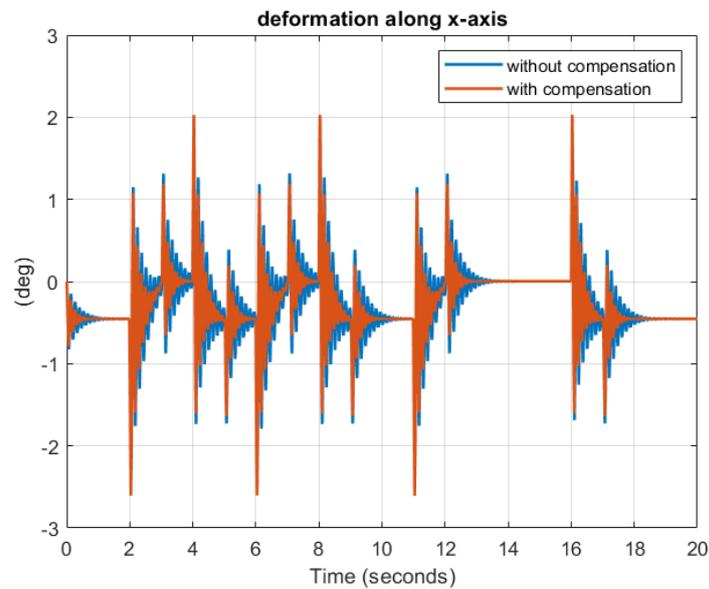


Figure 71 Deformation along x-axis with and without compensation PRBM and rigid robot

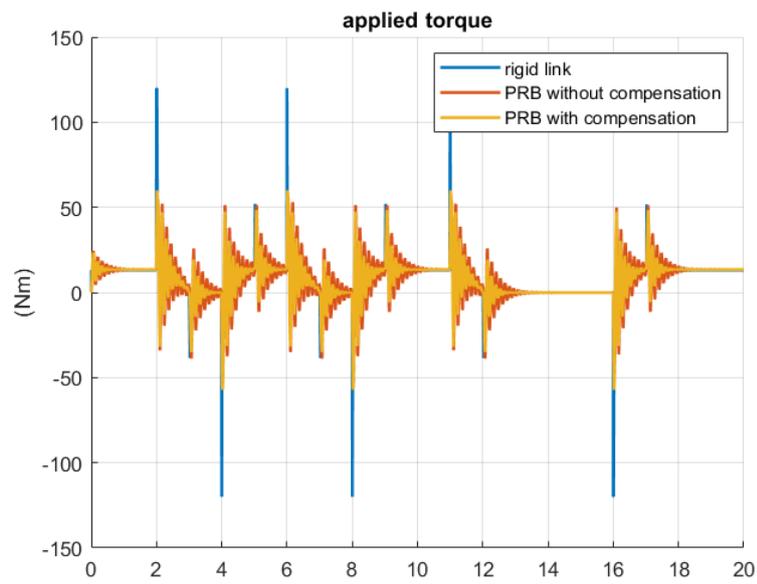


Figure 72 Torque applied to the joint with and without compensation PRBM and rigid robot

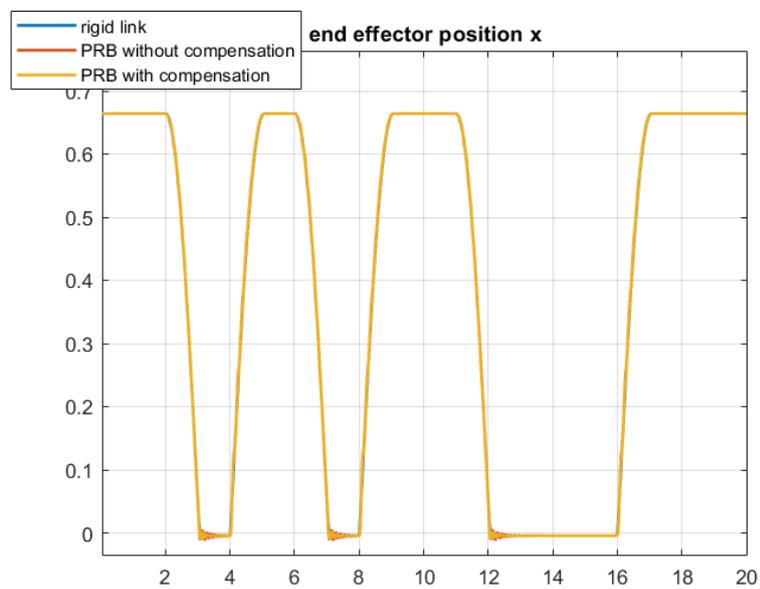


Figure 73 End effector position along x-axis with and without compensation PRBM and rigid robot

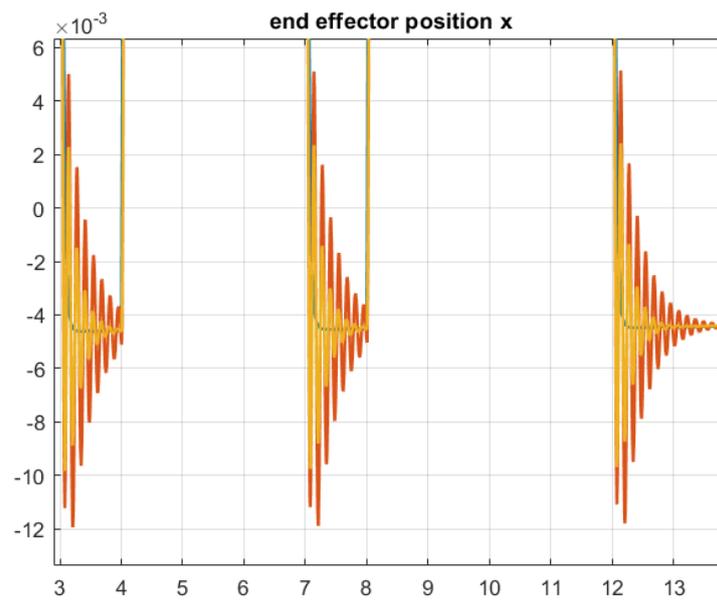


Figure 74 Zoom of end effector position along x-axis

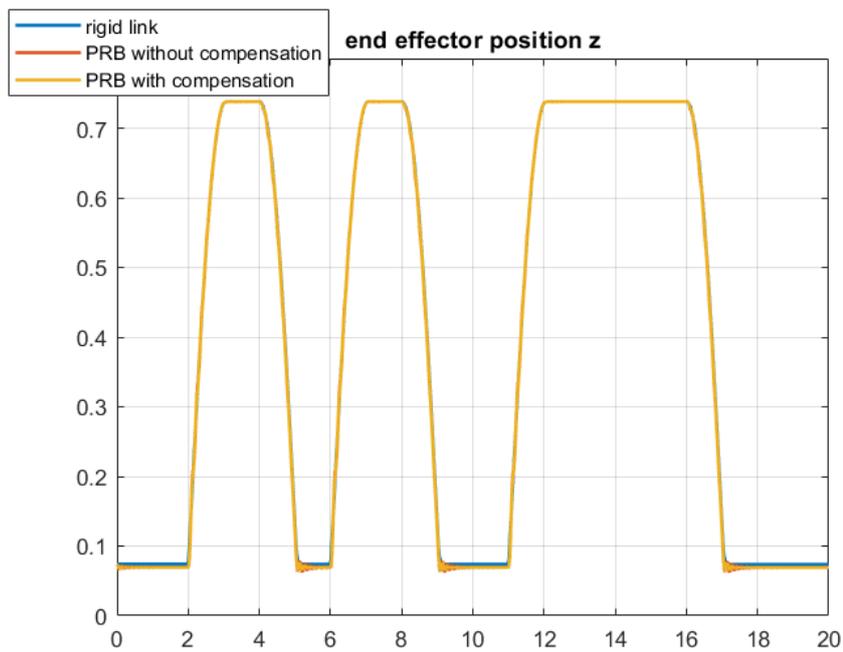


Figure 75 End effector position along z-axis with and without compensation PRBM and rigid robot

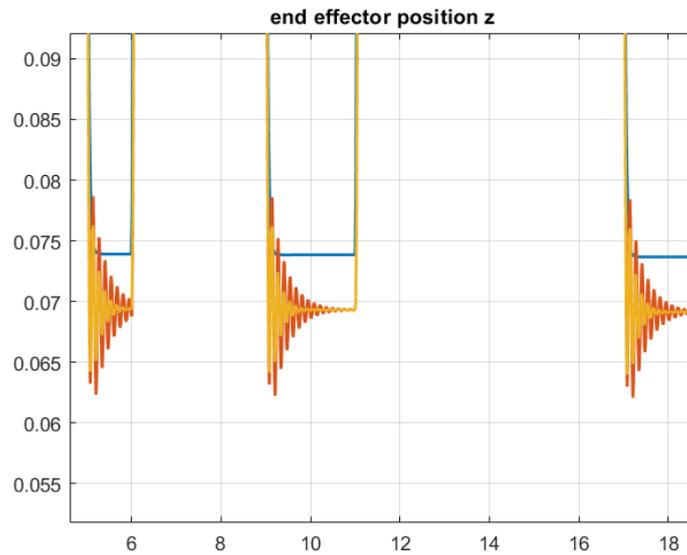


Figure 76 Zoom of end effector position along z-axis

7.2 Vibration damping control applied to the entire system

The Simulink controlled scheme of the entire system is shown in the figure 77. This control scheme is explained before for the simplified system. In the whole system 3 PID control actions are sent to the Decentralized Velocity block, for the first motoreducer the difference of relative acceleration of the links along y-axis is sent to the PID controller and it is used as additional control input. The difference is taken, since the y-axes of the two links are oriented in an opposite direction.

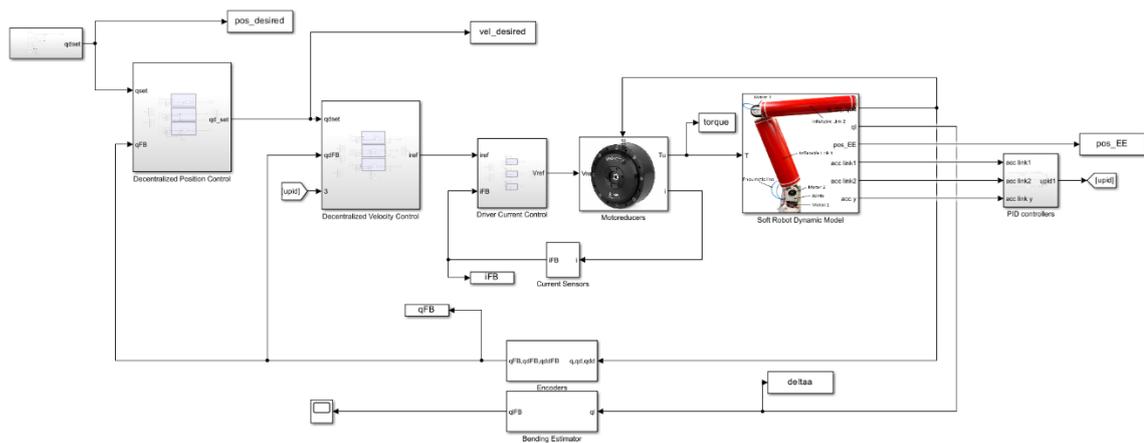


Figure 77 POPUP control scheme for the entire system

7.2.1 Simulations without gravity

The fifth simulation is done with these parameters:

- The input reference position is a step with a delay of 2 seconds and $\pi/2$ of amplitude only in the third joint;
- Cut-off frequency of the filter equals to 13.5 rad/s;
- PID controllers: $k_p = 70$;
- 2 kg of payload;
- With gravity;
- $q_0 = [0^\circ, 90^\circ, 0^\circ]$;
- PI parameters Decentralized Position Control: $k_p = 30$, $k_i = 0.25$.
- PI parameters Decentralized Velocity Control: $k_p = 5000$, $k_i = 100$.

It's possible to notice that applying an input only in the third motoreducer both links are subjected to deformations, and this is due to the coupling effects. For what regards the advantages of the additional control input it's possible to see a decreasing in the deformations along the links (Figure 78, 79, 80 and 81), a reduction in end effector accelerations (Figure 82) and applied torques (Figure 83). The most important result is a more damped end effector position since less oscillations are present (Figure 84, 85, 86 and 87).

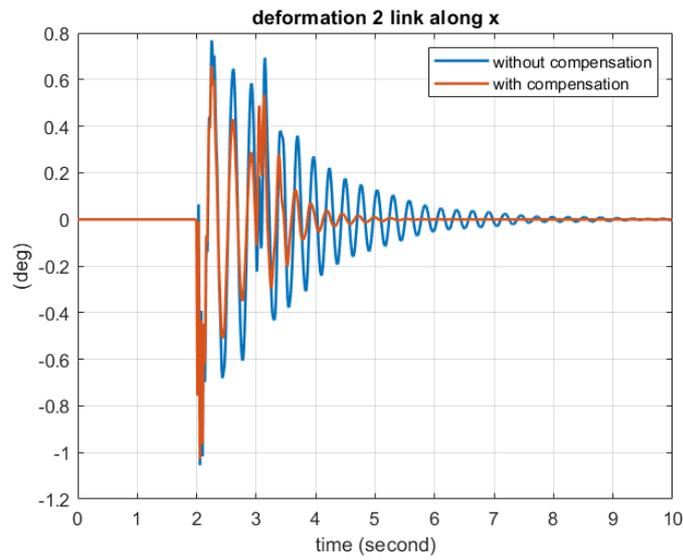


Figure 78 Deformation link 2 along x-axis with and without compensation

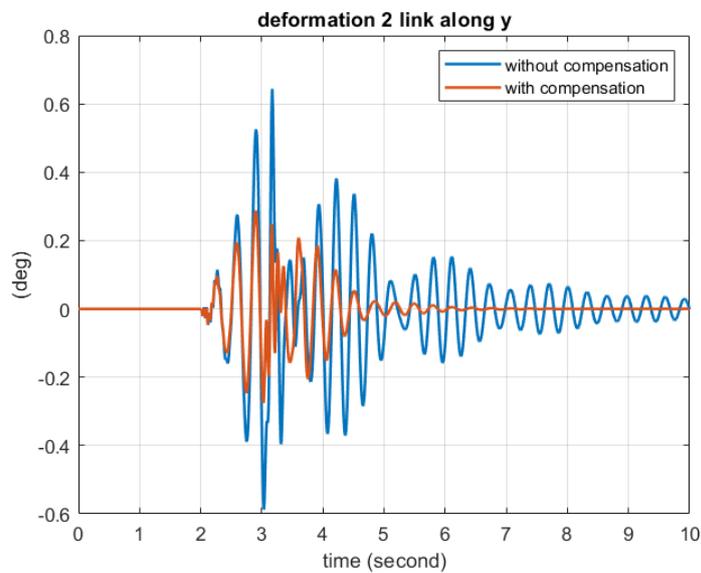


Figure 79 Deformation link 2 along y-axis with and without compensation

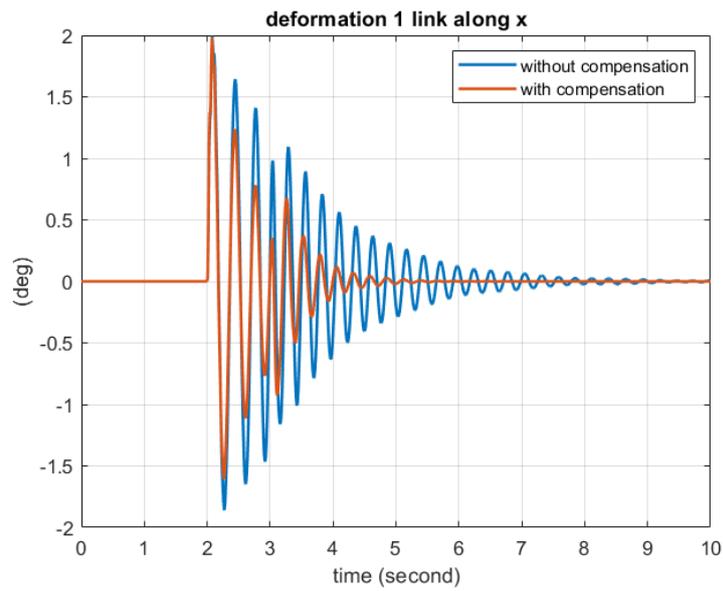


Figure 80 Deformation link 1 along x-axis

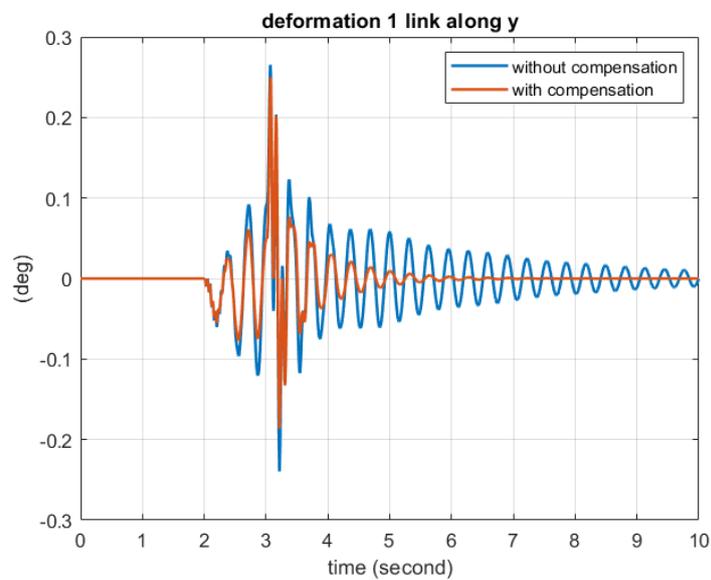


Figure 81 Deformation link 1 along y-axis

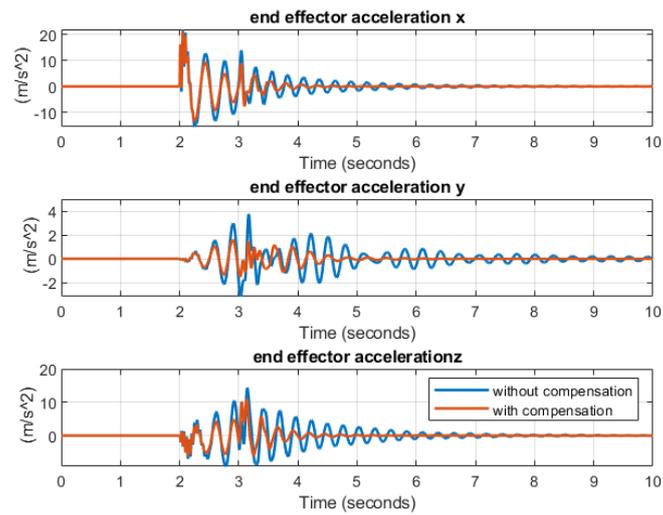


Figure 82 End effector accelerations with and without compensation

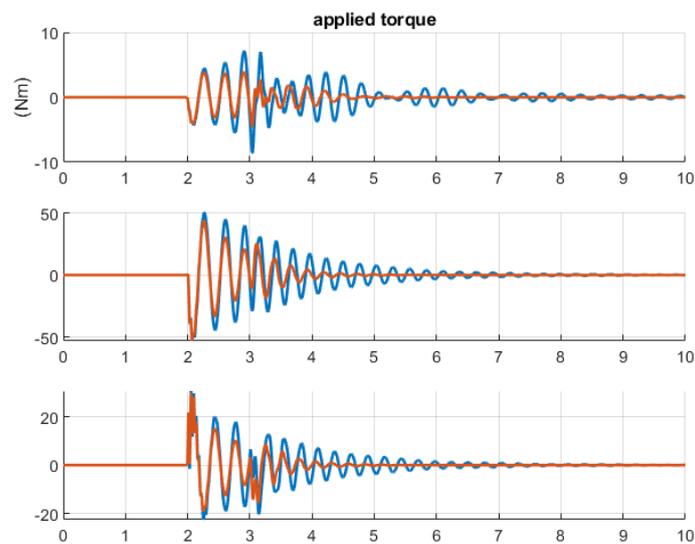


Figure 83 Torques applied to the joints with and without compensation

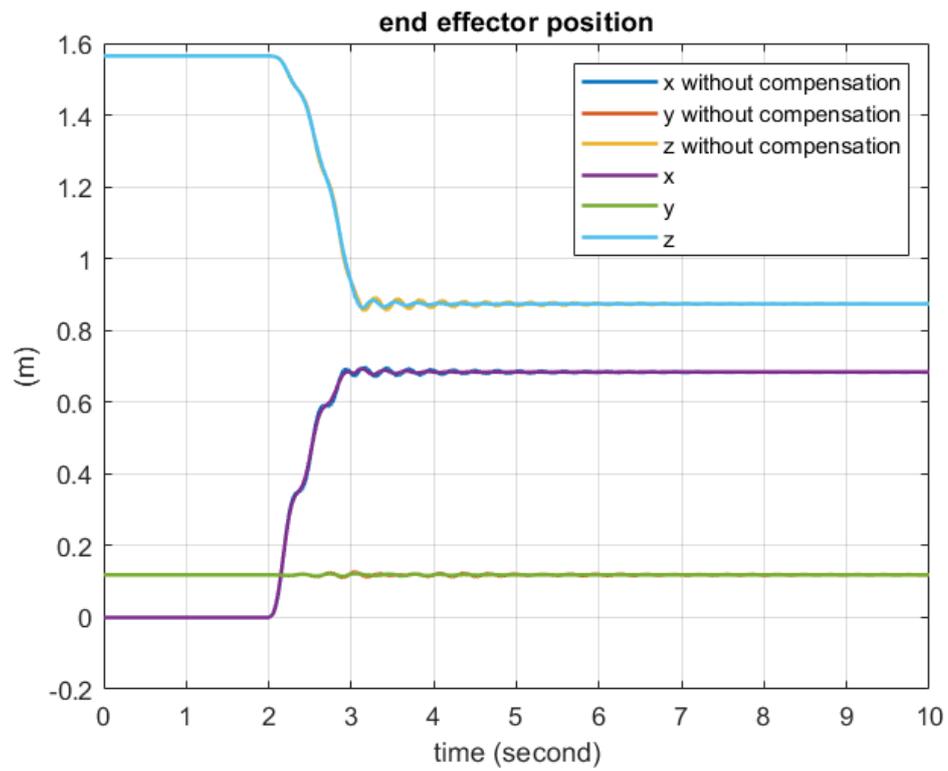


Figure 84 End effector position with and without compensation

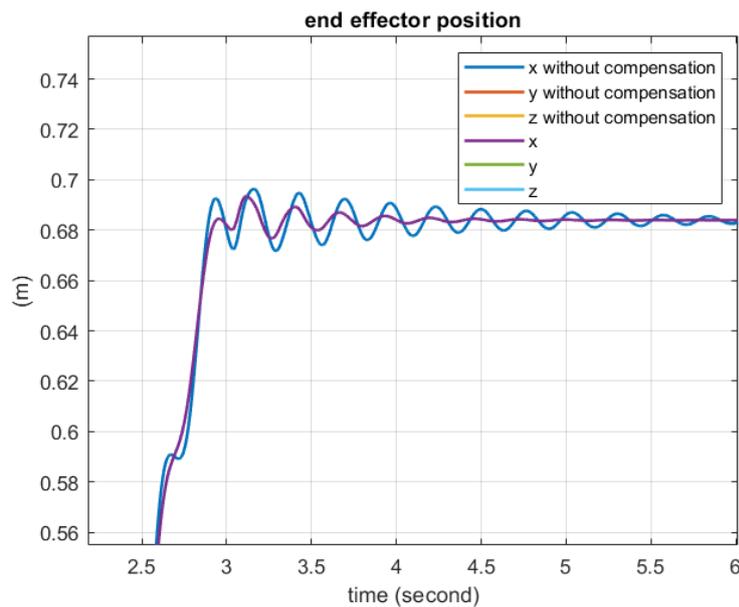


Figure 85 Zoom of end effector position along x-axis

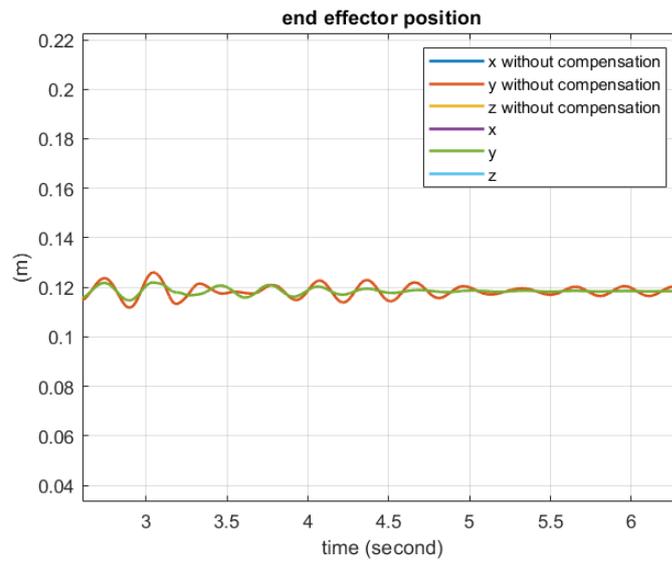


Figure 86 Zoom of end effector position along y-axis

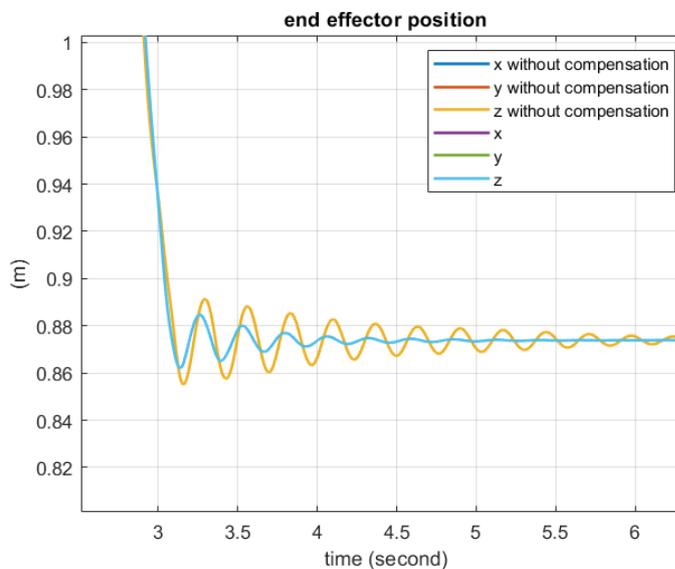


Figure 87 Zoom of end effector position along z-axis

The sixth simulation is done with these parameters:

- The input reference position is a step with a delay of 2 seconds and $\pi/4$ of amplitude in the second and third joint;
- Cut-off frequency of the filter equals to 13.5 rad/s;
- PID controllers: $k_p = 70$; $k_i = 100$; $k_d = 20$;

- 2 kg of payload;
- With gravity;
- $q_0 = [0^\circ, 90^\circ, 0^\circ]$;
- PI parameters Decentralized Position Control: $k_p = 30$, $k_i = 0.25$.
- PI parameters Decentralized Velocity Control: $k_p = 5000$, $k_i = 100$.

Adding a reference position input also in the second motoreducer the results are the same as before, so the deformations along links are more damped (Figures 88, 89, 90 and 91) the torques applied to the links and the end effector accelerations are decreased (Figures 96 and 97 respectively) and the end effector positions present less oscillations (Figures 92, 93, 94 and 95)

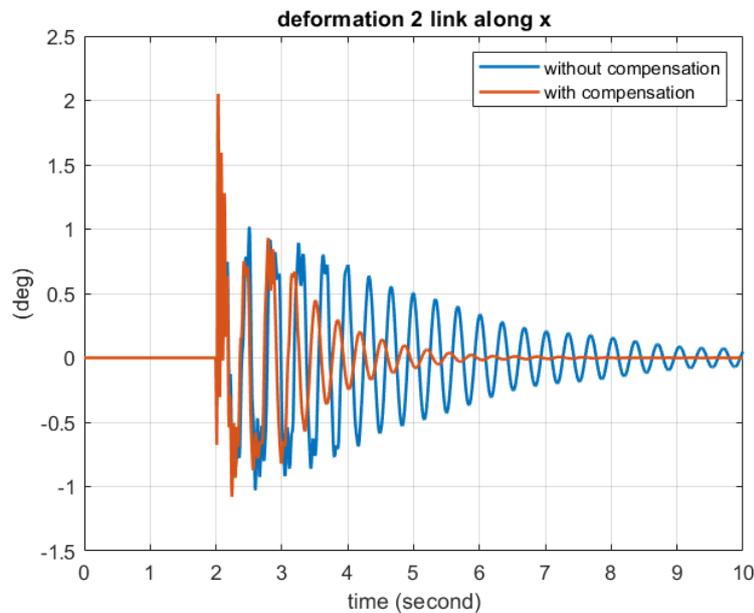


Figure 88 Deformation link 2 along x-axis with and without compensation

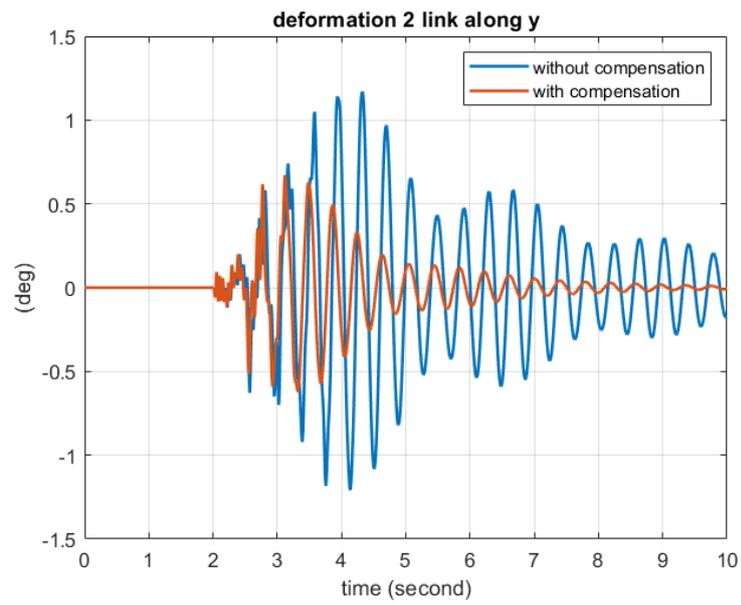


Figure 89 Deformation link 2 along y-axis with and without compensation

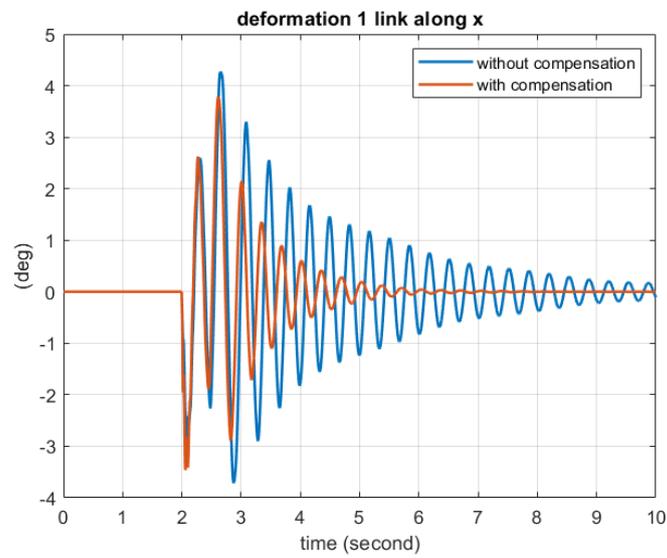


Figure 90 Deformation link 1 along x-axis with and without compensation

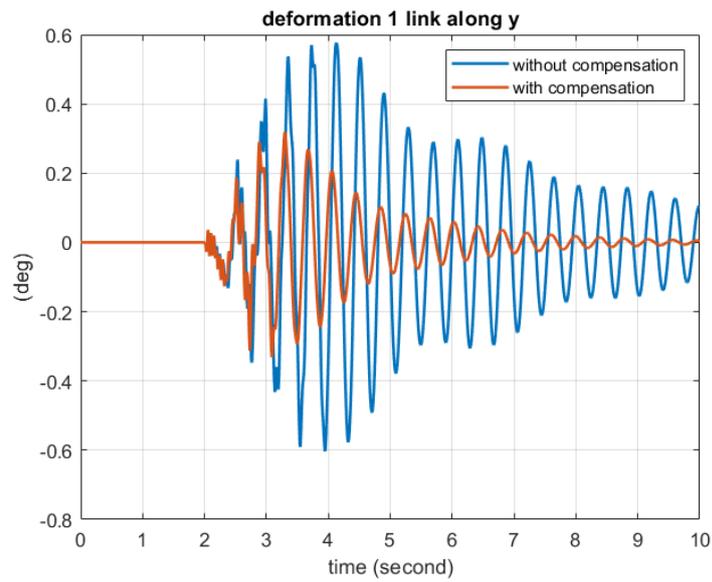


Figure 91 Deformation link 1 along y-axis with and without compensation

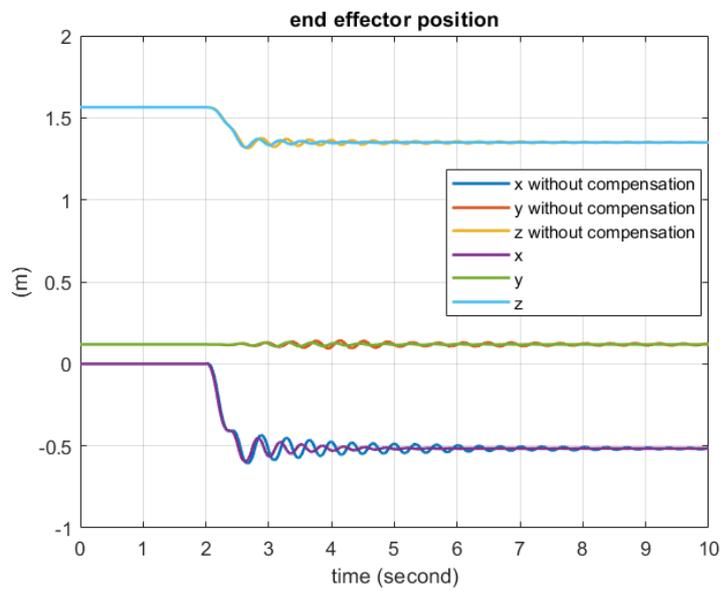


Figure 92 End effector position with and without compensation

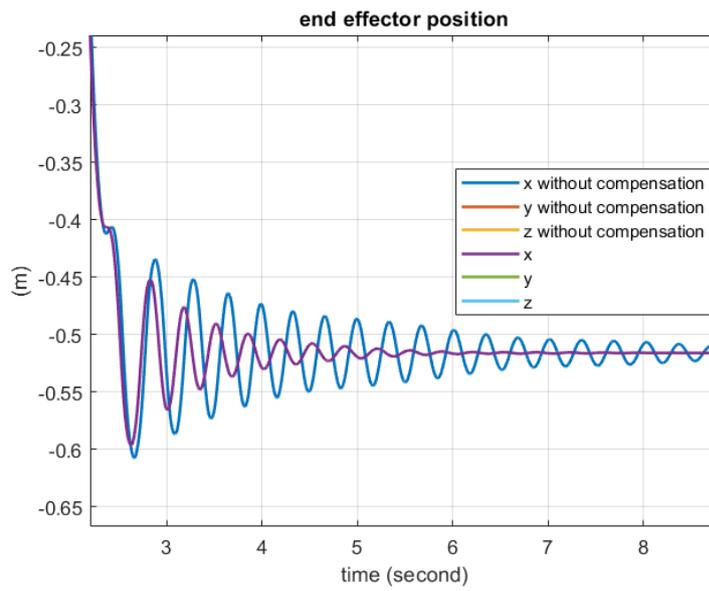


Figure 93 Zoom of end effector position along x-axis

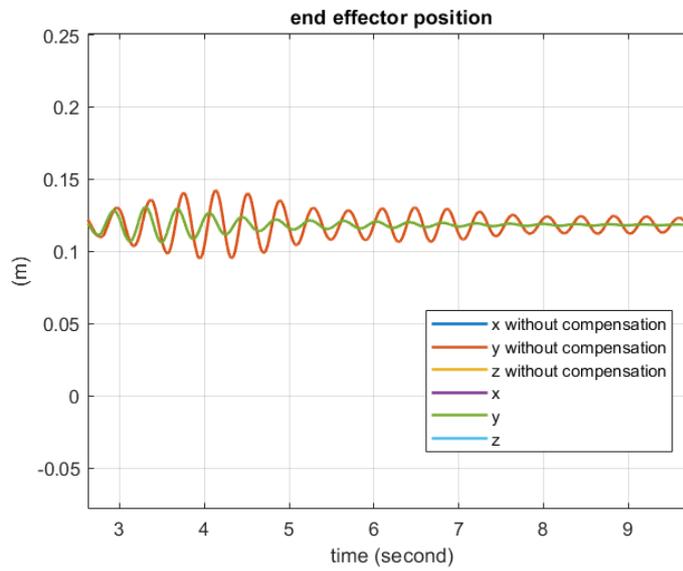


Figure 94 Zoom of end effector position along y-axis

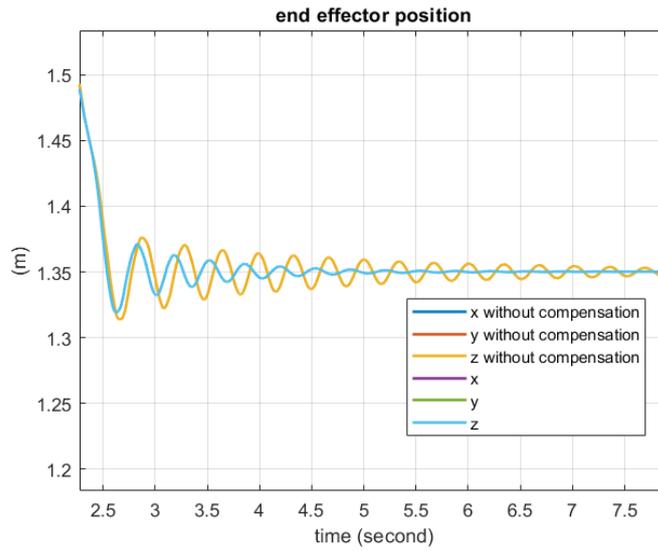


Figure 95 Zoom of end effector position along z-axis

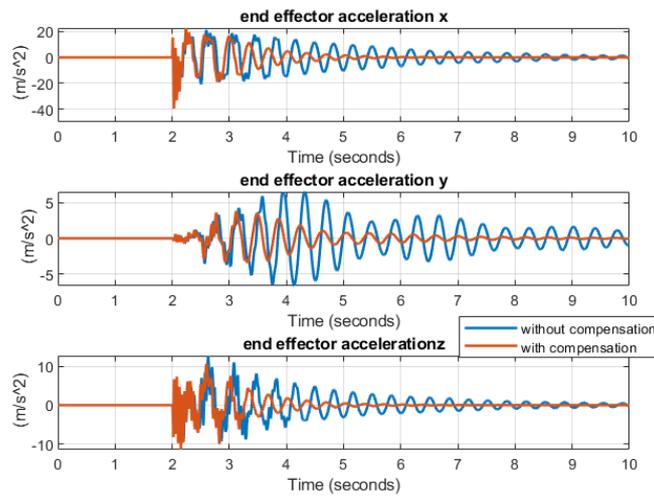


Figure 96 End effector accelerations with and without compensation

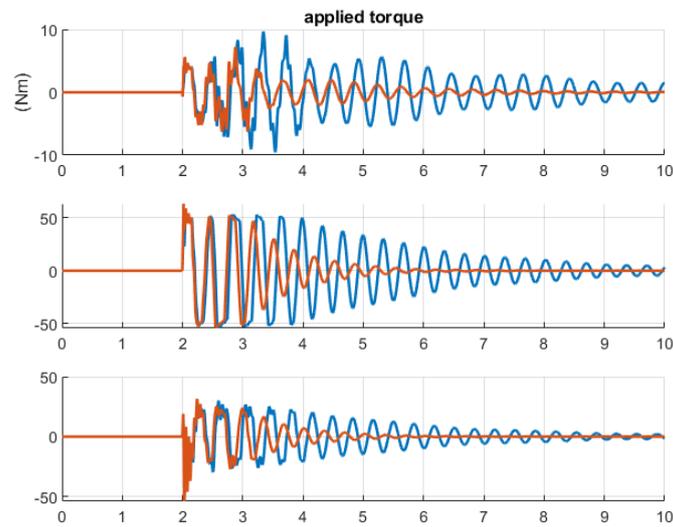


Figure 97 Torques applied to the joints with and without compensation

7.2.2 Simulations with gravity

The seventh simulation is done with these parameters:

- The input reference position is a step with a delay of 2 seconds and $\pi/4$ of amplitude in the second and third joint;
- Cut-off frequency of the filter equals to 13.5 rad/s;
- PID controllers: $k_p = 70$; $k_p = 60$; $k_p = 20$;
- 2 kg of payload;
- With gravity;
- $q_0 = [0^\circ, 90^\circ, 0^\circ]$;
- PI parameters Decentralized Position Control: $k_p = 30$, $k_i = 0.25$.
- PI parameters Decentralized Velocity Control: $k_p = 5000$, $k_i = 100$.

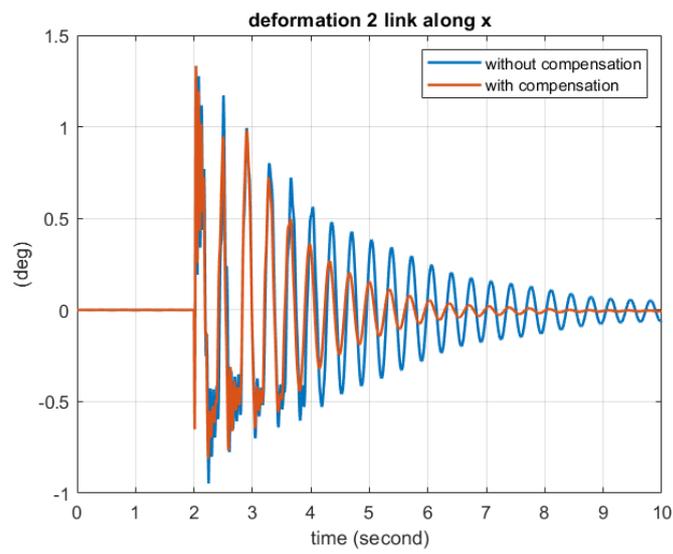


Figure 98 Deformation link 2 along x-axis with and without compensation

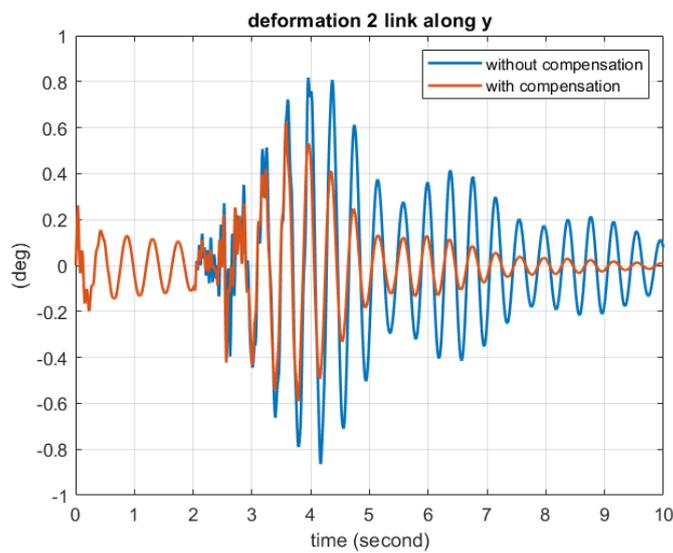


Figure 99 Deformation link 2 along y-axis with and without compensation

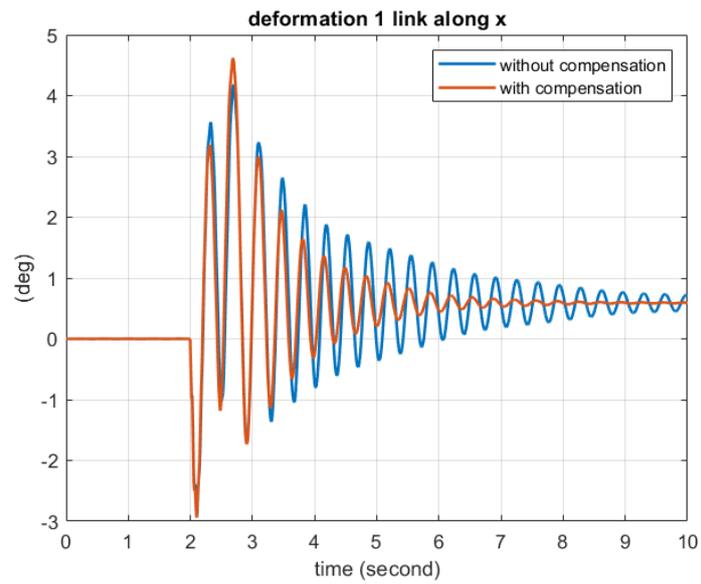


Figure 100 Deformation link 1 along x-axis with and without compensation

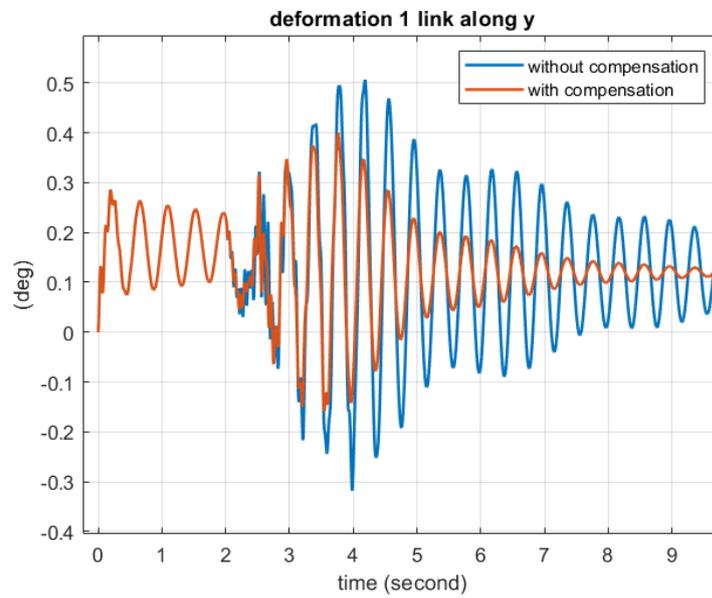


Figure 101 Deformation link 1 along y-axis with and without compensation

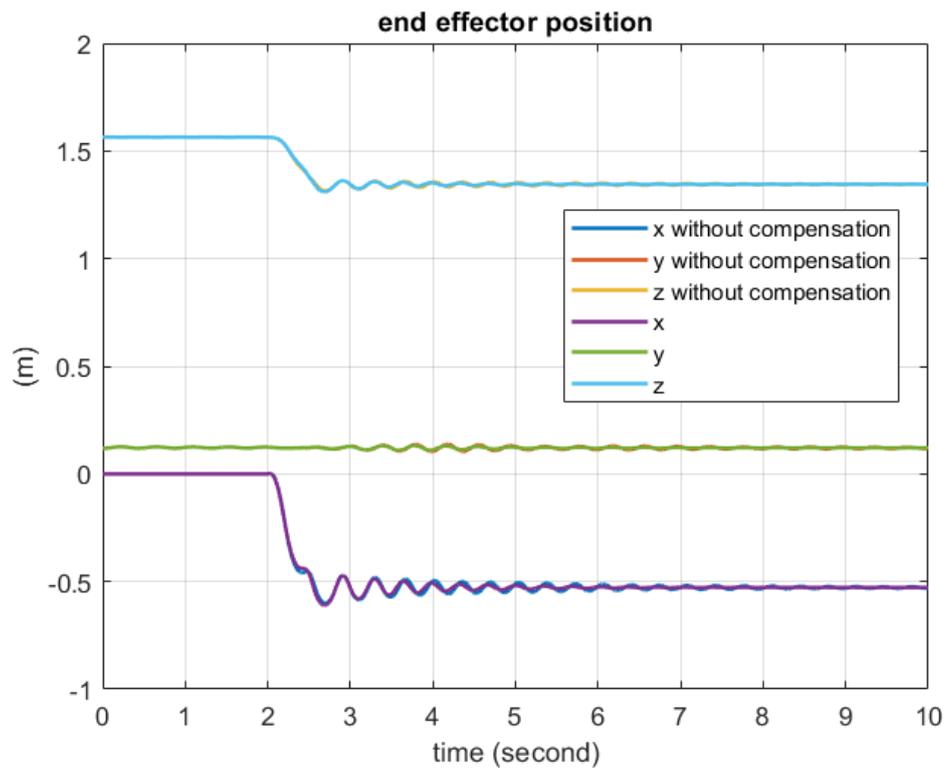


Figure 102 End effector positions with and without compensation

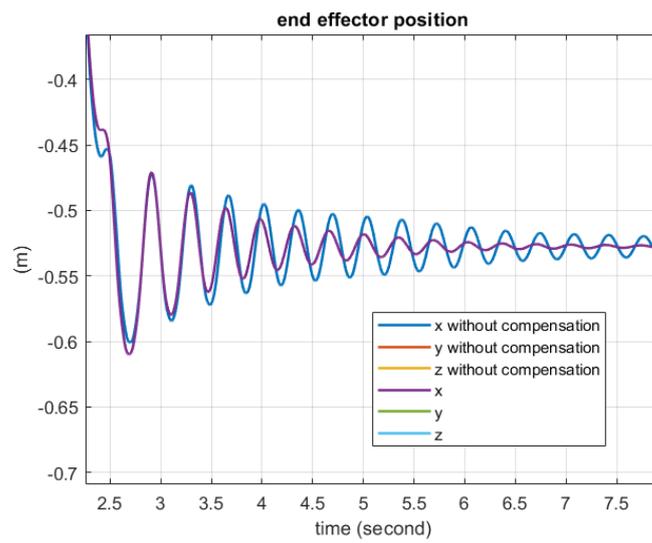


Figure 103 Zoom of end effector position along x-axis

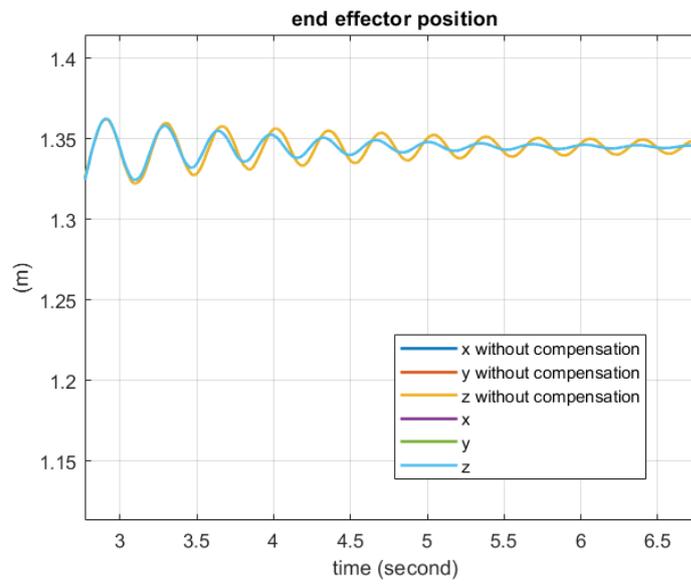


Figure 104 Zoom of end effector position along z-axis

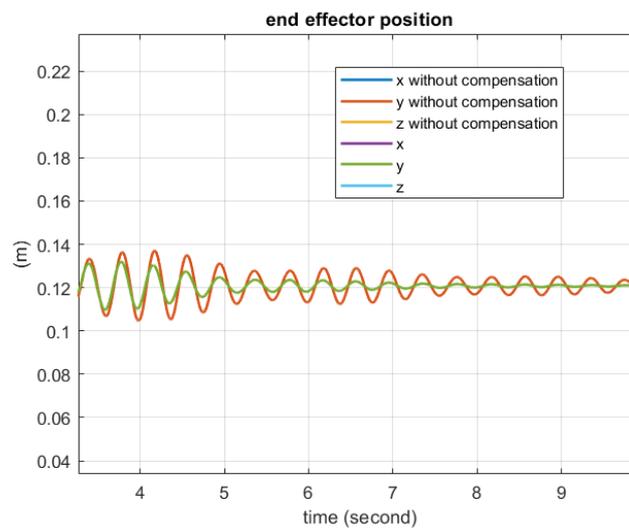


Figure 105 Zoom of end effector position along y-axis

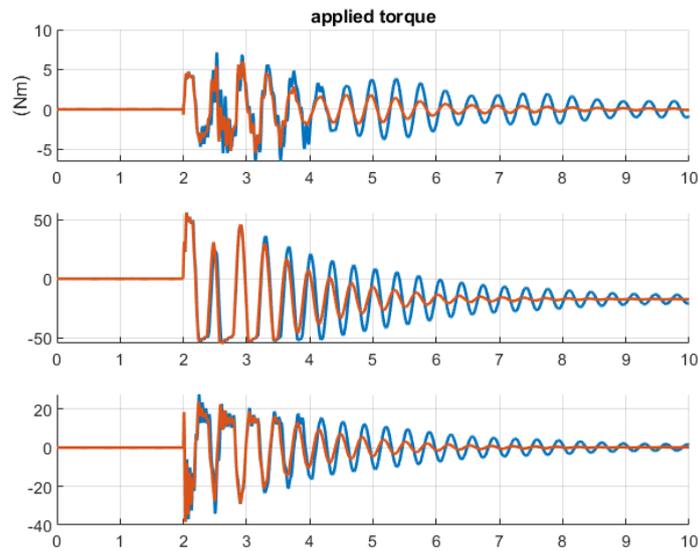


Figure 106 Torques applied to the joints with and without compensation

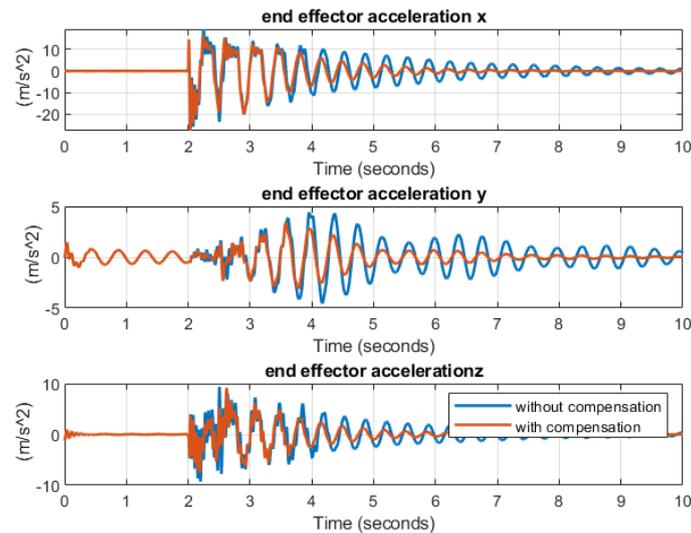


Figure 107 End effector accelerations with and without compensation

The results of the simulation in presence of gravity are almost the same because there are less oscillations in end effector positions (Figures 102, 103, 104 and 105) and the deformations along the links are decreased but there is an offset in the deformations occurring along the links due to gravity that implies a final value different from zero.

Chapter 8

Differential kinematic control with vibration damping control

In the previous chapter a known input signal is provided in the joint space. Since the specification are often expressed in the operational space, let's apply the control scheme previously discussed in a control scheme where a target position must be reached. In order to do that, a differential kinematics approach is applied, and the end effector velocities are translated in joint angular velocities through the transpose of the Jacobian matrix of the 3 DOF representation (rigid equivalent robotic arm). This control scheme is shown in figure 108.

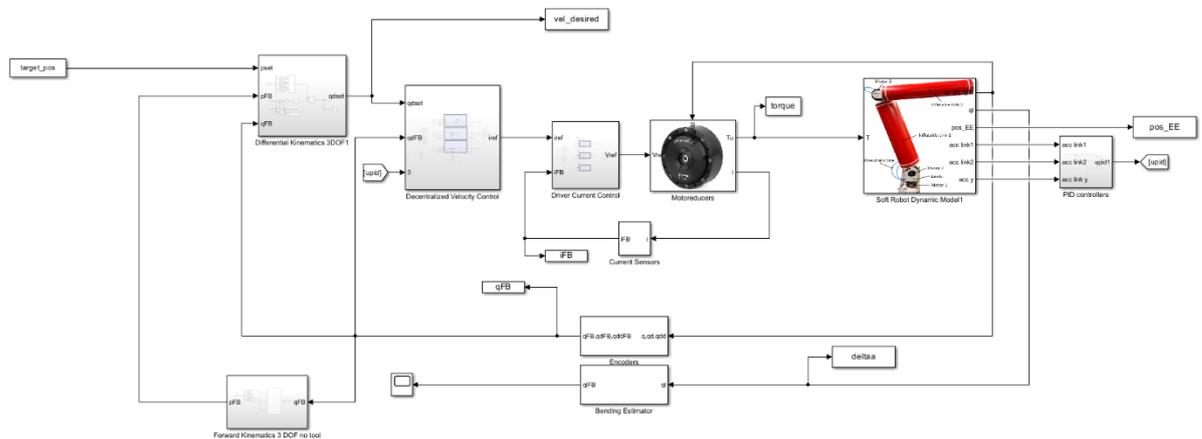


Figure 108 POPUOP control scheme with differential kinematics and vibration damping

In this control scheme, the position of the end effector is derived exploiting the forward kinematics of the 3 DOF (rigid equivalent robotic arm). This model doesn't account for the deflection along the links resulting in a final error even if with the

additional vibration damping control the angular displacement of virtual joints decrease.

8.1 Simulations

The first simulation is done with these parameters:

- Target position = $[-0.8 \ -0.8 \ 0.9]$ m;
- Cut-off frequency of the filter equals to 13.5 rad/s;
- PID controller: $k_p = 0; k_p = 50; k_p = 50$;
- 2 kg of payload;
- Without gravity;
- $q_0 = [0^\circ \ 60^\circ \ -120^\circ]$;
- PI parameters Decentralized Velocity Control: $k_p = 1000, k_i = 10$.

In this simulation, only a compensation in the third and second motoreducer is applied and the results are a more damped deformations along x-axis of the virtual joints (Figures 109 and 111), the deformations occurring along y-axis of the virtual joints are almost the same since no additional control input is applied (Figures 110 and 112). The end effector positions present less oscillations only along x and z axis while along y-axis is almost the same (Figures 113, 114 and 115).

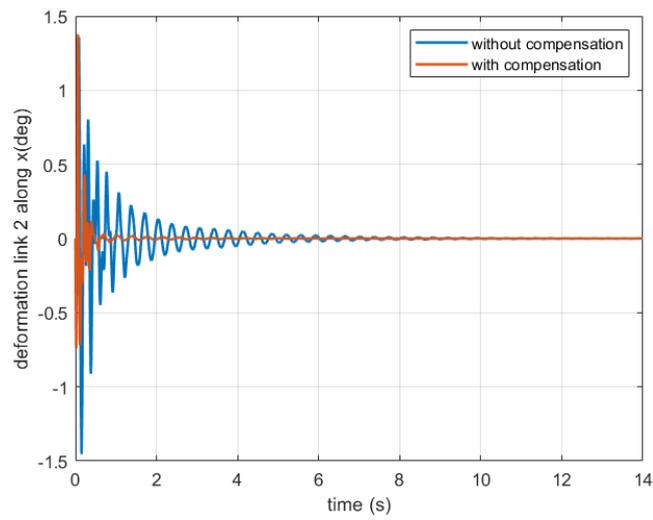


Figure 109 Deformation link 2 along x-axis with and without compensation

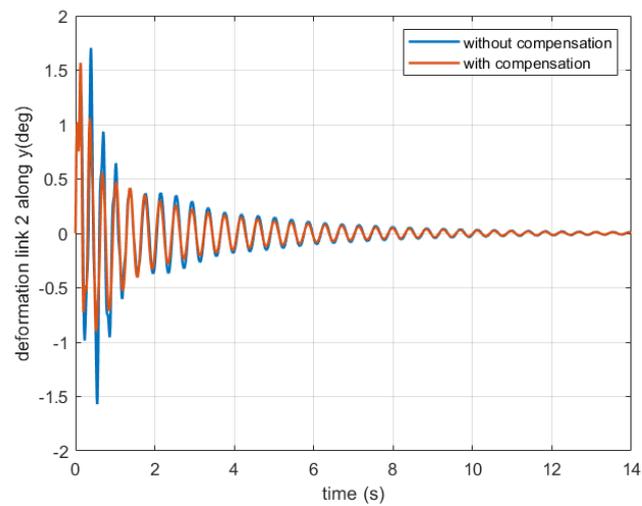


Figure 110 Deformation link 2 along y-axis with and without compensation

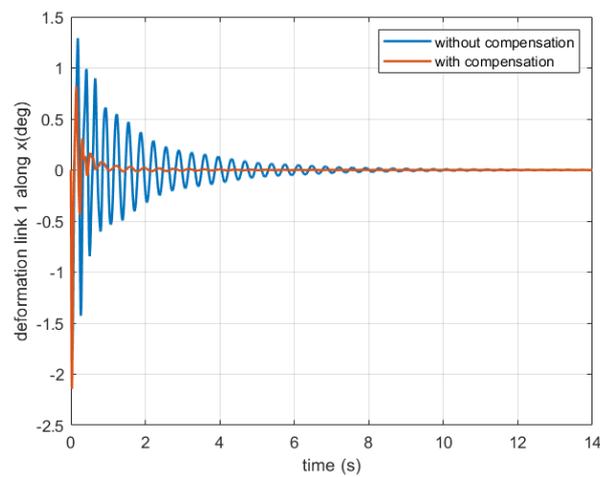


Figure 111 Deformation link 1 along x-axis with and without compensation

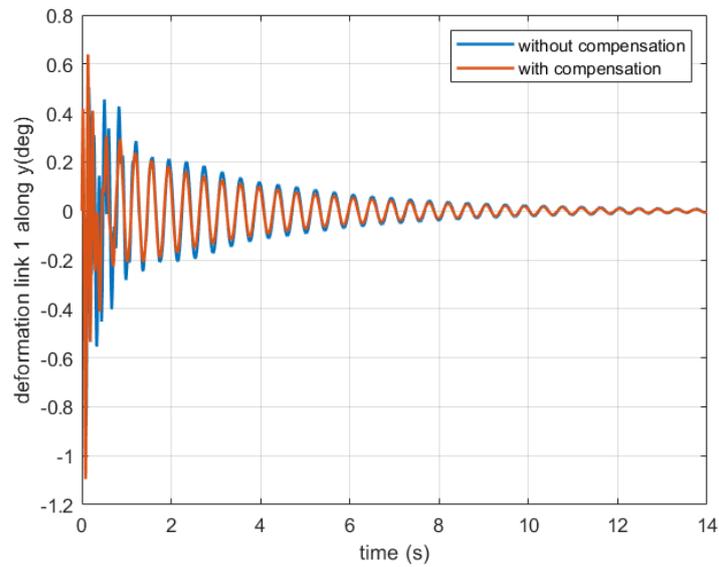


Figure 112 Deformation link 1 along y-axis with and without compensation

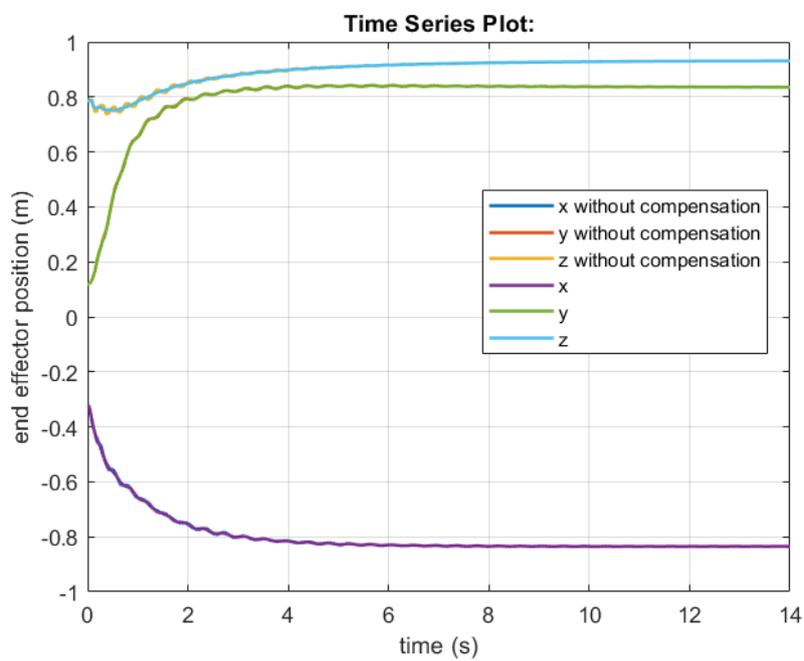


Figure 113 End effector position with and without compensation

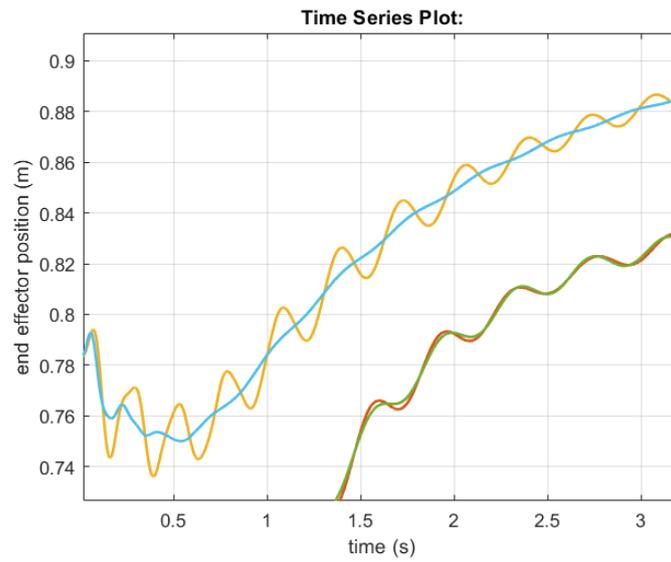


Figure 114 Zoom of end effector positions along x and y axes

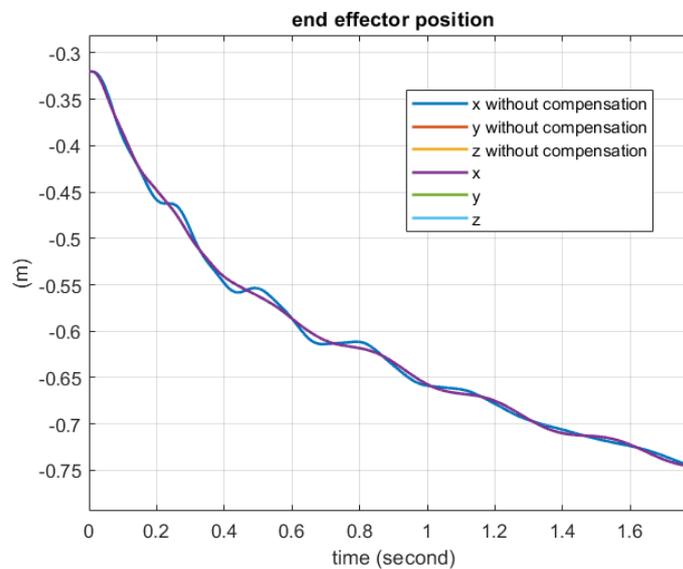


Figure 115 Zoom of end effector positions along z-axis

For what regards the applied torques only the ones applied to the second and third motoreducers are decreased (Figure 116) and the end effector accelerations are decreased too (Figure 117).

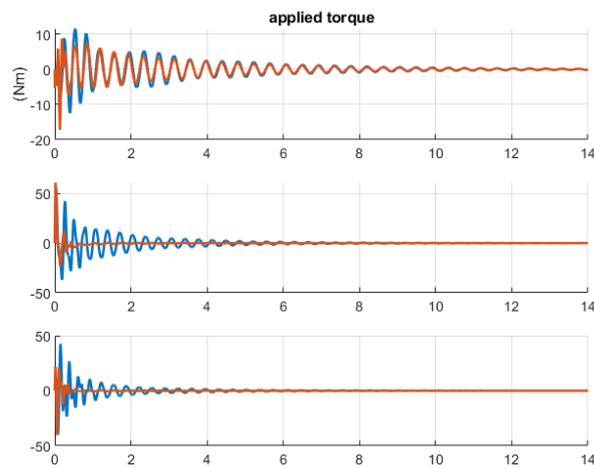


Figure 116 Torques applied to the joints with and without compensation

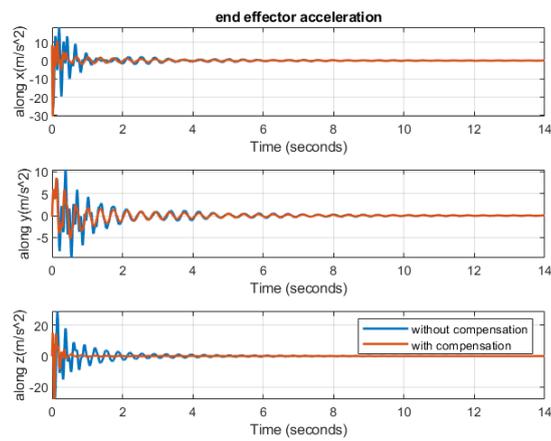


Figure 117 End effector accelerations with and without compensation

Figure 118 shows the end effector trajectory, and the results of this additional control input is a more damped trajectory.

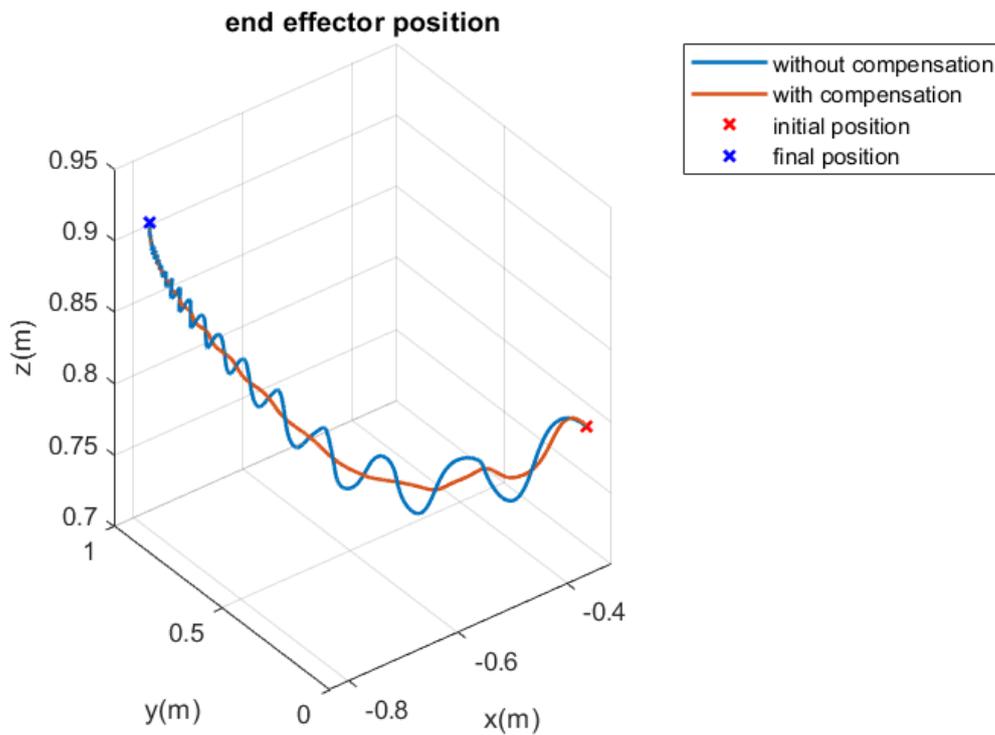


Figure 118 End effector trajectory with and without compensation

Let's add a compensation also in the first motoreducer in order to obtain less oscillations in the end effector position also in y direction.

The second simulation is done with these parameters:

- Target position = $[-0.8 \ -0.8 \ 0.9]$ m;
- Cut-off frequency of the filter equals to 13.5 rad/s;
- PID controller: $k_p = 30; k_p = 60; k_p = 70, k_i = k_d = 0.1$;
- 2 kg of payload;
- Without gravity;
- $q_0 = [0^\circ \ 60^\circ \ -120^\circ]$;
- PI parameters Decentralized Velocity Control: $k_p = 1000, k_i = 10$.

The result of adding an additional control input in the first joint that accounts for the relative acceleration of both link along y -axis is a decreasing also in the deformation along y -axis of the virtual joints (Figures 119, 120, 121 and 122).

Now, the end effector position presents less oscillations along x, y and z axes (Figures 123, 124 and 125).

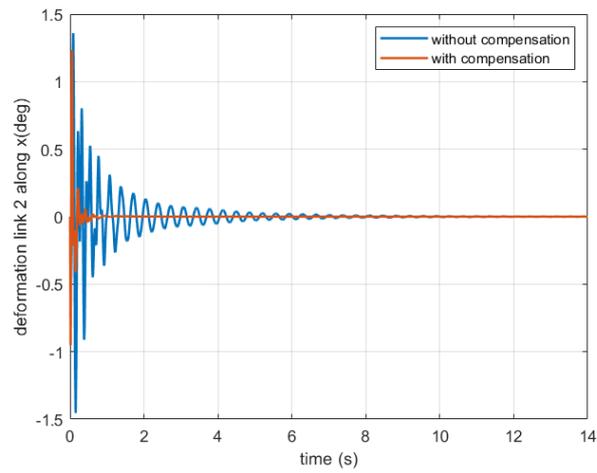


Figure 119 Deformation link 2 along x-axis with and without compensation

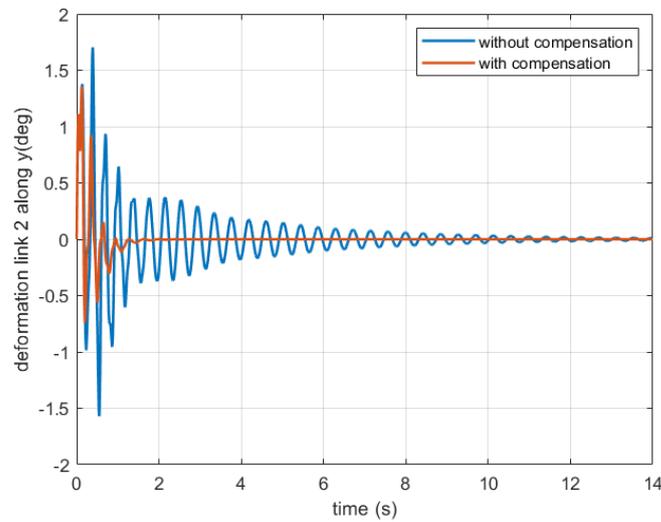


Figure 120 Deformation link 2 along y-axis with and without compensation

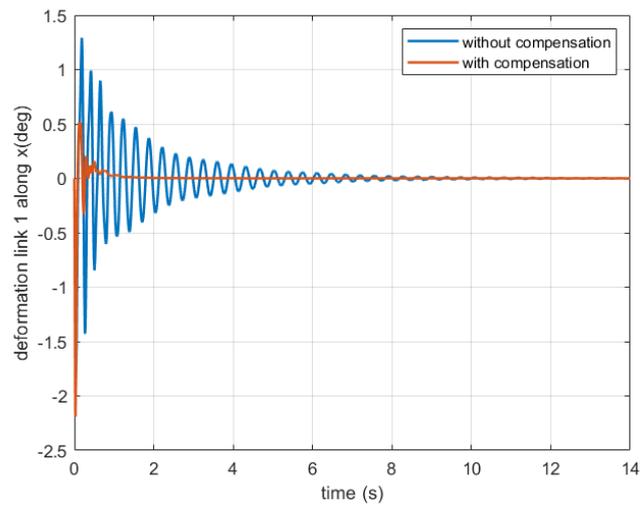


Figure 121 Deformation link 1 along x-axis with and without compensation

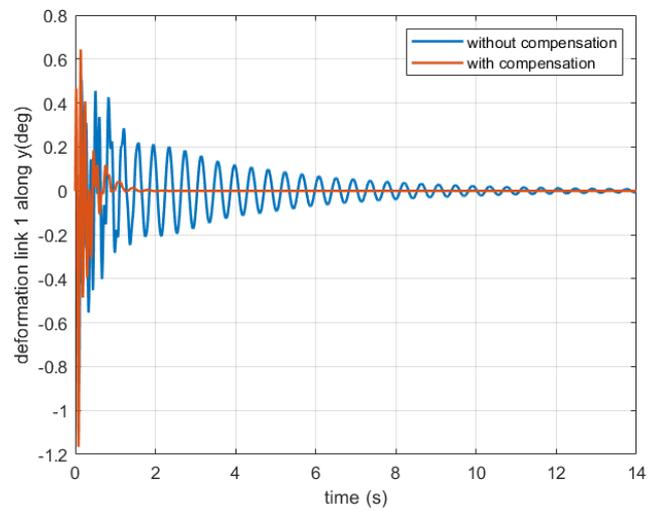


Figure 122 Deformation link 1 along y-axis with and without compensation

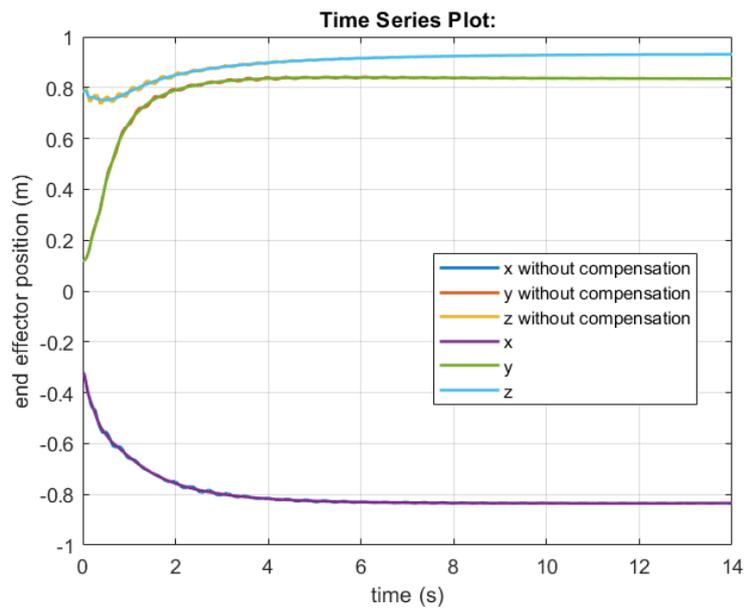


Figure 123 End effector positions with and without compensation

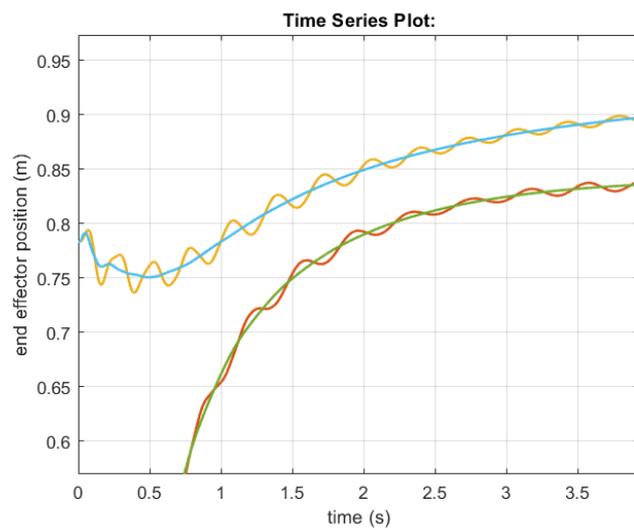


Figure 124 Zoom end effector positions along x and y axes

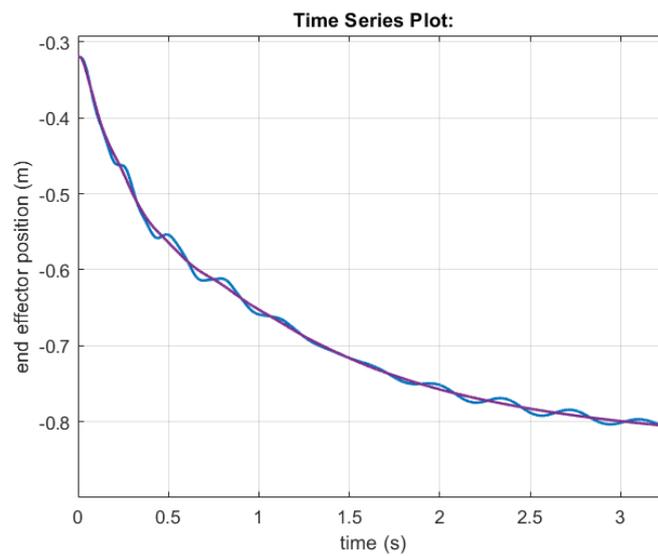


Figure 125 Zoom end effector position along z-axis

The torques applied to the links decreased (Figure 126) as well as the end effector accelerations (Figure 127).

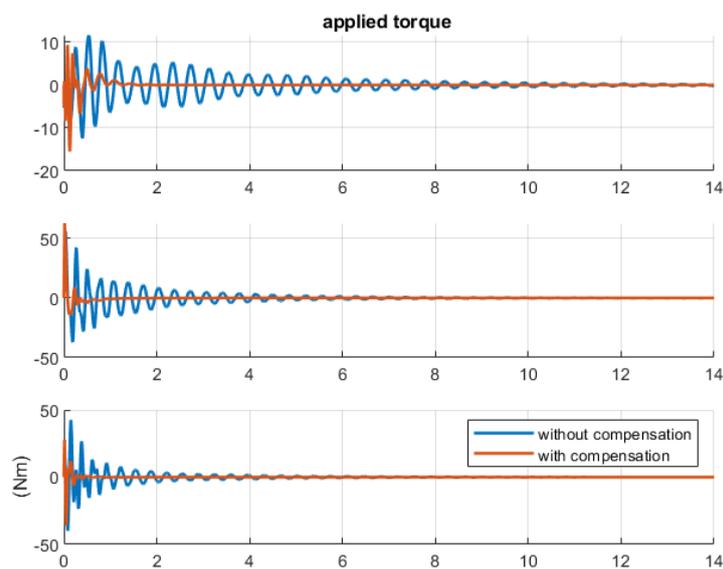


Figure 126 Torques applied to the joints with and without compensation

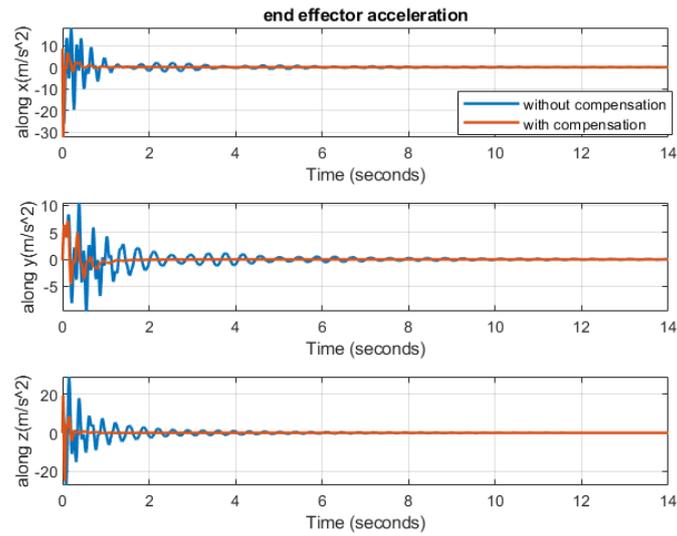


Figure 127 End effector accelerations with and without compensation

Figure 128 shows the trajectory of end point position with and without compensation and the result of additional control input is a more damped trajectory.

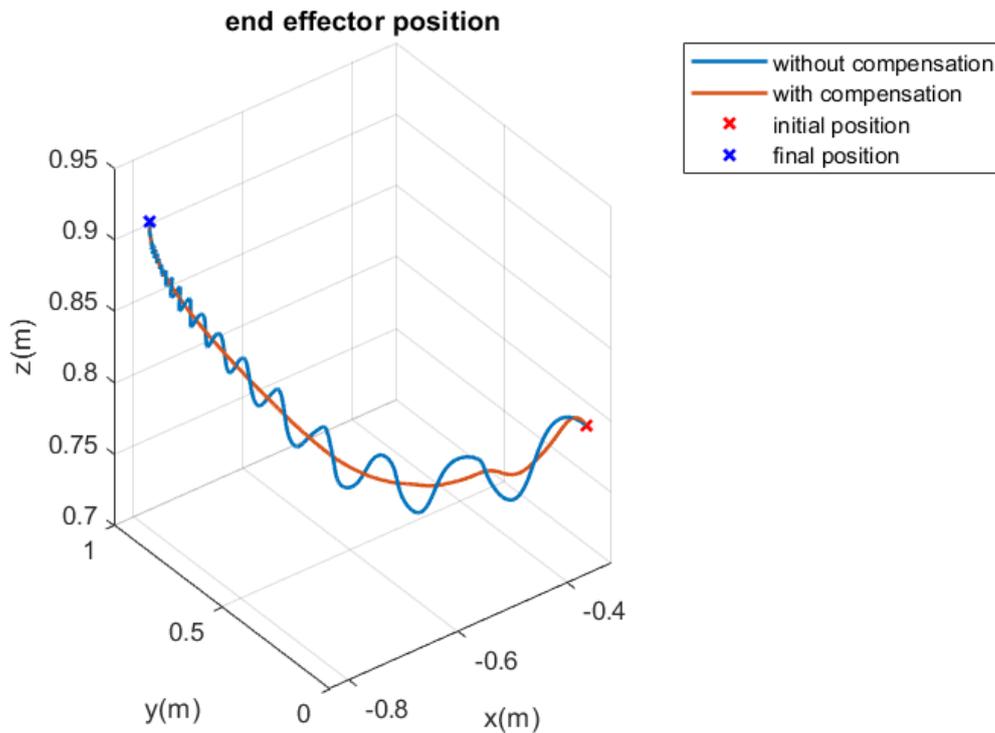


Figure 128 End effector trajectory with and without compensation

In Figures 129 and 130 a comparison between the behaviour of the rigid equivalent robotic arm and the PRBM is shown in terms of end effector trajectory and by adding a compensation expect for a transient the behaviour is similar in the absence of gravity while in the presence of gravity there is a gap. In the absence of gravity (Figure 129) the end point trajectory is almost the same as the one of the rigid system, while in the presence of gravity (Figure 130) there is an offset. This offset is responsible for the deformations that don't converge to zero and this is due to the gravity effect.

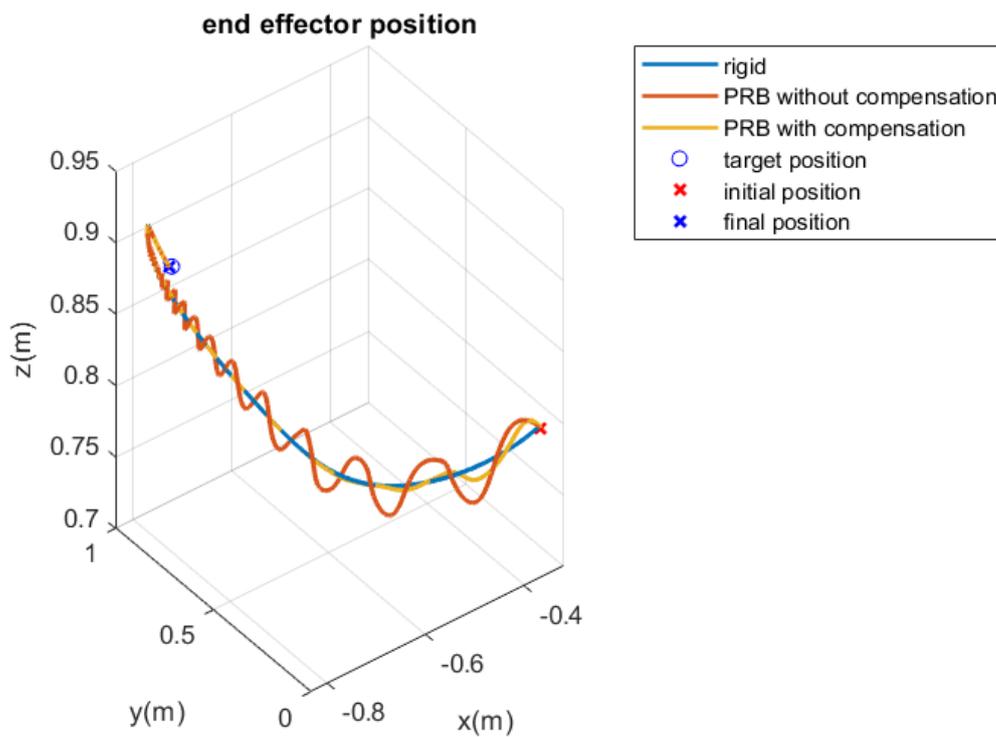


Figure 129 End effector trajectory with and without compensation PRBM and rigid robot without gravity 2 kg of payload

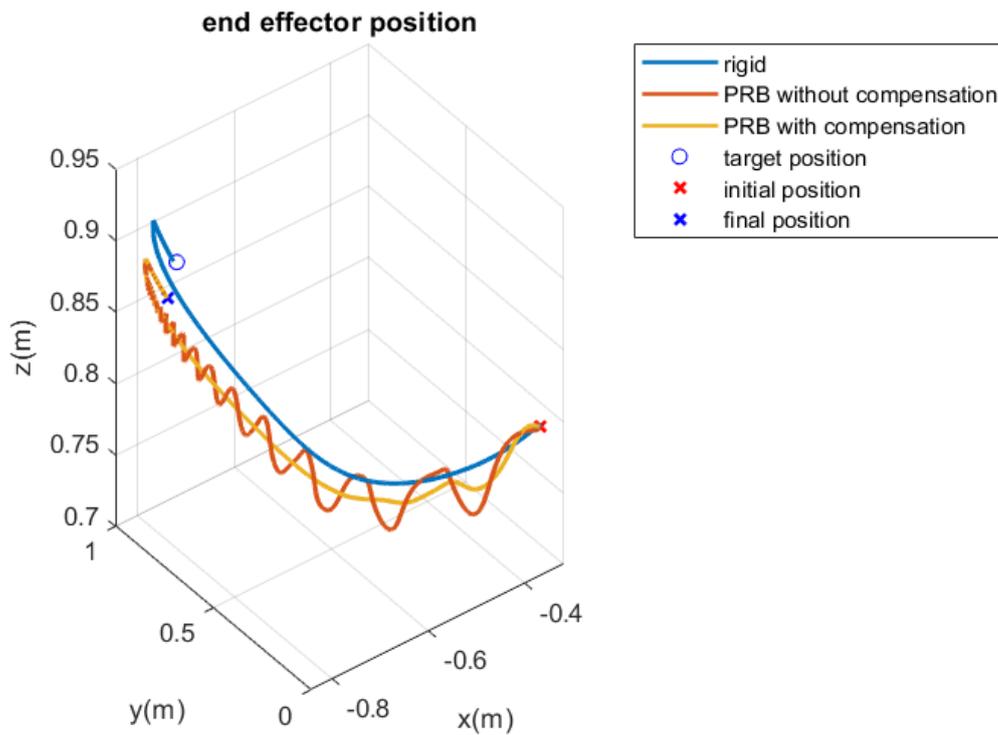


Figure 130 End effector trajectory with and without compensation PRBM and rigid robot with gravity 2 kg of payload

In order to test the robustness of this control algorithm, simulations with the same PID parameters for a payload of 1 kg (Figures 131 and 132) and 0 kg (Figures 133 and 134) is performed with and without the presence of gravity. It's possible to notice that by decreasing the payload, also the gap between the PRBM end point trajectory and the rigid one decreases since the deformations are correlated to the payload. Also in these case the trajectory presents less oscillations.

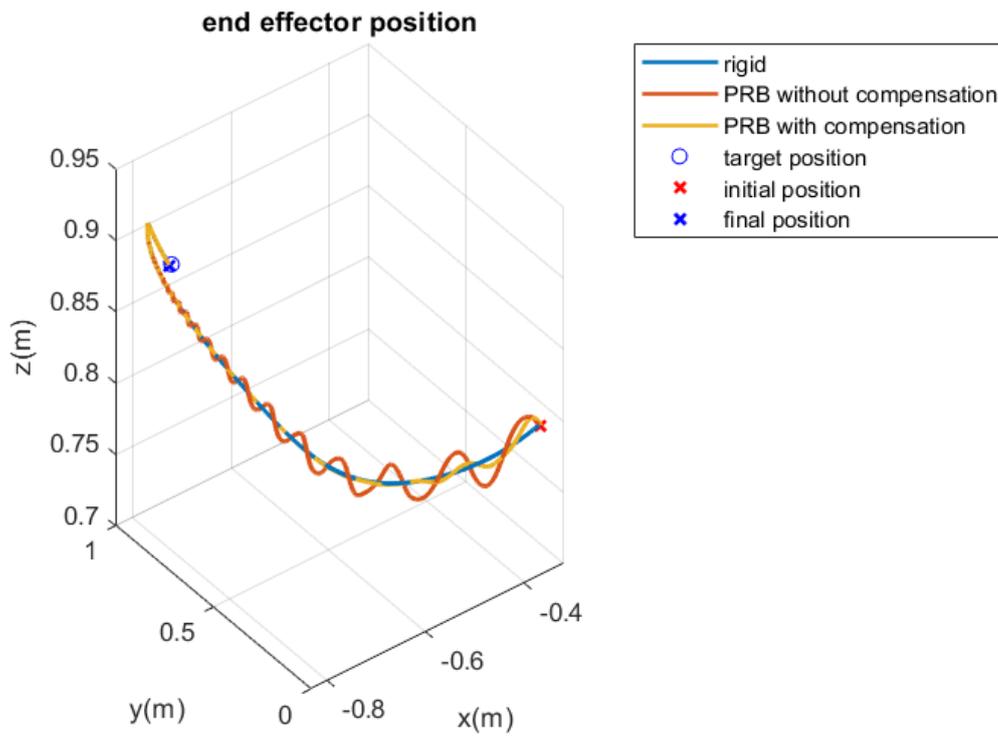


Figure 131 End effector trajectory with and without compensation PRBM and rigid robot without gravity 1 kg of payload

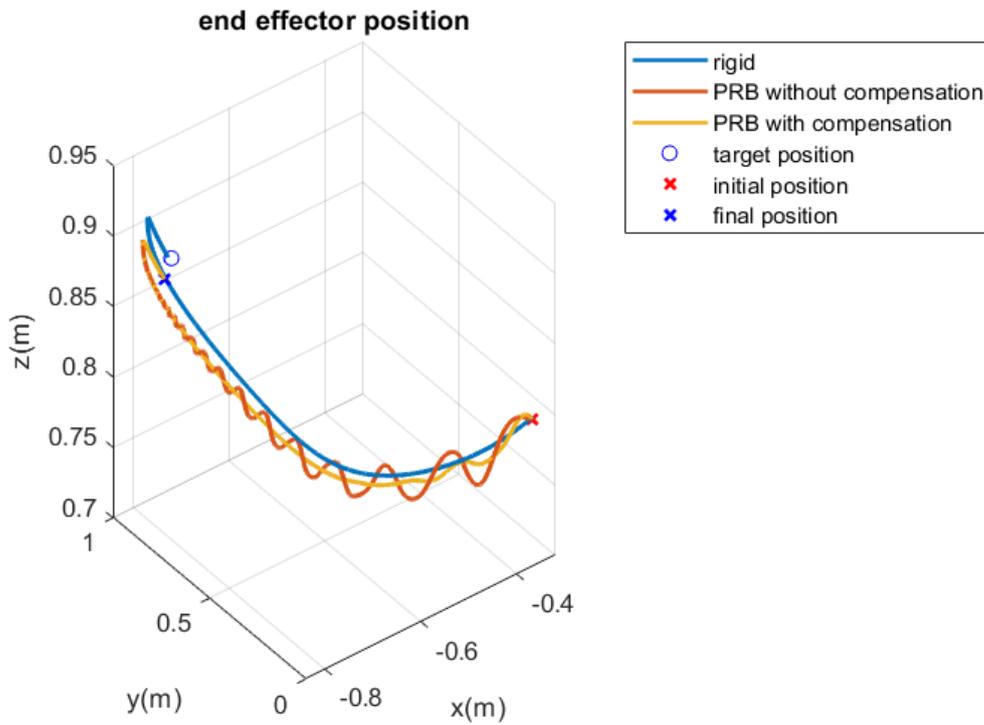


Figure 132 End effector trajectory with and without compensation PRBM and rigid robot with gravity 1 kg of payload

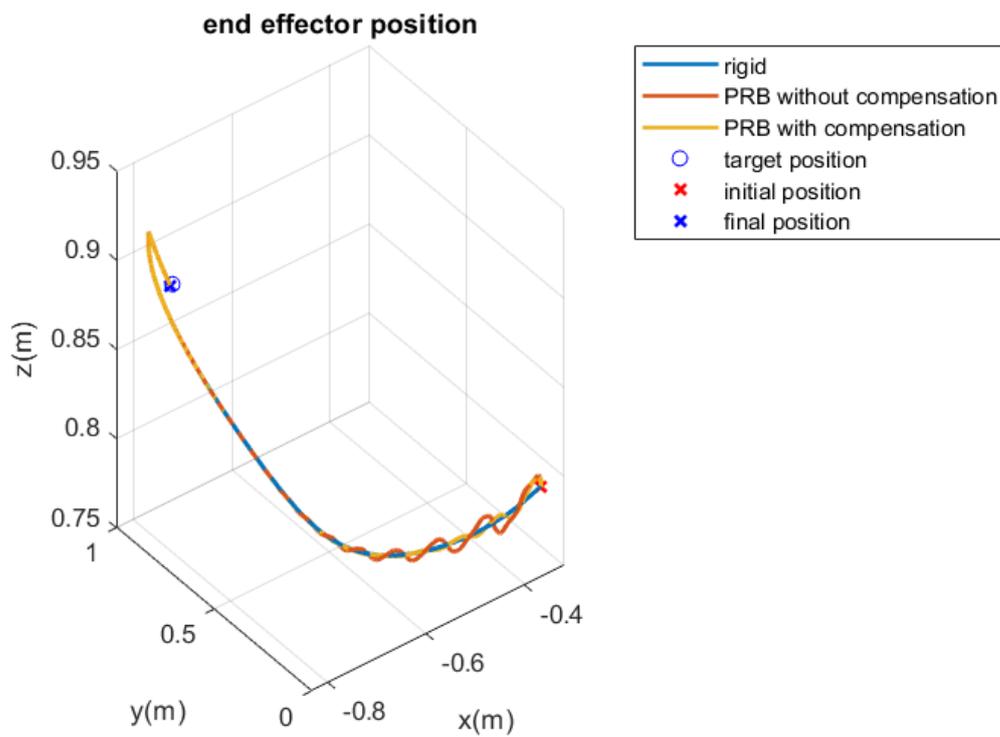


Figure 133 End effector trajectory with and without compensation PRBM and rigid robot without gravity 0 kg of payload

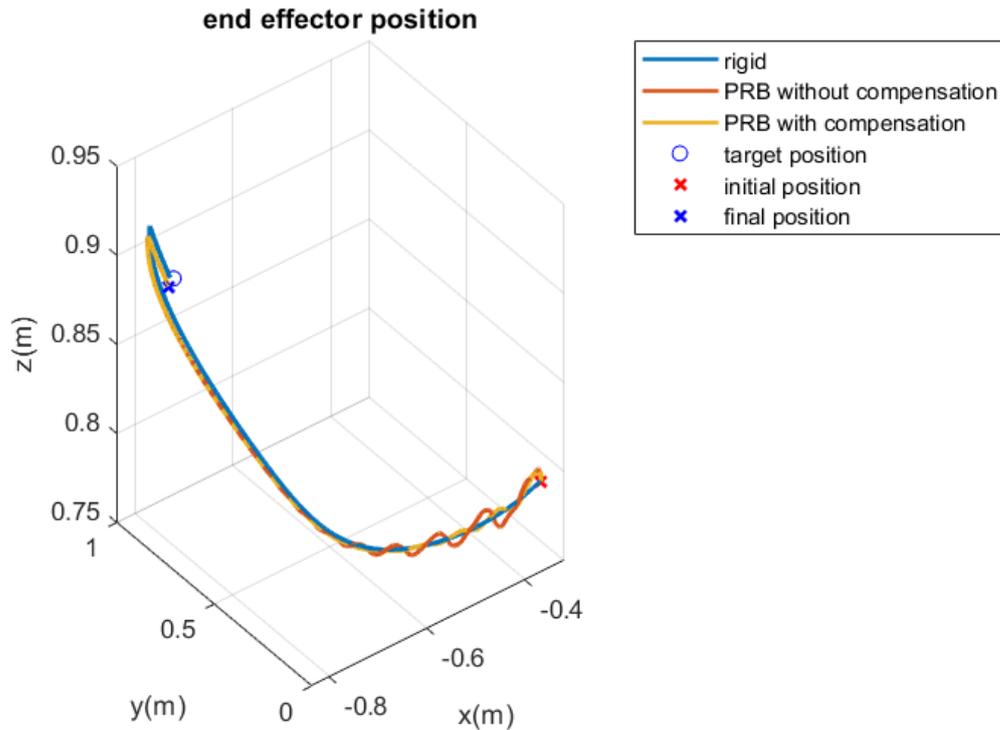


Figure 134 End effector trajectory with and without compensation PRBM and rigid robot without gravity 0 kg of payload

Conclusions

In this thesis work two different ways to model an inflatable robot manipulator are discussed and implemented in Simscape. Then, a frequency analysis is performed in order to tune a vibration damping control parameters. The proposed control aims to use link relative acceleration signals to properly damp undesired vibrations. The applied control scheme is tested in simulations in the Simulink environment both with and without the presence of gravity. Results show the efficacy of the methodology in compensation of end-effector oscillations. Moreover, with the vibration damping control, the deformations occurring along virtual joints (since a pseudo-rigid body is tested in simulations) are more damped and also the torques applied to the motoreducers decrease. This control system could be useful since the inflatable robotic arm, after grasping an object, could be affected by undesired oscillations that should be damped. Firstly, the proposed control scheme is tested by providing a known angular position reference input in the joint space and by exploiting a decentralized control scheme with position and velocity feedback chains. Then, a cartesian position is reached by using the differential kinematics of the 3 degrees of freedom robotic arm and with the additional control input the performances improve in terms of a more damped end effector trajectory and more damped oscillations. Furthermore, payload variation does not change the results and it shows that the control algorithm is insensible to payload variations.

Appendix

Set membership identification MATLAB code

```

clear all, close all, clc

%% load data
load out
x = out.posFB.Data(:,2);
theta2 = out.qFB.Data(:,4);
theta2_dot = out.qFB.Data(:,5);
theta3 = out.qFB.Data(:,7);
theta3_dot = out.qFB.Data(:,8);
N = length(x);
% N = 100;
delta_eta = 0.005;
figure(1)
plot(1:N,out.posFB.Data(1:N,2),'r')
legend('measured x')

figure(2)
plot(1:N,out.qFB.Data(1:N,4),'b')
legend('measured q2')

figure(3)
plot(1:N,out.qFB.Data(1:N,5),'r')
legend('measured q2dot')

figure(4)
plot(1:N,out.qFB.Data(1:N,7),'b')
legend('measured q3')

figure(5)
plot(1:N,out.qFB.Data(1:N,4),'r')
legend('measured q3dot')
x = out.posFB.Data(1:end,1);
y = out.posFB.Data(1:end,2);
z = out.posFB.Data(1:end,3);
posEE = [x y z];
in = [out.qFB.Data(1:N,1) out.qFB.Data(1:N,2) out.qFB.Data(1:N,3)
out.qFB.Data(1:N,4)...
out.qFB.Data(1:N,5) out.qFB.Data(1:N,6) out.qFB.Data(1:N,7)
out.qFB.Data(1:N,8)...
out.qFB.Data(1:N,9)];
%% sparsePOP
for j = 1:2
    if j == 1
        c = 1;
    else
        c = -1;
    end
end

```

Appendix

```
for k = 1:20
    %Objective function
    objPoly.typeCone = 1;
    objPoly.degree = 1;
    objPoly.dimVar = 20+N;
    objPoly.noTerms = 1;
    aux = zeros(1,20+N);
    aux(1,k) = 1;
    objPoly.supports = sparse(aux);
    objPoly.coef = c;

    %Equality constraint
    for t = 9:N
        ineqPolySys{t-8}.typeCone = -1;
        ineqPolySys{t-8}.degree = 5;
        ineqPolySys{t-8}.dimVar = 20+N;
        ineqPolySys{t-8}.noTerms = 486;
        aux1 = zeros(486,20+N);
        coef = -zeros(486,1);
        % x(t)
        %1 a1*x(t-1)
        aux1(1,1) = 1;
        coef(1,1) = x(t-1);
        %2 a3*x(t-1)
        aux1(2,3) = 1;
        coef(2,1) = x(t-1);
        %3 a5*x(t-1)
        aux1(3,5) = 1;
        coef(3,1) = x(t-1);
        %4 a7*x(t-1)
        aux1(4,7) = 1;
        coef(4,1) = x(t-1);
        %5 a2*x(t-2)
        aux1(5,2) = 1;
        coef(5,1) = x(t-2);
        %6 a4*x(t-2)
        aux1(6,4) = 1;
        coef(6,1) = x(t-2);
        %7 a6*x(t-2)
        aux1(7,6) = 1;
        coef(7,1) = x(t-2);
        %8 a8*x(t-2)
        aux1(8,8) = 1;
        coef(8,1) = x(t-2) ;
        %9 a1*a3*x(t-2)
        aux1(9,1) = 1;
        aux1(9,3) = 1;
        coef(9,1) = x(t-2);
        %10 a1*a4*x(t-3)
        aux1(10,1) = 1;
        aux1(10,4) = 1;
        coef(10,1) = x(t-3);
        %11 a1*a5*x(t-2)
        aux1(11,1) = 1;
        aux1(11,5) = 1;
        coef(11,1) = x(t-2);
```

```

%12 a2*a3*x(t-3)
aux1(12,2) = 1;
aux1(12,3) = 1;
coef(12,1) = x(t-3);
%13 a1*a6*x(t-3)
aux1(13,1) = 1;
aux1(13,6) = 1;
coef(13,1) = x(t-3);
%14 a1*a7*x(t-2)
aux1(14,1) = 1;
aux1(14,7) = 1;
coef(14,1) = x(t-2);
%15 a2*a4*x(t-4)
aux1(15,2) = 1;
aux1(15,4) = 1;
coef(15,1) = x(t-4);
%16 a2*a5*x(t-3)
aux1(16,2) = 1;
aux1(16,5) = 1;
coef(16,1) = x(t-3);
%17 a3*a5*x(t-2)
aux1(17,3) = 1;
aux1(17,5) = 1;
coef(17,1) = x(t-2);
%18 a1*a8*x(t-3)
aux1(18,1) = 1;
aux1(18,8) = 1;
coef(18,1) = x(t-3);
%19 a2*a6*x(t-4)
aux1(19,2) = 1;
aux1(19,6) = 1;
coef(19,1) = x(t-4);
%20 a2*a7*x(t-3)
aux1(20,2) = 1;
aux1(20,7) = 1;
coef(20,1) = x(t-3);
%21 a3*a6*x(t-3)
aux1(21,3) = 1;
aux1(21,6) = 1;
coef(21,1) = x(t-3);
%22 a3*a7*x(t-2)
aux1(22,3) = 1;
aux1(22,7) = 1;
coef(22,1) = x(t-2);
%23 a4*a5*x(t-3)
aux1(23,4) = 1;
aux1(23,5) = 1;
coef(23,1) = x(t-3);
%24 a2*a8*x(t-4)
aux1(24,2) = 1;
aux1(24,8) = 1;
coef(24,1) = x(t-4);
%25 a3*a8*x(t-3)
aux1(25,3) = 1;
aux1(25,8) = 1;
coef(25,1) = x(t-3);
%26 a4*a6*x(t-4)

```

Appendix

```
aux1(26,4) = 1;
aux1(26,6) = 1;
coef(26,1) = x(t-4);
%27 a4*a7*x(t-3)
aux1(27,4) = 1;
aux1(27,7) = 1;
coef(27,1) = x(t-3);
%28 a5*a7*x(t-2)
aux1(28,5) = 1;
aux1(28,7) = 1;
coef(28,1) = x(t-2);
%29 a4*a8*x(t-4)
aux1(29,4) = 1;
aux1(29,8) = 1;
coef(29,1) = x(t-4);
%30 a5*a8*x(t-3)
aux1(30,5) = 1;
aux1(30,8) = 1;
coef(30,1) = x(t-3);
%31 a6*a7*x(t-3)
aux1(31,6) = 1;
aux1(31,7) = 1;
coef(31,1) = x(t-3);
%32 a6*a8*x(t-4)
aux1(32,6) = 1;
aux1(32,8) = 1;
coef(32,1) = x(t-4);
%33 a1*a3*a5*x(t-3)
aux1(33,1) = 1;
aux1(33,3) = 1;
aux1(33,5) = 1;
coef(33,1) = x(t-3);
%34 a1*a3*a6*x(t-4)
aux1(34,1) = 1;
aux1(34,3) = 1;
aux1(34,6) = 1;
coef(34,1) = x(t-4);
%35 a1*a3*a7*x(t-3)
aux1(35,1) = 1;
aux1(35,3) = 1;
aux1(35,7) = 1;
coef(35,1) = x(t-3);
%36 a1*a4*a5*x(t-4)
aux1(36,1) = 1;
aux1(36,4) = 1;
aux1(36,5) = 1;
coef(36,1) = x(t-4);
%37 a2*a3*a5*x(t-4)
aux1(37,2) = 1;
aux1(37,3) = 1;
aux1(37,5) = 1;
coef(37,1) = x(t-4);
%38 a1*a3*a8*x(t-4)
aux1(38,1) = 1;
aux1(38,3) = 1;
aux1(38,8) = 1;
coef(38,1) = x(t-4);
```

Appendix

```
%39 a1*a4*a6*x(t-5)
aux1(39,1) = 1;
aux1(39,4) = 1;
aux1(39,6) = 1;
coef(39,1) = x(t-5);
%40 a1*a4*a7*x(t-4)
aux1(40,1) = 1;
aux1(40,4) = 1;
aux1(40,7) = 1;
coef(40,1) = x(t-4);
%41 a1*a5*a7*x(t-3)
aux1(41,1) = 1;
aux1(41,5) = 1;
aux1(41,7) = 1;
coef(41,1) = x(t-3);
%42 a2*a3*a6*x(t-5)
aux1(42,2) = 1;
aux1(42,3) = 1;
aux1(42,6) = 1;
coef(42,1) = x(t-5);
%43 a2*a3*a7*x(t-4)
aux1(43,2) = 1;
aux1(43,3) = 1;
aux1(43,7) = 1;
coef(43,1) = x(t-4);
%44 a2*a4*a5*x(t-5)
aux1(44,2) = 1;
aux1(44,4) = 1;
aux1(44,5) = 1;
coef(44,1) = x(t-5);
%45 a1*a4*a8*x(t-5)
aux1(45,1) = 1;
aux1(45,4) = 1;
aux1(45,8) = 1;
coef(45,1) = x(t-5);
%46 a1*a5*a8*x(t-4)
aux1(46,1) = 1;
aux1(46,5) = 1;
aux1(46,8) = 1;
coef(46,1) = x(t-4);
%47 a1*a6*a7*x(t-4)
aux1(47,1) = 1;
aux1(47,6) = 1;
aux1(47,7) = 1;
coef(47,1) = x(t-4);
%48 a2*a3*a8*x(t-5)
aux1(48,2) = 1;
aux1(48,3) = 1;
aux1(48,8) = 1;
coef(48,1) = x(t-5);
%49 a2*a4*a6*x(t-6)
aux1(49,2) = 1;
aux1(49,4) = 1;
aux1(49,6) = 1;
coef(49,1) = x(t-6);
%50 a2*a4*a7*x(t-5)
aux1(50,2) = 1;
```

Appendix

```
aux1(50,4) = 1;
aux1(50,7) = 1;
coef(50,1) = x(t-5);
%51 a2*a5*a7*x(t-4)
aux1(51,2) = 1;
aux1(51,5) = 1;
aux1(51,7) = 1;
coef(51,1) = x(t-4);
%52 a3*a5*a7*x(t-3)
aux1(52,3) = 1;
aux1(52,5) = 1;
aux1(52,7) = 1;
coef(52,1) = x(t-3);
%53 a1*a6*a8*x(t-5)
aux1(53,1) = 1;
aux1(53,6) = 1;
aux1(53,8) = 1;
coef(53,1) = x(t-5);
%54 a2*a4*a8*x(t-6)
aux1(54,2) = 1;
aux1(54,4) = 1;
aux1(54,8) = 1;
coef(54,1) = x(t-6);
%55 a2*a5*a8*x(t-5)
aux1(55,2) = 1;
aux1(55,5) = 1;
aux1(55,8) = 1;
coef(55,1) = x(t-5);
%56 a2*a6*a7*x(t-5)
aux1(56,2) = 1;
aux1(56,6) = 1;
aux1(56,7) = 1;
coef(56,1) = x(t-5);
%57 a3*a5*a8*x(t-4)
aux1(57,3) = 1;
aux1(57,5) = 1;
aux1(57,8) = 1;
coef(57,1) = x(t-4);
%58 a3*a6*a7*x(t-4)
aux1(58,3) = 1;
aux1(58,6) = 1;
aux1(58,7) = 1;
coef(58,1) = x(t-4);
%59 a4*a5*a7*x(t-4)
aux1(59,4) = 1;
aux1(59,5) = 1;
aux1(59,7) = 1;
coef(59,1) = x(t-4);
%60 a2*a6*a8*x(t-6)
aux1(60,2) = 1;
aux1(60,6) = 1;
aux1(60,8) = 1;
coef(60,1) = x(t-6);
%61 a3*a6*a8*x(t-5)
aux1(61,3) = 1;
aux1(61,6) = 1;
aux1(61,8) = 1;
```

Appendix

```
coef(61,1) = x(t-5);
%62 a4*a5*a8*x(t-5)
aux1(62,4) = 1;
aux1(62,5) = 1;
aux1(62,8) = 1;
coef(62,1) = x(t-5);
%63 a4*a6*a7*x(t-5)
aux1(63,4) = 1;
aux1(63,6) = 1;
aux1(63,7) = 1;
coef(63,1) = x(t-5);
%64 a4*a6*a8*x(t-6)
aux1(64,4) = 1;
aux1(64,6) = 1;
aux1(64,8) = 1;
coef(64,1) = x(t-6);
%65 a1*a3*a5*a7*x(t-4)
aux1(65,1) = 1;
aux1(65,3) = 1;
aux1(65,5) = 1;
aux1(65,7) = 1;
coef(65,1) = x(t-4);
% 66 a1*a3*a5*a8*x(t-5)
aux1(66,1) = 1;
aux1(66,3) = 1;
aux1(66,5) = 1;
aux1(66,8) = 1;
coef(66,1) = x(t-5);
%67 a1*a3*a6*a7*x(t-5)
aux1(67,1) = 1;
aux1(67,3) = 1;
aux1(67,6) = 1;
aux1(67,7) = 1;
coef(67,1) = x(t-5);
%68 a1*a4*a5*a7*x(t-5)
aux1(68,1) = 1;
aux1(68,4) = 1;
aux1(68,5) = 1;
aux1(68,7) = 1;
coef(68,1) = x(t-5);
%69 a2*a3*a5*a7*x(t-5)
aux1(69,2) = 1;
aux1(69,3) = 1;
aux1(69,5) = 1;
aux1(69,7) = 1;
coef(69,1) = x(t-5);
%70 a1*a3*a6*a8*x(t-6)
aux1(70,1) = 1;
aux1(70,3) = 1;
aux1(70,6) = 1;
aux1(70,8) = 1;
coef(70,1) = x(t-6);
%71 a1*a4*a5*a8*x(t-6)
aux1(71,1) = 1;
aux1(71,4) = 1;
aux1(71,5) = 1;
aux1(71,8) = 1;
```

Appendix

```
coef(71,1) = x(t-6);
%72 a1*a4*a6*a7*x(t-6)
aux1(72,1) = 1;
aux1(72,4) = 1;
aux1(72,6) = 1;
aux1(72,7) = 1;
coef(72,1) = x(t-6);
%73 a2*a3*a5*a8*x(t-6)
aux1(73,2) = 1;
aux1(73,3) = 1;
aux1(73,5) = 1;
aux1(73,8) = 1;
coef(73,1) = x(t-6);
%74 a2*a3*a6*a7*x(t-6)
aux1(74,2) = 1;
aux1(74,3) = 1;
aux1(74,6) = 1;
aux1(74,7) = 1;
coef(74,1) = x(t-6);
%75 a2*a4*a5*a7*x(t-6)
aux1(75,2) = 1;
aux1(75,4) = 1;
aux1(75,5) = 1;
aux1(75,7) = 1;
coef(75,1) = x(t-6);
%76 a1*a4*a6*a8*x(t-7)
aux1(76,1) = 1;
aux1(76,4) = 1;
aux1(76,6) = 1;
aux1(76,8) = 1;
coef(76,1) = x(t-7);
%77 a2*a3*a6*a8*x(t-7)
aux1(77,2) = 1;
aux1(77,3) = 1;
aux1(77,6) = 1;
aux1(77,8) = 1;
coef(77,1) = x(t-7);
%78 a2*a4*a5*a8*x(t-7)
aux1(78,2) = 1;
aux1(78,4) = 1;
aux1(78,5) = 1;
aux1(78,8) = 1;
coef(78,1) = x(t-7);
%79 a2*a4*a6*a7*x(t-7)
aux1(79,2) = 1;
aux1(79,4) = 1;
aux1(79,6) = 1;
aux1(79,7) = 1;
coef(79,1) = x(t-7);
%80 a2*a4*a6*a8*x(t-8)
aux1(80,2) = 1;
aux1(80,4) = 1;
aux1(80,6) = 1;
aux1(80,8) = 1;
coef(80,1) = x(t-8);
%81 x(t)
coef(81,1) = x(t);
```

```

% theta2
%82 -b1*theta2(t)
aux1(82,9) = 1;
coef(82,1) = -theta2(t);
%83 -b2*theta2(t-1)
aux1(83,10) = 1;
coef(83,1) = -theta2(t-1);
%84 -b3*theta2(t-2)
aux1(84,11) = 1;
coef(84,1) = -theta2(t-2);
%85 -a3*b1*theta2(t-1)
aux1(85,3) = 1;
aux1(85,9) = 1;
coef(85,1) = -theta2(t-1);
%86 -a5*b1*theta2(t-1)
aux1(86,5) = 1;
aux1(86,9) = 1;
coef(86,1) = -theta2(t-1);
%87 -a7*b1*theta2(t-1)
aux1(87,7) = 1;
aux1(87,9) = 1;
coef(87,1) = -theta2(t-1);
%88 -a3*b2*theta2(t-2)
aux1(88,3) = 1;
aux1(88,10) = 1;
coef(88,1) = -theta2(t-2);
%89 -a4*b1*theta2(t-2)
aux1(89,4) = 1;
aux1(89,9) = 1;
coef(89,1) = -theta2(t-2);
%90 -a3*b3*theta2(t-3)
aux1(90,3) = 1;
aux1(90,11) = 1;
coef(90,1) = -theta2(t-3);
%91 -a4*b2*theta2(t-3)
aux1(91,4) = 1;
aux1(91,10) = 1;
coef(91,1) = -theta2(t-3);
%92 -a5*b2*theta2(t-2)
aux1(92,5) = 1;
aux1(92,10) = 1;
coef(92,1) = -theta2(t-2);
%93 -a6*b1*theta2(t-2)
aux1(93,6) = 1;
aux1(93,9) = 1;
coef(93,1) = -theta2(t-2);
%94 -a4*b3*theta2(t-4)
aux1(94,4) = 1;
aux1(94,11) = 1;
coef(94,1) = -theta2(t-4);
%95 -a5*b3*theta2(t-3)
aux1(95,5) = 1;
aux1(95,11) = 1;
coef(95,1) = -theta2(t-3);
%96 -a6*b2*theta2(t-3)
aux1(96,6) = 1;
aux1(96,10) = 1;

```

Appendix

```
coef(96,1) = -theta2(t-3);
%97 -a7*b2*theta2(t-2)
aux1(97,7) = 1;
aux1(97,10) = 1;
coef(97,1) = -theta2(t-2);
%98 -a8*b1*theta2(t-2)
aux1(98,8) = 1;
aux1(98,9) = 1;
coef(98,1) = -theta2(t-2);
%99 -a6*b3*theta2(t-4)
aux1(99,6) = 1;
aux1(99,11) = 1;
coef(99,1) = -theta2(t-4);
%100 -a7*b3*theta2(t-3)
aux1(100,7) = 1;
aux1(100,11) = 1;
coef(100,1) = -theta2(t-3);
%101 -a8*b2*theta2(t-3)
aux1(101,8) = 1;
aux1(101,10) = 1;
coef(101,1) = -theta2(t-3);
%102 -a8*b3*theta2(t-4)
aux1(102,8) = 1;
aux1(102,11) = 1;
coef(102,1) = -theta2(t-4);
%103 -a3*a5*b1*theta2(t-2)
aux1(103,3) = 1;
aux1(103,5) = 1;
aux1(103,9) = 1;
coef(103,1) = -theta2(t-2);
%104 -a3*a5*b2*theta2(t-3)
aux1(104,3) = 1;
aux1(104,5) = 1;
aux1(104,10) = 1;
coef(104,1) = -theta2(t-3);
%105 -a3*a6*b1*theta2(t-3)
aux1(105,3) = 1;
aux1(105,6) = 1;
aux1(105,9) = 1;
coef(105,1) = -theta2(t-3);
%106 -a3*a7*b1*theta2(t-2)
aux1(106,3) = 1;
aux1(106,7) = 1;
aux1(106,9) = 1;
coef(106,1) = -theta2(t-2);
%107 -a4*a5*b1*theta2(t-3)
aux1(107,4) = 1;
aux1(107,5) = 1;
aux1(107,9) = 1;
coef(107,1) = -theta2(t-3);
%108 -a3*a5*b3*theta2(t-4)
aux1(108,3) = 1;
aux1(108,5) = 1;
aux1(108,11) = 1;
coef(108,1) = -theta2(t-4);
%109 -a3*a6*b2*theta2(t-4)
aux1(109,3) = 1;
```

Appendix

```
aux1(109,6) = 1;
aux1(109,10) = 1;
coef(109,1) = -theta2(t-4);
%110 -a3*a7*b2*theta2(t-3)
aux1(110,3) = 1;
aux1(110,7) = 1;
aux1(110,10) = 1;
coef(110,1) = -theta2(t-3);
%111 -a3*a8*b1*theta2(t-3)
aux1(111,3) = 1;
aux1(111,8) = 1;
aux1(111,9) = 1;
coef(111,1) = -theta2(t-3);
%112 -a4*a5*b2*theta2(t-4)
aux1(112,4) = 1;
aux1(112,5) = 1;
aux1(112,10) = 1;
coef(112,1) = -theta2(t-4);
%113 -a4*a6*b1*theta2(t-4)
aux1(113,4) = 1;
aux1(113,6) = 1;
aux1(113,9) = 1;
coef(113,1) = -theta2(t-4);
%114 -a4*a7*b1*theta2(t-3)
aux1(114,4) = 1;
aux1(114,7) = 1;
aux1(114,9) = 1;
coef(114,1) = -theta2(t-3);
%115 -a5*a7*b1*theta2(t-2)
aux1(115,5) = 1;
aux1(115,7) = 1;
aux1(115,9) = 1;
coef(115,1) = -theta2(t-2);
%116 -a3*a6*b3*theta2(t-5)
aux1(116,3) = 1;
aux1(116,6) = 1;
aux1(116,11) = 1;
coef(116,1) = -theta2(t-5);
%117 -a3*a7*b3*theta2(t-4)
aux1(117,3) = 1;
aux1(117,7) = 1;
aux1(117,11) = 1;
coef(117,1) = -theta2(t-4);
%118 -a3*a8*b2*theta2(t-4)
aux1(118,3) = 1;
aux1(118,8) = 1;
aux1(118,10) = 1;
coef(118,1) = -theta2(t-4);
%119 -a4*a5*b3*theta2(t-5)
aux1(119,4) = 1;
aux1(119,5) = 1;
aux1(119,11) = 1;
coef(119,1) = -theta2(t-5);
%120 -a4*a6*b2*theta2(t-5)
aux1(120,4) = 1;
aux1(120,6) = 1;
aux1(120,10) = 1;
```

Appendix

```
coef(120,1) = -theta2(t-5);
%121 -a4*a7*b2*theta2(t-4)
aux1(121,4) = 1;
aux1(121,7) = 1;
aux1(121,10) = 1;
coef(121,1) = -theta2(t-4);
%122 -a4*a8*b1*theta2(t-4)
aux1(122,4) = 1;
aux1(122,8) = 1;
aux1(122,9) = 1;
coef(122,1) = -theta2(t-4);
%123 -a5*a7*b2*theta2(t-3)
aux1(123,5) = 1;
aux1(123,7) = 1;
aux1(123,10) = 1;
coef(123,1) = -theta2(t-3);
%124 -a5*a8*b1*theta2(t-3)
aux1(124,5) = 1;
aux1(124,8) = 1;
aux1(124,9) = 1;
coef(124,1) = -theta2(t-3);
%125 -a6*a7*b1*theta2(t-3)
aux1(125,6) = 1;
aux1(125,7) = 1;
aux1(125,9) = 1;
coef(125,1) = -theta2(t-3);
%126 -a3*a8*b3*theta2(t-5)
aux1(126,3) = 1;
aux1(126,8) = 1;
aux1(126,11) = 1;
coef(126,1) = -theta2(t-5);
%127 -a4*a6*b3*theta2(t-6)
aux1(127,4) = 1;
aux1(127,6) = 1;
aux1(127,11) = 1;
coef(127,1) = -theta2(t-6);
%128 -a4*a7*b3*theta2(t-5)
aux1(128,4) = 1;
aux1(128,7) = 1;
aux1(128,11) = 1;
coef(128,1) = -theta2(t-5);
%129 -a4*a8*b2*theta2(t-5)
aux1(129,4) = 1;
aux1(129,8) = 1;
aux1(129,10) = 1;
coef(129,1) = -theta2(t-5);
%130 -a5*a7*b3*theta2(t-4)
aux1(130,5) = 1;
aux1(130,7) = 1;
aux1(130,11) = 1;
coef(130,1) = -theta2(t-4);
%131 -a5*a8*b2*theta2(t-4)
aux1(131,5) = 1;
aux1(131,8) = 1;
aux1(131,10) = 1;
coef(131,1) = -theta2(t-4);
%132 -a6*a7*b2*theta2(t-4)
```

Appendix

```
aux1(132,6) = 1;
aux1(132,7) = 1;
aux1(132,10) = 1;
coef(132,1) = -theta2(t-4);
%133 -a6*a8*b1*theta2(t-4)
aux1(133,6) = 1;
aux1(133,8) = 1;
aux1(133,9) = 1;
coef(133,1) = -theta2(t-4);
%134 -a4*a8*b3*theta2(t-6)
aux1(134,4) = 1;
aux1(134,8) = 1;
aux1(134,11) = 1;
coef(134,1) = -theta2(t-6);
%135 -a5*a8*b3*theta2(t-5)
aux1(135,5) = 1;
aux1(135,8) = 1;
aux1(135,11) = 1;
coef(135,1) = -theta2(t-5);
%136 -a6*a7*b3*theta2(t-5)
aux1(136,6) = 1;
aux1(136,7) = 1;
aux1(136,11) = 1;
coef(136,1) = -theta2(t-5);
%137 -a6*a8*b2*theta2(t-5)
aux1(137,6) = 1;
aux1(137,8) = 1;
aux1(137,10) = 1;
coef(137,1) = -theta2(t-5);
%138 -a6*a8*b3*theta2(t-6)
aux1(138,6) = 1;
aux1(138,8) = 1;
aux1(138,11) = 1;
coef(138,1) = -theta2(t-6);
%139 -a3*a5*a7*b1*theta2(t-3)
aux1(139,3) = 1;
aux1(139,5) = 1;
aux1(139,7) = 1;
aux1(139,9) = 1;
coef(139,1) = -theta2(t-3);
%140 -a3*a5*a7*b2*theta2(t-4)
aux1(140,3) = 1;
aux1(140,5) = 1;
aux1(140,7) = 1;
aux1(140,10) = 1;
coef(140,1) = -theta2(t-4);
%141 -a3*a5*a8*b1*theta2(t-4)
aux1(141,3) = 1;
aux1(141,5) = 1;
aux1(141,8) = 1;
aux1(141,9) = 1;
coef(141,1) = -theta2(t-4);
%142 -a3*a6*a7*b1*theta2(t-4)
aux1(142,3) = 1;
aux1(142,6) = 1;
aux1(142,7) = 1;
aux1(142,9) = 1;
```

Appendix

```
coef(142,1) = -theta2(t-4);
%143 -a4*a5*a7*b1*theta2(t-4)
aux1(143,4) = 1;
aux1(143,5) = 1;
aux1(143,7) = 1;
aux1(143,9) = 1;
coef(143,1) = -theta2(t-4);
%144 -a3*a5*a7*b3*theta2(t-5)
aux1(144,3) = 1;
aux1(144,5) = 1;
aux1(144,7) = 1;
aux1(144,11) = 1;
coef(144,1) = -theta2(t-5);
%145 -a3*a5*a8*b2*theta2(t-5)
aux1(145,3) = 1;
aux1(145,5) = 1;
aux1(145,8) = 1;
aux1(145,10) = 1;
coef(145,1) = -theta2(t-5);
%146 -a3*a6*a7*b2*theta2(t-5)
aux1(146,3) = 1;
aux1(146,6) = 1;
aux1(146,7) = 1;
aux1(146,10) = 1;
coef(146,1) = -theta2(t-5);
%147 -a3*a6*a8*b1*theta2(t-5)
aux1(147,3) = 1;
aux1(147,6) = 1;
aux1(147,8) = 1;
aux1(147,9) = 1;
coef(147,1) = -theta2(t-5);
%148 -a4*a5*a7*b2*theta2(t-5)
aux1(148,4) = 1;
aux1(148,5) = 1;
aux1(148,7) = 1;
aux1(148,10) = 1;
coef(148,1) = -theta2(t-5);
%149 -a4*a5*a8*b1*theta2(t-5)
aux1(149,4) = 1;
aux1(149,5) = 1;
aux1(149,8) = 1;
aux1(149,9) = 1;
coef(149,1) = -theta2(t-5);
%150 -a4*a6*a7*b1*theta2(t-5)
aux1(150,4) = 1;
aux1(150,6) = 1;
aux1(150,7) = 1;
aux1(150,9) = 1;
coef(150,1) = -theta2(t-5);
%151 -a3*a5*a8*b3*theta2(t-6)
aux1(151,3) = 1;
aux1(151,5) = 1;
aux1(151,8) = 1;
aux1(151,11) = 1;
coef(151,1) = -theta2(t-6);
%152 -a3*a6*a7*b3*theta2(t-6)
aux1(152,3) = 1;
```

Appendix

```
aux1(152,6) = 1;
aux1(152,7) = 1;
aux1(152,11) = 1;
coef(152,1) = -theta2(t-6);
%153 -a3*a6*a8*b2*theta2(t-6)
aux1(153,3) = 1;
aux1(153,6) = 1;
aux1(153,8) = 1;
aux1(153,10) = 1;
coef(153,1) = -theta2(t-6);
%154 -a4*a5*a7*b3*theta2(t-6)
aux1(154,4) = 1;
aux1(154,5) = 1;
aux1(154,7) = 1;
aux1(154,11) = 1;
coef(154,1) = -theta2(t-6);
%155 -a4*a5*a8*b2*theta2(t-6)
aux1(155,4) = 1;
aux1(155,5) = 1;
aux1(155,8) = 1;
aux1(155,10) = 1;
coef(155,1) = -theta2(t-6);
%156 -a4*a6*a7*b2*theta2(t-6)
aux1(156,4) = 1;
aux1(156,6) = 1;
aux1(156,7) = 1;
aux1(156,10) = 1;
coef(156,1) = -theta2(t-6);
%157 -a4*a6*a8*b1*theta2(t-6)
aux1(157,4) = 1;
aux1(157,6) = 1;
aux1(157,8) = 1;
aux1(157,9) = 1;
coef(157,1) = -theta2(t-6);
%158 -a3*a6*a8*b3*theta2(t-7)
aux1(158,3) = 1;
aux1(158,6) = 1;
aux1(158,8) = 1;
aux1(158,11) = 1;
coef(158,1) = -theta2(t-7);
%159 -a4*a5*a8*b3*theta2(t-7)
aux1(159,4) = 1;
aux1(159,5) = 1;
aux1(159,8) = 1;
aux1(159,11) = 1;
coef(159,1) = -theta2(t-7);
%160 -a4*a6*a7*b3*theta2(t-7)
aux1(160,4) = 1;
aux1(160,6) = 1;
aux1(160,7) = 1;
aux1(160,11) = 1;
coef(160,1) = -theta2(t-7);
%161 -a4*a6*a8*b2*theta2(t-7)
aux1(161,4) = 1;
aux1(161,6) = 1;
aux1(161,8) = 1;
aux1(161,10) = 1;
```

Appendix

```
coef(161,1) = -theta2(t-7);
%162 -a4*a6*a8*b3*theta2(t-8)
aux1(162,4) = 1;
aux1(162,6) = 1;
aux1(162,8) = 1;
aux1(162,11) = 1;
coef(162,1) = -theta2(t-8);
% theta2_dot
%163 -b4*theta2_dot(t)
aux1(163,12) = 1;
coef(163,1) = -theta2_dot(t);
%164 -b5*theta2_dot(t-1)
aux1(164,13) = 1;
coef(164,1) = -theta2_dot(t-1);
%165 -b6*theta2_dot(t-2)
aux1(165,14) = 1;
coef(165,1) = -theta2_dot(t-2);
%166 -a1*b4*theta2_dot(t-1)
aux1(166,1) = 1;
aux1(166,12) = 1;
coef(166,1) = -theta2_dot(t-1);
%167 -a5*b4*theta2_dot(t-1)
aux1(167,5) = 1;
aux1(167,12) = 1;
coef(167,1) = -theta2_dot(t-1);
%168 -a7*b4*theta2_dot(t-1)
aux1(168,7) = 1;
aux1(168,12) = 1;
coef(168,1) = -theta2_dot(t-1);
%169 -a1*b5*theta2_dot(t-2)
aux1(169,1) = 1;
aux1(169,13) = 1;
coef(169,1) = -theta2_dot(t-2);
%170 -a2*b4*theta2_dot(t-2)
aux1(170,2) = 1;
aux1(170,12) = 1;
coef(170,1) = -theta2_dot(t-2);
%171 -a1*b6*theta2_dot(t-3)
aux1(171,1) = 1;
aux1(171,14) = 1;
coef(171,1) = -theta2_dot(t-3);
%172 -a2*b5*theta2_dot(t-3)
aux1(172,2) = 1;
aux1(172,13) = 1;
coef(172,1) = -theta2_dot(t-3);
%173 -a2*b6*theta2_dot(t-4)
aux1(173,2) = 1;
aux1(173,14) = 1;
coef(173,1) = -theta2_dot(t-4);
%174 -a5*b5*theta2_dot(t-2)
aux1(174,5) = 1;
aux1(174,13) = 1;
coef(174,1) = -theta2_dot(t-2);
%175 -a6*b4*theta2_dot(t-2)
aux1(175,6) = 1;
aux1(175,12) = 1;
coef(175,1) = -theta2_dot(t-2);
```

```

%176 -a5*b6*theta2_dot(t-3)
aux1(176,5) = 1;
aux1(176,14) = 1;
coef(176,1) = -theta2_dot(t-3);
%177 -a6*b5*theta2_dot(t-3)
aux1(177,6) = 1;
aux1(177,13) = 1;
coef(177,1) = -theta2_dot(t-3);
%178 -a7*b5*theta2_dot(t-2)
aux1(178,7) = 1;
aux1(178,13) = 1;
coef(178,1) = -theta2_dot(t-2);
%179 -a8*b4*theta2_dot(t-2)
aux1(179,8) = 1;
aux1(179,12) = 1;
coef(179,1) = -theta2_dot(t-2);
%180 -a6*b6*theta2_dot(t-4)
aux1(180,6) = 1;
aux1(180,14) = 1;
coef(180,1) = -theta2_dot(t-4);
%181 -a7*b6*theta2_dot(t-3)
aux1(181,7) = 1;
aux1(181,14) = 1;
coef(181,1) = -theta2_dot(t-3);
%182 -a8*b5*theta2_dot(t-3)
aux1(182,8) = 1;
aux1(182,13) = 1;
coef(182,1) = -theta2_dot(t-3);
%183 -a8*b6*theta2_dot(t-4)
aux1(183,8) = 1;
aux1(183,14) = 1;
coef(183,1) = -theta2_dot(t-4);
%184 -a1*a5*b4*theta2_dot(t-2)
aux1(184,1) = 1;
aux1(184,5) = 1;
aux1(184,12) = 1;
coef(184,1) = -theta2_dot(t-2);
%185 -a1*a5*b5*theta2_dot(t-3)
aux1(185,1) = 1;
aux1(185,5) = 1;
aux1(185,13) = 1;
coef(185,1) = -theta2_dot(t-3);
%186 -a1*a6*b4*theta2_dot(t-3)
aux1(186,1) = 1;
aux1(186,6) = 1;
aux1(186,12) = 1;
coef(186,1) = -theta2_dot(t-3);
%187 a1*a7*b4*theta2_dot(t-2)
aux1(187,1) = 1;
aux1(187,7) = 1;
aux1(187,12) = 1;
coef(187,1) = -theta2_dot(t-2);
%188 -a2*a5*b4*theta2_dot(t-3)
aux1(188,2) = 1;
aux1(188,5) = 1;
aux1(188,12) = 1;
coef(188,1) = -theta2_dot(t-3);

```

Appendix

```
%189 -a1*a5*b6*theta2_dot(t-4)
aux1(189,1) = 1;
aux1(189,5) = 1;
aux1(189,14) = 1;
coef(189,1) = -theta2_dot(t-4);
%190 -a1*a6*b5*theta2_dot(t-4)
aux1(190,1) = 1;
aux1(190,6) = 1;
aux1(190,13) = 1;
coef(190,1) = -theta2_dot(t-4);
%191 -a1*a7*b5*theta2_dot(t-3)
aux1(191,1) = 1;
aux1(191,7) = 1;
aux1(191,13) = 1;
coef(191,1) = -theta2_dot(t-3);
%192 -a1*a8*b4*theta2_dot(t-3)
aux1(192,1) = 1;
aux1(192,8) = 1;
aux1(192,12) = 1;
coef(192,1) = -theta2_dot(t-3);
%193 -a2*a5*b5*theta2_dot(t-4)
aux1(193,2) = 1;
aux1(193,5) = 1;
aux1(193,13) = 1;
coef(193,1) = -theta2_dot(t-4);
%194 -a2*a6*b4*theta2_dot(t-4)
aux1(194,2) = 1;
aux1(194,6) = 1;
aux1(194,12) = 1;
coef(194,1) = -theta2_dot(t-4);
%195 -a2*a7*b4*theta2_dot(t-3)
aux1(195,2) = 1;
aux1(195,7) = 1;
aux1(195,12) = 1;
coef(195,1) = -theta2_dot(t-3);
%196 -a1*a6*b6*theta2_dot(t-5)
aux1(196,1) = 1;
aux1(196,6) = 1;
aux1(196,14) = 1;
coef(196,1) = -theta2_dot(t-5);
%197 -a1*a7*b6*theta2_dot(t-4)
aux1(197,1) = 1;
aux1(197,7) = 1;
aux1(197,14) = 1;
coef(197,1) = -theta2_dot(t-4);
%198 -a1*a8*b5*theta2_dot(t-4)
aux1(198,1) = 1;
aux1(198,8) = 1;
aux1(198,13) = 1;
coef(198,1) = -theta2_dot(t-4);
%199 -a2*a5*b6*theta2_dot(t-5)
aux1(199,2) = 1;
aux1(199,5) = 1;
aux1(199,14) = 1;
coef(199,1) = -theta2_dot(t-5);
%200 -a2*a6*b5*theta2_dot(t-5)
aux1(200,2) = 1;
```

Appendix

```
aux1(200,6) = 1;
aux1(200,13) = 1;
coef(200,1) = -theta2_dot(t-5);
%201 -a2*a7*b5*theta2_dot(t-4)
aux1(201,2) = 1;
aux1(201,7) = 1;
aux1(201,13) = 1;
coef(201,1) = -theta2_dot(t-4);
%202 -a2*a8*b4*theta2_dot(t-4)
aux1(202,2) = 1;
aux1(202,8) = 1;
aux1(202,12) = 1;
coef(202,1) = -theta2_dot(t-4);
%203 -a5*a7*b4*theta2_dot(t-2)
aux1(203,5) = 1;
aux1(203,7) = 1;
aux1(203,12) = 1;
coef(203,1) = -theta2_dot(t-2);
%204 -a1*a8*b6*theta2_dot(t-5)
aux1(204,1) = 1;
aux1(204,8) = 1;
aux1(204,14) = 1;
coef(204,1) = -theta2_dot(t-5);
%205 -a2*a6*b6*theta2_dot(t-6)
aux1(205,2) = 1;
aux1(205,6) = 1;
aux1(205,14) = 1;
coef(205,1) = -theta2_dot(t-6);
%206 -a2*a7*b6*theta2_dot(t-5)
aux1(206,2) = 1;
aux1(206,7) = 1;
aux1(206,14) = 1;
coef(206,1) = -theta2_dot(t-5);
%207 -a2*a8*b5*theta2_dot(t-5)
aux1(207,2) = 1;
aux1(207,8) = 1;
aux1(207,13) = 1;
coef(207,1) = -theta2_dot(t-5);
%208 -a5*a7*b5*theta2_dot(t-3)
aux1(208,5) = 1;
aux1(208,7) = 1;
aux1(208,13) = 1;
coef(208,1) = -theta2_dot(t-3);
%209 -a5*a8*b4*theta2_dot(t-3)
aux1(209,5) = 1;
aux1(209,8) = 1;
aux1(209,12) = 1;
coef(209,1) = -theta2_dot(t-3);
%210 -a6*a7*b4*theta2_dot(t-3)
aux1(210,6) = 1;
aux1(210,7) = 1;
aux1(210,12) = 1;
coef(210,1) = -theta2_dot(t-3);
%211 -a2*a8*b6*theta2_dot(t-6)
aux1(211,2) = 1;
aux1(211,8) = 1;
aux1(211,14) = 1;
```

```

coef(211,1) = -theta2_dot(t-6);
%212 -a5*a7*b6*theta2_dot(t-4)
aux1(212,5) = 1;
aux1(212,7) = 1;
aux1(212,14) = 1;
coef(212,1) = -theta2_dot(t-4);
%213 -a5*a8*b5*theta2_dot(t-4)
aux1(213,5) = 1;
aux1(213,8) = 1;
aux1(213,13) = 1;
coef(213,1) = -theta2_dot(t-4);
%214 -a6*a7*b5*theta2_dot(t-4)
aux1(214,6) = 1;
aux1(214,7) = 1;
aux1(214,13) = 1;
coef(214,1) = -theta2_dot(t-4);
%215 -a6*a8*b4*theta2_dot(t-4)
aux1(215,6) = 1;
aux1(215,8) = 1;
aux1(215,12) = 1;
coef(215,1) = -theta2_dot(t-4);
%216 -a5*a8*b6*theta2_dot(t-5)
aux1(216,5) = 1;
aux1(216,8) = 1;
aux1(216,14) = 1;
coef(216,1) = -theta2_dot(t-5);
%217 -a6*a7*b6*theta2_dot(t-5)
aux1(217,6) = 1;
aux1(217,7) = 1;
aux1(217,14) = 1;
coef(217,1) = -theta2_dot(t-5);
%218 -a6*a8*b5*theta2_dot(t-5)
aux1(218,6) = 1;
aux1(218,8) = 1;
aux1(218,13) = 1;
coef(218,1) = -theta2_dot(t-5);
%219 -a6*a8*b6*theta2_dot(t-6)
aux1(219,6) = 1;
aux1(219,8) = 1;
aux1(219,14) = 1;
coef(219,1) = -theta2_dot(t-6);
%220 -a1*a5*a7*b4*theta2_dot(t-3)
aux1(220,1) = 1;
aux1(220,5) = 1;
aux1(220,7) = 1;
aux1(220,12) = 1;
coef(220,1) = -theta2_dot(t-3);
%221 -a1*a5*a7*b5*theta2_dot(t-4)
aux1(221,1) = 1;
aux1(221,5) = 1;
aux1(221,7) = 1;
aux1(221,13) = 1;
coef(221,1) = -theta2_dot(t-4);
%222 -a1*a5*a8*b4*theta2_dot(t-4)
aux1(222,1) = 1;
aux1(222,5) = 1;
aux1(222,8) = 1;

```

Appendix

```
aux1(222,12) = 1;
coef(222,1) = -theta2_dot(t-4);
%223 -a1*a6*a7*b4*theta2_dot(t-4)
aux1(223,1) = 1;
aux1(223,6) = 1;
aux1(223,7) = 1;
aux1(223,12) = 1;
coef(223,1) = -theta2_dot(t-4);
%224 -a2*a5*a7*b4*theta2_dot(t-4)
aux1(224,2) = 1;
aux1(224,5) = 1;
aux1(224,7) = 1;
aux1(224,12) = 1;
coef(224,1) = -theta2_dot(t-4);
%225 -a1*a5*a7*b6*theta2_dot(t-5)
aux1(225,1) = 1;
aux1(225,5) = 1;
aux1(225,7) = 1;
aux1(225,14) = 1;
coef(225,1) = -theta2_dot(t-5);
%226 -a1*a5*a8*b5*theta2_dot(t-5)
aux1(226,1) = 1;
aux1(226,5) = 1;
aux1(226,8) = 1;
aux1(226,13) = 1;
coef(226,1) = -theta2_dot(t-5);
%227 -a1*a6*a7*b5*theta2_dot(t-5)
aux1(227,1) = 1;
aux1(227,6) = 1;
aux1(227,7) = 1;
aux1(227,13) = 1;
coef(227,1) = -theta2_dot(t-5);
%228 -a1*a6*a8*b4*theta2_dot(t-5)
aux1(228,1) = 1;
aux1(228,6) = 1;
aux1(228,8) = 1;
aux1(228,12) = 1;
coef(228,1) = -theta2_dot(t-5);
%229 -a2*a5*a7*b5*theta2_dot(t-5)
aux1(229,2) = 1;
aux1(229,5) = 1;
aux1(229,7) = 1;
aux1(229,13) = 1;
coef(229,1) = -theta2_dot(t-5);
%230 -a2*a5*a8*b4*theta2_dot(t-5)
aux1(230,2) = 1;
aux1(230,5) = 1;
aux1(230,8) = 1;
aux1(230,12) = 1;
coef(230,1) = -theta2_dot(t-5);
%231 -a2*a6*a7*b4*theta2_dot(t-5)
aux1(231,2) = 1;
aux1(231,6) = 1;
aux1(231,7) = 1;
aux1(231,12) = 1;
coef(231,1) = -theta2_dot(t-5);
%232 -a1*a5*a8*b6*theta2_dot(t-6)
```

Appendix

```
aux1(232,1) = 1;
aux1(232,5) = 1;
aux1(232,8) = 1;
aux1(232,14) = 1;
coef(232,1) = -theta2_dot(t-6);
%233 -a1*a6*a7*b6*theta2_dot(t-6)
aux1(233,1) = 1;
aux1(233,6) = 1;
aux1(233,7) = 1;
aux1(233,14) = 1;
coef(233,1) = -theta2_dot(t-6);
%234 -a1*a6*a8*b5*theta2_dot(t-6)
aux1(234,1) = 1;
aux1(234,6) = 1;
aux1(234,8) = 1;
aux1(234,13) = 1;
coef(234,1) = -theta2_dot(t-6);
%235 -a2*a5*a7*b6*theta2_dot(t-6)
aux1(235,2) = 1;
aux1(235,5) = 1;
aux1(235,7) = 1;
aux1(235,14) = 1;
coef(235,1) = -theta2_dot(t-6);
%236 -a2*a5*a8*b5*theta2_dot(t-6)
aux1(236,2) = 1;
aux1(236,5) = 1;
aux1(236,8) = 1;
aux1(236,13) = 1;
coef(236,1) = -theta2_dot(t-6);
%237 -a2*a6*a7*b5*theta2_dot(t-6)
aux1(237,2) = 1;
aux1(237,6) = 1;
aux1(237,7) = 1;
aux1(237,13) = 1;
coef(237,1) = -theta2_dot(t-6);
%238 -a2*a6*a8*b4*theta2_dot(t-6)
aux1(238,2) = 1;
aux1(238,6) = 1;
aux1(238,8) = 1;
aux1(238,12) = 1;
coef(238,1) = -theta2_dot(t-6);
%239 -a1*a6*a8*b6*theta2_dot(t-7)
aux1(239,1) = 1;
aux1(239,6) = 1;
aux1(239,8) = 1;
aux1(239,14) = 1;
coef(239,1) = -theta2_dot(t-7);
%240 -a2*a5*a8*b6*theta2_dot(t-7)
aux1(240,2) = 1;
aux1(240,5) = 1;
aux1(240,8) = 1;
aux1(240,14) = 1;
coef(240,1) = -theta2_dot(t-7);
%241 -a2*a6*a7*b6*theta2_dot(t-7)
aux1(241,2) = 1;
aux1(241,6) = 1;
aux1(241,7) = 1;
```

Appendix

```
aux1(241,14) = 1;
coef(241,1) = -theta2_dot(t-7);
%242 -a2*a6*a8*b5*theta2_dot(t-7)
aux1(242,2) = 1;
aux1(242,6) = 1;
aux1(242,8) = 1;
aux1(242,13) = 1;
coef(242,1) = -theta2_dot(t-7);
%243 -a2*a6*a8*b6*theta2_dot(t-8)
aux1(243,2) = 1;
aux1(243,6) = 1;
aux1(243,8) = 1;
aux1(243,14) = 1;
coef(243,1) = -theta2_dot(t-8);
%theta3
%244 -b7*theta3(t)
aux1(244,15) = 1;
coef(244,1) = -theta3(t);
%245 -b8*theta3(t-1)
aux1(245,16) = 1;
coef(245,1) = -theta3(t-1);
%246 -b9*theta3(t-2)
aux1(246,17) = 1;
coef(246,1) = -theta3(t-2);
%247 -a1*b7*theta3(t-1)
aux1(247,1) = 1;
aux1(247,15) = 1;
coef(247,1) = -theta3(t-1);
%248 -a3*b7*theta3(t-1)
aux1(248,3) = 1;
aux1(248,15) = 1;
coef(248,1) = -theta3(t-1);
%249 -a7*b7*theta3(t-1)
aux1(249,7) = 1;
aux1(249,15) = 1;
coef(249,1) = -theta3(t-1);
%250 -a1*b8*theta3(t-2)
aux1(250,1) = 1;
aux1(250,16) = 1;
coef(250,1) = -theta3(t-2);
%251 -a2*b7*theta3(t-2)
aux1(251,2) = 1;
aux1(251,15) = 1;
coef(251,1) = -theta3(t-2);
%252 -a1*b9*theta3(t-3)
aux1(252,1) = 1;
aux1(252,17) = 1;
coef(252,1) = -theta3(t-3);
%253 -a2*b8*theta3(t-3)
aux1(253,2) = 1;
aux1(253,16) = 1;
coef(253,1) = -theta3(t-3);
%254 -a3*b8*theta3(t-2)
aux1(254,3) = 1;
aux1(254,16) = 1;
coef(254,1) = -theta3(t-2);
%255 -a4*b7*theta3(t-2)
```

Appendix

```
aux1(255,4) = 1;
aux1(255,15) = 1;
coef(255,1) = -theta3(t-2);
%256 -a2*b9*theta3(t-4)
aux1(256,2) = 1;
aux1(256,17) = 1;
coef(256,1) = -theta3(t-4);
%257 - a3*b9*theta3(t-3)
aux1(257,3) = 1;
aux1(257,17) = 1;
coef(257,1) = -theta3(t-3);
%258 -a4*b8*theta3(t-3)
aux1(258,4) = 1;
aux1(258,16) = 1;
coef(258,1) = -theta3(t-3);
%259 -a4*b9*theta3(t-4)
aux1(259,4) = 1;
aux1(259,17) = 1;
coef(259,1) = -theta3(t-4);
%260 -a7*b8*theta3(t-2)
aux1(260,7) = 1;
aux1(260,16) = 1;
coef(260,1) = -theta3(t-2);
%261 -a8*b7*theta3(t-2)
aux1(261,8) = 1;
aux1(261,15) = 1;
coef(261,1) = -theta3(t-2);
%262 -a7*b9*theta3(t-3)
aux1(262,7) = 1;
aux1(262,17) = 1;
coef(262,1) = -theta3(t-3);
%263 -a8*b8*theta3(t-3)
aux1(263,8) = 1;
aux1(263,16) = 1;
coef(263,1) = -theta3(t-3);
%264 -a8*b9*theta3(t-4)
aux1(264,8) = 1;
aux1(264,17) = 1;
coef(264,1) = -theta3(t-4);
%265 -a1*a3*b7*theta3(t-2)
aux1(265,1) = 1;
aux1(265,3) = 1;
aux1(265,15) = 1;
coef(265,1) = -theta3(t-2);
%266 -a1*a3*b8*theta3(t-3)
aux1(266,1) = 1;
aux1(266,3) = 1;
aux1(266,16) = 1;
coef(266,1) = -theta3(t-3);
%267 -a1*a4*b7*theta3(t-3)
aux1(267,1) = 1;
aux1(267,4) = 1;
aux1(267,15) = 1;
coef(267,1) = -theta3(t-3);
%268 -a2*a3*b7*theta3(t-3)
aux1(268,2) = 1;
aux1(268,3) = 1;
```

Appendix

```
aux1(268,15) = 1;
coef(268,1) = -theta3(t-3);
%269 -a1*a3*b9*theta3(t-4)
aux1(269,1) = 1;
aux1(269,3) = 1;
aux1(269,17) = 1;
coef(269,1) = -theta3(t-4);
%270 -a1*a4*b8*theta3(t-4)
aux1(270,1) = 1;
aux1(270,4) = 1;
aux1(270,16) = 1;
coef(270,1) = -theta3(t-4);
%271 -a1*a7*b7*theta3(t-2)
aux1(271,1) = 1;
aux1(271,7) = 1;
aux1(271,15) = 1;
coef(271,1) = -theta3(t-2);
%272 -a2*a3*b8*theta3(t-4)
aux1(272,2) = 1;
aux1(272,3) = 1;
aux1(272,16) = 1;
coef(272,1) = -theta3(t-4);
%273 -a2*a4*b7*theta3(t-4)
aux1(273,2) = 1;
aux1(273,4) = 1;
aux1(273,15) = 1;
coef(273,1) = -theta3(t-4);
%274 -a1*a4*b9*theta3(t-5)
aux1(274,1) = 1;
aux1(274,4) = 1;
aux1(274,17) = 1;
coef(274,1) = -theta3(t-5);
%275 -a1*a7*b8*theta3(t-3)
aux1(275,1) = 1;
aux1(275,7) = 1;
aux1(275,16) = 1;
coef(275,1) = -theta3(t-3);
%276 -a1*a8*b7*theta3(t-3)
aux1(276,1) = 1;
aux1(276,8) = 1;
aux1(276,15) = 1;
coef(276,1) = -theta3(t-3);
%277 - a2*a3*b9*theta3(t-5)
aux1(277,2) = 1;
aux1(277,3) = 1;
aux1(277,17) = 1;
coef(277,1) = -theta3(t-5);
%278 -a2*a4*b8*theta3(t-5)
aux1(278,2) = 1;
aux1(278,4) = 1;
aux1(278,16) = 1;
coef(278,1) = -theta3(t-5);
%279 -a2*a7*b7*theta3(t-3)
aux1(279,2) = 1;
aux1(279,7) = 1;
aux1(279,15) = 1;
coef(279,1) = -theta3(t-3);
```

Appendix

```
%280 -a3*a7*b7*theta3(t-2)
aux1(280,3) = 1;
aux1(280,7) = 1;
aux1(280,15) = 1;
coef(280,1) = -theta3(t-2);
%281 -a1*a7*b9*theta3(t-4)
aux1(281,1) = 1;
aux1(281,7) = 1;
aux1(281,17) = 1;
coef(281,1) = -theta3(t-4);
%282 -a1*a8*b8*theta3(t-4)
aux1(282,1) = 1;
aux1(282,8) = 1;
aux1(282,16) = 1;
coef(282,1) = -theta3(t-4);
%283 -a2*a4*b9*theta3(t-6)
aux1(283,2) = 1;
aux1(283,4) = 1;
aux1(283,17) = 1;
coef(283,1) = -theta3(t-6);
%284 -a2*a7*b8*theta3(t-4)
aux1(284,2) = 1;
aux1(284,7) = 1;
aux1(284,16) = 1;
coef(284,1) = -theta3(t-4);
%285 -a2*a8*b7*theta3(t-4)
aux1(285,2) = 1;
aux1(285,8) = 1;
aux1(285,15) = 1;
coef(285,1) = -theta3(t-4);
%286 -a3*a7*b8*theta3(t-3)
aux1(286,3) = 1;
aux1(286,7) = 1;
aux1(286,16) = 1;
coef(286,1) = -theta3(t-3);
%287 -a3*a8*b7*theta3(t-3)
aux1(287,3) = 1;
aux1(287,8) = 1;
aux1(287,15) = 1;
coef(287,1) = -theta3(t-3);
%288 -a4*a7*b7*theta3(t-3)
aux1(288,4) = 1;
aux1(288,7) = 1;
aux1(288,15) = 1;
coef(288,1) = -theta3(t-3);
%289 -a1*a8*b9*theta3(t-5)
aux1(289,1) = 1;
aux1(289,8) = 1;
aux1(289,17) = 1;
coef(289,1) = -theta3(t-5);
%290 -a2*a7*b9*theta3(t-5)
aux1(290,2) = 1;
aux1(290,7) = 1;
aux1(290,17) = 1;
coef(290,1) = -theta3(t-5);
%291 -a2*a8*b8*theta3(t-5)
aux1(291,2) = 1;
```

Appendix

```
aux1(291,8) = 1;
aux1(291,16) = 1;
coef(291,1) = -theta3(t-5);
%292 -a3*a7*b9*theta3(t-4)
aux1(292,3) = 1;
aux1(292,7) = 1;
aux1(292,17) = 1;
coef(292,1) = -theta3(t-4);
%293 -a3*a8*b8*theta3(t-4)
aux1(293,3) = 1;
aux1(293,8) = 1;
aux1(293,16) = 1;
coef(293,1) = -theta3(t-4);
%294 -a4*a7*b8*theta3(t-4)
aux1(294,4) = 1;
aux1(294,7) = 1;
aux1(294,16) = 1;
coef(294,1) = -theta3(t-4);
%295 -a4*a8*b7*theta3(t-4)
aux1(295,4) = 1;
aux1(295,8) = 1;
aux1(295,15) = 1;
coef(295,1) = -theta3(t-4);
%296 -a2*a8*b9*theta3(t-6)
aux1(296,2) = 1;
aux1(296,8) = 1;
aux1(296,17) = 1;
coef(296,1) = -theta3(t-6);
%297 -a3*a8*b9*theta3(t-5)
aux1(297,3) = 1;
aux1(297,8) = 1;
aux1(297,17) = 1;
coef(297,1) = -theta3(t-5);
%298 -a4*a7*b9*theta3(t-5)
aux1(298,4) = 1;
aux1(298,7) = 1;
aux1(298,17) = 1;
coef(298,1) = -theta3(t-5);
%299 -a4*a8*b8*theta3(t-5)
aux1(299,4) = 1;
aux1(299,8) = 1;
aux1(299,16) = 1;
coef(299,1) = -theta3(t-5);
%300 -a4*a8*b9*theta3(t-6)
aux1(300,4) = 1;
aux1(300,8) = 1;
aux1(300,17) = 1;
coef(300,1) = -theta3(t-6);
%301 -a1*a3*a7*b7*theta3(t-3)
aux1(301,1) = 1;
aux1(301,3) = 1;
aux1(301,7) = 1;
aux1(301,15) = 1;
coef(301,1) = -theta3(t-3);
%302 -a1*a3*a7*b8*theta3(t-4)
aux1(302,1) = 1;
aux1(302,3) = 1;
```

Appendix

```
aux1(302,7) = 1;
aux1(302,16) = 1;
coef(302,1) = -theta3(t-4);
%303 -a1*a3*a8*b7*theta3(t-4)
aux1(303,1) = 1;
aux1(303,3) = 1;
aux1(303,8) = 1;
aux1(303,15) = 1;
coef(303,1) = -theta3(t-4);
%304 -a1*a4*a7*b7*theta3(t-4)
aux1(304,1) = 1;
aux1(304,4) = 1;
aux1(304,7) = 1;
aux1(304,15) = 1;
coef(304,1) = -theta3(t-4);
%305 -a2*a3*a7*b7*theta3(t-4)
aux1(305,2) = 1;
aux1(305,3) = 1;
aux1(305,7) = 1;
aux1(305,15) = 1;
coef(305,1) = -theta3(t-4);
%306 -a1*a3*a7*b9*theta3(t-5)
aux1(306,1) = 1;
aux1(306,3) = 1;
aux1(306,7) = 1;
aux1(306,17) = 1;
coef(306,1) = -theta3(t-5);
%307 -a1*a3*a8*b8*theta3(t-5)
aux1(307,1) = 1;
aux1(307,3) = 1;
aux1(307,8) = 1;
aux1(307,16) = 1;
coef(307,1) = -theta3(t-5);
%308 - a1*a4*a7*b8*theta3(t-5)
aux1(308,1) = 1;
aux1(308,4) = 1;
aux1(308,7) = 1;
aux1(308,16) = 1;
coef(308,1) = -theta3(t-5);
%309 -a1*a4*a8*b7*theta3(t-5)
aux1(309,1) = 1;
aux1(309,4) = 1;
aux1(309,8) = 1;
aux1(309,15) = 1;
coef(309,1) = -theta3(t-5);
%310 -a2*a3*a7*b8*theta3(t-5)
aux1(310,2) = 1;
aux1(310,3) = 1;
aux1(310,7) = 1;
aux1(310,16) = 1;
coef(310,1) = -theta3(t-5);
%311 -a2*a3*a8*b7*theta3(t-5)
aux1(311,2) = 1;
aux1(311,3) = 1;
aux1(311,8) = 1;
aux1(311,15) = 1;
coef(311,1) = -theta3(t-5);
```

Appendix

```
%312 -a2*a4*a7*b7*theta3(t-5)
aux1(312,2) = 1;
aux1(312,4) = 1;
aux1(312,7) = 1;
aux1(312,15) = 1;
coef(312,1) = -theta3(t-5);
%313 -a1*a3*a8*b9*theta3(t-6)
aux1(313,1) = 1;
aux1(313,3) = 1;
aux1(313,8) = 1;
aux1(313,17) = 1;
coef(313,1) = -theta3(t-6);
%314 -a1*a4*a7*b9*theta3(t-6)
aux1(314,1) = 1;
aux1(314,4) = 1;
aux1(314,7) = 1;
aux1(314,17) = 1;
coef(314,1) = -theta3(t-6);
%315 -a1*a4*a8*b8*theta3(t-6)
aux1(315,1) = 1;
aux1(315,4) = 1;
aux1(315,8) = 1;
aux1(315,16) = 1;
coef(315,1) = -theta3(t-6);
%316 -a2*a3*a7*b9*theta3(t-6)
aux1(316,2) = 1;
aux1(316,3) = 1;
aux1(316,7) = 1;
aux1(316,17) = 1;
coef(316,1) = -theta3(t-6);
%317 -a2*a3*a8*b8*theta3(t-6)
aux1(317,2) = 1;
aux1(317,3) = 1;
aux1(317,8) = 1;
aux1(317,16) = 1;
coef(317,1) = -theta3(t-6);
%318 -a2*a4*a7*b8*theta3(t-6)
aux1(318,2) = 1;
aux1(318,4) = 1;
aux1(318,7) = 1;
aux1(318,16) = 1;
coef(318,1) = -theta3(t-6);
%319 -a2*a4*a8*b7*theta3(t-6)
aux1(319,2) = 1;
aux1(319,4) = 1;
aux1(319,8) = 1;
aux1(319,15) = 1;
coef(319,1) = -theta3(t-6);
%320 -a1*a4*a8*b9*theta3(t-7)
aux1(320,1) = 1;
aux1(320,4) = 1;
aux1(320,8) = 1;
aux1(320,17) = 1;
coef(320,1) = -theta3(t-7);
%321 -a2*a3*a8*b9*theta3(t-7)
aux1(321,2) = 1;
aux1(321,3) = 1;
```

Appendix

```
aux1(321,8) = 1;
aux1(321,17) = 1;
coef(321,1) = -theta3(t-7);
%322 -a2*a4*a7*b9*theta3(t-7)
aux1(322,2) = 1;
aux1(322,4) = 1;
aux1(322,7) = 1;
aux1(322,17) = 1;
coef(322,1) = -theta3(t-7);
%323 -a2*a4*a8*b8*theta3(t-7)
aux1(323,2) = 1;
aux1(323,4) = 1;
aux1(323,8) = 1;
aux1(323,16) = 1;
coef(323,1) = -theta3(t-7);
%324 -a2*a4*a8*b9*theta3(t-8)
aux1(324,2) = 1;
aux1(324,4) = 1;
aux1(324,8) = 1;
aux1(324,17) = 1;
coef(324,1) = -theta3(t-8);
%theta3_dot
%325 -b10*theta3_dot(t)
aux1(325,18) = 1;
coef(325,1) = -theta3_dot(t);
%326 -b11*theta3_dot(t-1)
aux1(326,19) = 1;
coef(326,1) = -theta3_dot(t-1);
%327 -b12*theta3_dot(t-2)
aux1(327,20) = 1;
coef(327,1) = -theta3_dot(t-2);
%328 -a1*b10*theta3_dot(t-1)
aux1(328,1) = 1;
aux1(328,18) = 1;
coef(328,1) = -theta3_dot(t-1);
%329 -a3*b10*theta3_dot(t-1)
aux1(329,3) = 1;
aux1(329,18) = 1;
coef(329,1) = -theta3_dot(t-1);
%330 -a5*b10*theta3_dot(t-1)
aux1(330,5) = 1;
aux1(330,18) = 1;
coef(330,1) = -theta3_dot(t-1);
%331 -a1*b11*theta3_dot(t-2)
aux1(331,1) = 1;
aux1(331,19) = 1;
coef(331,1) = -theta3_dot(t-2);
%332 -a2*b10*theta3_dot(t-2)
aux1(332,2) = 1;
aux1(332,18) = 1;
coef(332,1) = -theta3_dot(t-2);
%333 -a1*b12*theta3_dot(t-3)
aux1(333,1) = 1;
aux1(333,20) = 1;
coef(333,1) = -theta3_dot(t-3);
%334 -a2*b11*theta3_dot(t-3)
aux1(334,2) = 1;
```

Appendix

```
aux1(334,19) = 1;
coef(334,1) = -theta3_dot(t-3);
%335 -a3*b11*theta3_dot(t-2)
aux1(335,3) = 1;
aux1(335,19) = 1;
coef(335,1) = -theta3_dot(t-2);
%336 - a4*b10*theta3_dot(t-2)
aux1(336,4) = 1;
aux1(336,18) = 1;
coef(336,1) = -theta3_dot(t-2);
%337 -a2*b12*theta3_dot(t-4)
aux1(337,2) = 1;
aux1(337,20) = 1;
coef(337,1) = -theta3_dot(t-4);
%338 -a3*b12*theta3_dot(t-3)
aux1(338,3) = 1;
aux1(338,20) = 1;
coef(338,1) = -theta3_dot(t-3);
%339 - a4*b11*theta3_dot(t-3)
aux1(339,4) = 1;
aux1(339,19) = 1;
coef(339,1) = -theta3_dot(t-3);
%340 -a5*b11*theta3_dot(t-2)
aux1(340,5) = 1;
aux1(340,19) = 1;
coef(340,1) = -theta3_dot(t-2);
%341 -a6*b10*theta3_dot(t-2)
aux1(341,6) = 1;
aux1(341,18) = 1;
coef(341,1) = -theta3_dot(t-2);
%342 -a4*b12*theta3_dot(t-4)
aux1(342,4) = 1;
aux1(342,20) = 1;
coef(342,1) = -theta3_dot(t-4);
%343 -a5*b12*theta3_dot(t-3)
aux1(343,5) = 1;
aux1(343,20) = 1;
coef(343,1) = -theta3_dot(t-3);
%344 - a6*b11*theta3_dot(t-3)
aux1(344,6) = 1;
aux1(344,19) = 1;
coef(344,1) = -theta3_dot(t-3);
%345 -a6*b12*theta3_dot(t-4)
aux1(345,6) = 1;
aux1(345,20) = 1;
coef(345,1) = -theta3_dot(t-4);
%346 -a1*a3*b10*theta3_dot(t-2)
aux1(346,1) = 1;
aux1(346,3) = 1;
aux1(346,18) = 1;
coef(346,1) = -theta3_dot(t-2);
%347 -a1*a3*b11*theta3_dot(t-3)
aux1(347,1) = 1;
aux1(347,3) = 1;
aux1(347,19) = 1;
coef(347,1) = -theta3_dot(t-3);
%348 -a1*a4*b10*theta3_dot(t-3)
```

Appendix

```
aux1(348,1) = 1;
aux1(348,4) = 1;
aux1(348,18) = 1;
coef(348,1) = -theta3_dot(t-3);
%349 -a1*a5*b10*theta3_dot(t-2)
aux1(349,1) = 1;
aux1(349,5) = 1;
aux1(349,18) = 1;
coef(349,1) = -theta3_dot(t-2);
%350 -a2*a3*b10*theta3_dot(t-3)
aux1(350,2) = 1;
aux1(350,3) = 1;
aux1(350,18) = 1;
coef(350,1) = -theta3_dot(t-3);
%351 -a1*a3*b12*theta3_dot(t-4)
aux1(351,1) = 1;
aux1(351,3) = 1;
aux1(351,20) = 1;
coef(351,1) = -theta3_dot(t-4);
%352 -a1*a4*b11*theta3_dot(t-4)
aux1(352,1) = 1;
aux1(352,4) = 1;
aux1(352,19) = 1;
coef(352,1) = -theta3_dot(t-4);
%353 -a1*a5*b11*theta3_dot(t-3)
aux1(353,1) = 1;
aux1(353,5) = 1;
aux1(353,19) = 1;
coef(353,1) = -theta3_dot(t-3);
%354 -a1*a6*b10*theta3_dot(t-3)
aux1(354,1) = 1;
aux1(354,6) = 1;
aux1(354,18) = 1;
coef(354,1) = -theta3_dot(t-3);
%355 -a2*a3*b11*theta3_dot(t-4)
aux1(355,2) = 1;
aux1(355,3) = 1;
aux1(355,19) = 1;
coef(355,1) = -theta3_dot(t-4);
%356 -a2*a4*b10*theta3_dot(t-4)
aux1(356,2) = 1;
aux1(356,4) = 1;
aux1(356,18) = 1;
coef(356,1) = -theta3_dot(t-4);
%357 -a2*a5*b10*theta3_dot(t-3)
aux1(357,2) = 1;
aux1(357,5) = 1;
aux1(357,18) = 1;
coef(357,1) = -theta3_dot(t-3);
%358 -a3*a5*b10*theta3_dot(t-2)
aux1(358,3) = 1;
aux1(358,5) = 1;
aux1(358,18) = 1;
coef(358,1) = -theta3_dot(t-2);
%359 -a1*a4*b12*theta3_dot(t-5)
aux1(359,1) = 1;
aux1(359,4) = 1;
```

Appendix

```
aux1(359,20) = 1;
coef(359,1) = -theta3_dot(t-5);
%360 -a1*a5*b12*theta3_dot(t-4)
aux1(360,1) = 1;
aux1(360,5) = 1;
aux1(360,20) = 1;
coef(360,1) = -theta3_dot(t-4);
%361 -a1*a6*b11*theta3_dot(t-4)
aux1(361,1) = 1;
aux1(361,6) = 1;
aux1(361,19) = 1;
coef(361,1) = -theta3_dot(t-4);
%362 -a2*a3*b12*theta3_dot(t-5)
aux1(362,2) = 1;
aux1(362,3) = 1;
aux1(362,20) = 1;
coef(362,1) = -theta3_dot(t-5);
%363 -a2*a4*b11*theta3_dot(t-5)
aux1(363,2) = 1;
aux1(363,4) = 1;
aux1(363,19) = 1;
coef(363,1) = -theta3_dot(t-5);
%364 -a2*a5*b11*theta3_dot(t-4)
aux1(364,2) = 1;
aux1(364,5) = 1;
aux1(364,19) = 1;
coef(364,1) = -theta3_dot(t-4);
%365 -a2*a6*b10*theta3_dot(t-4)
aux1(365,2) = 1;
aux1(365,6) = 1;
aux1(365,18) = 1;
coef(365,1) = -theta3_dot(t-4);
%366 -a3*a5*b11*theta3_dot(t-3)
aux1(366,3) = 1;
aux1(366,5) = 1;
aux1(366,19) = 1;
coef(366,1) = -theta3_dot(t-3);
%367 -a3*a6*b10*theta3_dot(t-3)
aux1(367,3) = 1;
aux1(367,6) = 1;
aux1(367,18) = 1;
coef(367,1) = -theta3_dot(t-3);
%368 -a4*a5*b10*theta3_dot(t-3)
aux1(368,4) = 1;
aux1(368,5) = 1;
aux1(368,18) = 1;
coef(368,1) = -theta3_dot(t-3);
%369 -a1*a6*b12*theta3_dot(t-5)
aux1(369,1) = 1;
aux1(369,6) = 1;
aux1(369,20) = 1;
coef(369,1) = -theta3_dot(t-5);
%370 -a2*a4*b12*theta3_dot(t-6)
aux1(370,2) = 1;
aux1(370,4) = 1;
aux1(370,20) = 1;
coef(370,1) = -theta3_dot(t-6);
```

Appendix

```
%371 -a2*a5*b12*theta3_dot(t-5)
aux1(371,2) = 1;
aux1(371,5) = 1;
aux1(371,20) = 1;
coef(371,1) = -theta3_dot(t-5);
%372 -a2*a6*b11*theta3_dot(t-5)
aux1(372,2) = 1;
aux1(372,6) = 1;
aux1(372,19) = 1;
coef(372,1) = -theta3_dot(t-5);
%373 -a3*a5*b12*theta3_dot(t-4)
aux1(373,3) = 1;
aux1(373,5) = 1;
aux1(373,20) = 1;
coef(373,1) = -theta3_dot(t-4);
%374 -a3*a6*b11*theta3_dot(t-4)
aux1(374,3) = 1;
aux1(374,6) = 1;
aux1(374,19) = 1;
coef(374,1) = -theta3_dot(t-4);
%375 -a4*a5*b11*theta3_dot(t-4)
aux1(375,4) = 1;
aux1(375,5) = 1;
aux1(375,19) = 1;
coef(375,1) = -theta3_dot(t-4);
%376 -a4*a6*b10*theta3_dot(t-4)
aux1(376,4) = 1;
aux1(376,6) = 1;
aux1(376,18) = 1;
coef(376,1) = -theta3_dot(t-4);
%377 -a2*a6*b12*theta3_dot(t-6)
aux1(377,2) = 1;
aux1(377,6) = 1;
aux1(377,20) = 1;
coef(377,1) = -theta3_dot(t-6);
%378 -a3*a6*b12*theta3_dot(t-5)
aux1(378,3) = 1;
aux1(378,6) = 1;
aux1(378,20) = 1;
coef(378,1) = -theta3_dot(t-5);
%379 -a4*a5*b12*theta3_dot(t-5)
aux1(379,4) = 1;
aux1(379,5) = 1;
aux1(379,20) = 1;
coef(379,1) = -theta3_dot(t-5);
%380 -a4*a6*b11*theta3_dot(t-5)
aux1(380,4) = 1;
aux1(380,6) = 1;
aux1(380,19) = 1;
coef(380,1) = -theta3_dot(t-5);
%381 -a4*a6*b12*theta3_dot(t-6)
aux1(381,4) = 1;
aux1(381,6) = 1;
aux1(381,20) = 1;
coef(381,1) = -theta3_dot(t-6);
%382 -a1*a3*a5*b10*theta3_dot(t-3)
aux1(382,1) = 1;
```

Appendix

```
aux1(382,3) = 1;
aux1(382,5) = 1;
aux1(382,18) = 1;
coef(382,1) = -theta3_dot(t-3);
%383 -a1*a3*a5*b11*theta3_dot(t-4)
aux1(383,1) = 1;
aux1(383,3) = 1;
aux1(383,5) = 1;
aux1(383,19) = 1;
coef(383,1) = -theta3_dot(t-4);
%384 -a1*a3*a6*b10*theta3_dot(t-4)
aux1(384,1) = 1;
aux1(384,3) = 1;
aux1(384,6) = 1;
aux1(384,18) = 1;
coef(384,1) = -theta3_dot(t-4);
%385 -a1*a4*a5*b10*theta3_dot(t-4)
aux1(385,1) = 1;
aux1(385,4) = 1;
aux1(385,5) = 1;
aux1(385,18) = 1;
coef(385,1) = -theta3_dot(t-4);
%386 -a2*a3*a5*b10*theta3_dot(t-4)
aux1(386,2) = 1;
aux1(386,3) = 1;
aux1(386,5) = 1;
aux1(386,18) = 1;
coef(386,1) = -theta3_dot(t-4);
%387 -a1*a3*a5*b12*theta3_dot(t-5)
aux1(387,1) = 1;
aux1(387,3) = 1;
aux1(387,5) = 1;
aux1(387,20) = 1;
coef(387,1) = -theta3_dot(t-5);
%388 -a1*a3*a6*b11*theta3_dot(t-5)
aux1(388,1) = 1;
aux1(388,3) = 1;
aux1(388,6) = 1;
aux1(388,19) = 1;
coef(388,1) = -theta3_dot(t-5);
%389 -a1*a4*a5*b11*theta3_dot(t-5)
aux1(389,1) = 1;
aux1(389,4) = 1;
aux1(389,5) = 1;
aux1(389,19) = 1;
coef(389,1) = -theta3_dot(t-5);
%390 -a1*a4*a6*b10*theta3_dot(t-5)
aux1(390,1) = 1;
aux1(390,4) = 1;
aux1(390,6) = 1;
aux1(390,18) = 1;
coef(390,1) = -theta3_dot(t-5);
%391 -a2*a3*a5*b11*theta3_dot(t-5)
aux1(391,2) = 1;
aux1(391,3) = 1;
aux1(391,5) = 1;
aux1(391,19) = 1;
```

```

coef(391,1) = -theta3_dot(t-5);
%392 -a2*a3*a6*b10*theta3_dot(t-5)
aux1(392,2) = 1;
aux1(392,3) = 1;
aux1(392,6) = 1;
aux1(392,18) = 1;
coef(392,1) = -theta3_dot(t-5);
%393 -a2*a4*a5*b10*theta3_dot(t-5)
aux1(393,2) = 1;
aux1(393,4) = 1;
aux1(393,5) = 1;
aux1(393,18) = 1;
coef(393,1) = -theta3_dot(t-5);
%394 -a1*a3*a6*b12*theta3_dot(t-6)
aux1(394,1) = 1;
aux1(394,3) = 1;
aux1(394,6) = 1;
aux1(394,20) = 1;
coef(394,1) = -theta3_dot(t-6);
%395 -a1*a4*a5*b12*theta3_dot(t-6)
aux1(395,1) = 1;
aux1(395,4) = 1;
aux1(395,5) = 1;
aux1(395,20) = 1;
coef(395,1) = -theta3_dot(t-6);
%396 -a1*a4*a6*b11*theta3_dot(t-6)
aux1(396,1) = 1;
aux1(396,4) = 1;
aux1(396,6) = 1;
aux1(396,19) = 1;
coef(396,1) = -theta3_dot(t-6);
%397 -a2*a3*a5*b12*theta3_dot(t-6)
aux1(397,2) = 1;
aux1(397,3) = 1;
aux1(397,5) = 1;
aux1(397,20) = 1;
coef(397,1) = -theta3_dot(t-6);
%398 -a2*a3*a6*b11*theta3_dot(t-6)
aux1(398,2) = 1;
aux1(398,3) = 1;
aux1(398,6) = 1;
aux1(398,19) = 1;
coef(398,1) = -theta3_dot(t-6);
%399 -a2*a4*a5*b11*theta3_dot(t-6)
aux1(399,2) = 1;
aux1(399,4) = 1;
aux1(399,5) = 1;
aux1(399,19) = 1;
coef(399,1) = -theta3_dot(t-6);
%400 -a2*a4*a6*b10*theta3_dot(t-6)
aux1(400,2) = 1;
aux1(400,4) = 1;
aux1(400,6) = 1;
aux1(400,18) = 1;
coef(400,1) = -theta3_dot(t-6);
%401 -a1*a4*a6*b12*theta3_dot(t-7)
aux1(401,1) = 1;

```

Appendix

```
aux1(401,4) = 1;
aux1(401,6) = 1;
aux1(401,20) = 1;
coef(401,1) = -theta3_dot(t-7);
%402 -a2*a3*a6*b12*theta3_dot(t-7)
aux1(402,2) = 1;
aux1(402,3) = 1;
aux1(402,6) = 1;
aux1(402,20) = 1;
coef(402,1) = -theta3_dot(t-7);
%403 -a2*a4*a5*b12*theta3_dot(t-7)
aux1(403,2) = 1;
aux1(403,4) = 1;
aux1(403,5) = 1;
aux1(403,20) = 1;
coef(403,1) = -theta3_dot(t-7);
%404 -a2*a4*a6*b11*theta3_dot(t-7)
aux1(404,2) = 1;
aux1(404,4) = 1;
aux1(404,6) = 1;
aux1(404,19) = 1;
coef(404,1) = -theta3_dot(t-7);
%405 -a2*a4*a6*b12*theta3_dot(t-8)
aux1(405,2) = 1;
aux1(405,4) = 1;
aux1(405,6) = 1;
aux1(405,20) = 1;
coef(405,1) = -theta3_dot(t-8);
% eta
%406 a1*eta(t-1)
aux1(406,1) = 1;
aux1(406,20+t-1) = 1;
coef(406,1) = -1;
%407 a3*eta(t-1)
aux1(407,3) = 1;
aux1(407,20+t-1) = 1;
coef(407,1) = -1;
%408 a5*eta(t-1)
aux1(408,5) = 1;
aux1(408,20+t-1) = 1;
coef(408,1) = -1;
%409 a7*eta(t-1)
aux1(409,7) = 1;
aux1(409,20+t-1) = 1;
coef(409,1) = -1;
%410 a2*eta(t-2)
aux1(410,2) = 1;
aux1(410,20+t-2) = 1;
coef(410,1) = -1;
%411 a4*eta(t-2)
aux1(411,4) = 1;
aux1(411,20+t-2) = 1;
coef(411,1) = -1;
%412 a6*eta(t-2)
aux1(412,6) = 1;
aux1(412,20+t-2) = 1;
coef(412,1) = -1;
```

Appendix

```
%413 a8*eta(t-2)
aux1(413,8) = 1;
aux1(413,20+t-2) = 1;
coef(413,1) = -1;
%414 a1*a3*eta(t-2)
aux1(414,1) = 1;
aux1(414,3) = 1;
aux1(414,20+t-2) = 1;
coef(414,1) = -1;
%415 a1*a4*eta(t-3)
aux1(415,1) = 1;
aux1(415,4) = 1;
aux1(415,20+t-3) = 1;
coef(415,1) = -1;
%416 a1*a5*eta(t-2)
aux1(416,1) = 1;
aux1(416,5) = 1;
aux1(416,20+t-2) = 1;
coef(416,1) = -1;
%417 a2*a3*eta(t-3)
aux1(417,2) = 1;
aux1(417,3) = 1;
aux1(417,20+t-3) = 1;
coef(417,1) = -1;
%418 a1*a6*eta(t-3)
aux1(418,1) = 1;
aux1(418,6) = 1;
aux1(418,20+t-3) = 1;
coef(418,1) = -1;
%419 a1*a7*eta(t-2)
aux1(419,1) = 1;
aux1(419,7) = 1;
aux1(419,20+t-2) = 1;
coef(419,1) = -1;
%420 a2*a4*eta(t-4)
aux1(420,2) = 1;
aux1(420,4) = 1;
aux1(420,20+t-4) = 1;
coef(420,1) = -1;
%421 a2*a5*eta(t-3)
aux1(421,2) = 1;
aux1(421,5) = 1;
aux1(421,20+t-3) = 1;
coef(421,1) = -1;
%422 a3*a5*eta(t-2)
aux1(422,3) = 1;
aux1(422,5) = 1;
aux1(422,20+t-2) = 1;
coef(422,1) = -1;
%423 a1*a8*eta(t-3)
aux1(423,1) = 1;
aux1(423,8) = 1;
aux1(423,20+t-3) = 1;
coef(423,1) = -1;
%424 a2*a6*eta(t-4)
aux1(424,2) = 1;
aux1(424,6) = 1;
```

Appendix

```
aux1(424,20+t-4) = 1;
coef(424,1) = -1;
%425 a2*a7*eta(t-3)
aux1(425,2) = 1;
aux1(425,7) = 1;
aux1(425,20+t-3) = 1;
coef(425,1) = -1;
%426 a3*a6*eta(t-3)
aux1(426,3) = 1;
aux1(426,6) = 1;
aux1(426,20+t-3) = 1;
coef(426,1) = -1;
%427 a3*a7*eta(t-2)
aux1(427,3) = 1;
aux1(427,7) = 1;
aux1(427,20+t-2) = 1;
coef(427,1) = -1;
%428 a4*a5*eta(t-3)
aux1(428,4) = 1;
aux1(428,5) = 1;
aux1(428,20+t-3) = 1;
coef(428,1) = -1;
%429 a2*a8*eta(t-4)
aux1(429,2) = 1;
aux1(429,8) = 1;
aux1(429,20+t-4) = 1;
coef(429,1) = -1;
%430 a3*a8*eta(t-3)
aux1(430,3) = 1;
aux1(430,8) = 1;
aux1(430,20+t-3) = 1;
coef(430,1) = -1;
%431 a4*a6*eta(t-4)
aux1(431,4) = 1;
aux1(431,6) = 1;
aux1(431,20+t-4) = 1;
coef(431,1) = -1;
%432 a4*a7*eta(t-3)
aux1(432,4) = 1;
aux1(432,7) = 1;
aux1(432,20+t-3) = 1;
coef(432,1) = -1;
%433 a5*a7*eta(t-2)
aux1(433,5) = 1;
aux1(433,7) = 1;
aux1(433,20+t-2) = 1;
coef(433,1) = -1;
%434 a4*a8*eta(t-4)
aux1(434,4) = 1;
aux1(434,8) = 1;
aux1(434,20+t-4) = 1;
coef(434,1) = -1;
%435 a5*a8*eta(t-3)
aux1(435,5) = 1;
aux1(435,8) = 1;
aux1(435,20+t-3) = 1;
coef(435,1) = -1;
```

Appendix

```
%436 a6*a7*eta(t-3)
aux1(436,6) = 1;
aux1(436,7) = 1;
aux1(436,20+t-3) = 1;
coef(436,1) = -1;
%437 a6*a8*eta(t-4)
aux1(437,6) = 1;
aux1(437,8) = 1;
aux1(437,20+t-4) = 1;
coef(437,1) = -1;
%438 a1*a3*a5*eta(t-3)
aux1(438,1) = 1;
aux1(438,3) = 1;
aux1(438,5) = 1;
aux1(438,20+t-3) = 1;
coef(438,1) = -1;
%439 a1*a3*a6*eta(t-4)
aux1(439,1) = 1;
aux1(439,3) = 1;
aux1(439,6) = 1;
aux1(439,20+t-4) = 1;
coef(439,1) = -1;
%440 a1*a3*a7*eta(t-3)
aux1(440,1) = 1;
aux1(440,3) = 1;
aux1(440,7) = 1;
aux1(440,20+t-3) = 1;
coef(440,1) = -1;
%441 a1*a4*a5*eta(t-4)
aux1(441,1) = 1;
aux1(441,4) = 1;
aux1(441,5) = 1;
aux1(441,20+t-4) = 1;
coef(441,1) = -1;
%442 a2*a3*a5*eta(t-4)
aux1(442,2) = 1;
aux1(442,3) = 1;
aux1(442,5) = 1;
aux1(442,20+t-4) = 1;
coef(442,1) = -1;
%443 a1*a3*a8*eta(t-4)
aux1(443,1) = 1;
aux1(443,3) = 1;
aux1(443,8) = 1;
aux1(443,20+t-4) = 1;
coef(443,1) = -1;
%444 a1*a4*a6*eta(t-5)
aux1(444,1) = 1;
aux1(444,4) = 1;
aux1(444,6) = 1;
aux1(444,20+t-5) = 1;
coef(444,1) = -1;
%445 a1*a4*a7*eta(t-4)
aux1(445,1) = 1;
aux1(445,4) = 1;
aux1(445,7) = 1;
aux1(445,20+t-4) = 1;
```

Appendix

```
coef(445,1) = -1;
%446 a1*a5*a7*eta(t-3)
aux1(446,1) = 1;
aux1(446,5) = 1;
aux1(446,7) = 1;
aux1(446,20+t-3) = 1;
coef(446,1) = -1;
%447 a2*a3*a6*eta(t-5)
aux1(447,2) = 1;
aux1(447,3) = 1;
aux1(447,6) = 1;
aux1(447,20+t-5) = 1;
coef(447,1) = -1;
%448 a2*a3*a7*eta(t-4)
aux1(448,2) = 1;
aux1(448,3) = 1;
aux1(448,7) = 1;
aux1(448,20+t-4) = 1;
coef(448,1) = -1;
%449 a2*a4*a5*eta(t-5)
aux1(449,2) = 1;
aux1(449,4) = 1;
aux1(449,5) = 1;
aux1(449,20+t-5) = 1;
coef(449,1) = -1;
%450 a1*a4*a8*eta(t-5)
aux1(450,1) = 1;
aux1(450,4) = 1;
aux1(450,8) = 1;
aux1(450,20+t-5) = 1;
coef(450,1) = -1;
%451 a1*a5*a8*eta(t-4)
aux1(451,1) = 1;
aux1(451,5) = 1;
aux1(451,8) = 1;
aux1(451,20+t-4) = 1;
coef(451,1) = -1;
%452 a1*a6*a7*eta(t-4)
aux1(452,1) = 1;
aux1(452,6) = 1;
aux1(452,7) = 1;
aux1(452,20+t-4) = 1;
coef(452,1) = -1;
%453 a2*a3*a8*eta(t-5)
aux1(453,2) = 1;
aux1(453,3) = 1;
aux1(453,8) = 1;
aux1(453,20+t-5) = 1;
coef(453,1) = -1;
%454 a2*a4*a6*eta(t-6)
aux1(454,2) = 1;
aux1(454,4) = 1;
aux1(454,6) = 1;
aux1(454,20+t-6) = 1;
coef(454,1) = -1;
%455 a2*a4*a7*eta(t-5)
aux1(455,2) = 1;
```

Appendix

```
aux1(455,4) = 1;
aux1(455,7) = 1;
aux1(455,20+t-5) = 1;
coef(455,1) = -1;
%456 a2*a5*a7*eta(t-4)
aux1(456,2) = 1;
aux1(456,5) = 1;
aux1(456,7) = 1;
aux1(456,20+t-4) = 1;
coef(456,1) = -1;
%457 a3*a5*a7*eta(t-3)
aux1(457,3) = 1;
aux1(457,5) = 1;
aux1(457,7) = 1;
aux1(457,20+t-3) = 1;
coef(457,1) = -1;
%458 a1*a6*a8*eta(t-5)
aux1(458,1) = 1;
aux1(458,6) = 1;
aux1(458,8) = 1;
aux1(458,20+t-5) = 1;
coef(458,1) = -1;
%459 a2*a4*a8*eta(t-6)
aux1(459,2) = 1;
aux1(459,4) = 1;
aux1(459,8) = 1;
aux1(459,20+t-6) = 1;
coef(459,1) = -1;
%460 a2*a5*a8*eta(t-5)
aux1(460,2) = 1;
aux1(460,5) = 1;
aux1(460,8) = 1;
aux1(460,20+t-5) = 1;
coef(460,1) = -1;
%461 a2*a6*a7*eta(t-5)
aux1(461,2) = 1;
aux1(461,6) = 1;
aux1(461,7) = 1;
aux1(461,20+t-5) = 1;
coef(461,1) = -1;
%462 a3*a5*a8*eta(t-4)
aux1(462,3) = 1;
aux1(462,5) = 1;
aux1(462,8) = 1;
aux1(462,20+t-4) = 1;
coef(462,1) = -1;
%463 a3*a6*a7*eta(t-4)
aux1(463,3) = 1;
aux1(463,6) = 1;
aux1(463,7) = 1;
aux1(463,20+t-4) = 1;
coef(463,1) = -1;
%464 a4*a5*a7*eta(t-4)
aux1(464,4) = 1;
aux1(464,5) = 1;
aux1(464,7) = 1;
aux1(464,20+t-4) = 1;
```

Appendix

```
coef(464,1) = -1;
%465 a2*a6*a8*eta(t-6)
aux1(465,2) = 1;
aux1(465,6) = 1;
aux1(465,8) = 1;
aux1(465,20+t-6) = 1;
coef(465,1) = -1;
%466 a3*a6*a8*eta(t-5)
aux1(466,3) = 1;
aux1(466,6) = 1;
aux1(466,8) = 1;
aux1(466,20+t-5) = 1;
coef(466,1) = -1;
%467 a4*a5*a8*eta(t-5)
aux1(467,4) = 1;
aux1(467,5) = 1;
aux1(467,8) = 1;
aux1(467,20+t-5) = 1;
coef(467,1) = -1;
%468 a4*a6*a7*eta(t-5)
aux1(468,4) = 1;
aux1(468,6) = 1;
aux1(468,7) = 1;
aux1(468,20+t-5) = 1;
coef(468,1) = -1;
%469 a4*a6*a8*eta(t-6)
aux1(469,4) = 1;
aux1(469,6) = 1;
aux1(469,8) = 1;
aux1(469,20+t-6) = 1;
coef(469,1) = -1;
%470 a1*a3*a5*a7*eta(t-4)
aux1(470,1) = 1;
aux1(470,3) = 1;
aux1(470,5) = 1;
aux1(470,7) = 1;
aux1(470,20+t-4) = 1;
coef(470,1) = -1;
%471 a1*a3*a5*a8*eta(t-5)
aux1(471,1) = 1;
aux1(471,3) = 1;
aux1(471,5) = 1;
aux1(471,8) = 1;
aux1(471,20+t-5) = 1;
coef(471,1) = -1;
%472 a1*a3*a6*a7*eta(t-5)
aux1(472,1) = 1;
aux1(472,3) = 1;
aux1(472,6) = 1;
aux1(472,7) = 1;
aux1(472,20+t-5) = 1;
coef(472,1) = -1;
%473 a1*a4*a5*a7*eta(t-5)
aux1(473,1) = 1;
aux1(473,4) = 1;
aux1(473,5) = 1;
aux1(473,7) = 1;
```

Appendix

```
aux1(473,20+t-5) = 1;
coef(473,1) = -1;
%474 a2*a3*a5*a7*eta(t-5)
aux1(474,2) = 1;
aux1(474,3) = 1;
aux1(474,5) = 1;
aux1(474,7) = 1;
aux1(474,20+t-5) = 1;
coef(474,1) = -1;
%475 a1*a3*a6*a8*eta(t-6)
aux1(475,1) = 1;
aux1(475,3) = 1;
aux1(475,6) = 1;
aux1(475,8) = 1;
aux1(475,20+t-6) = 1;
coef(475,1) = -1;
%476 a1*a4*a5*a8*eta(t-6)
aux1(476,1) = 1;
aux1(476,4) = 1;
aux1(476,5) = 1;
aux1(476,8) = 1;
aux1(476,20+t-6) = 1;
coef(476,1) = -1;
%477 a1*a4*a6*a7*eta(t-6)
aux1(477,1) = 1;
aux1(477,4) = 1;
aux1(477,6) = 1;
aux1(477,7) = 1;
aux1(477,20+t-6) = 1;
coef(477,1) = -1;
%478 a2*a3*a5*a8*eta(t-6)
aux1(478,2) = 1;
aux1(478,3) = 1;
aux1(478,5) = 1;
aux1(478,8) = 1;
aux1(478,20+t-6) = 1;
coef(478,1) = -1;
%479 a2*a3*a6*a7*eta(t-6)
aux1(479,2) = 1;
aux1(479,3) = 1;
aux1(479,6) = 1;
aux1(479,7) = 1;
aux1(479,20+t-6) = 1;
coef(479,1) = -1;
%480 a2*a4*a5*a7*eta(t-6)
aux1(480,2) = 1;
aux1(480,4) = 1;
aux1(480,5) = 1;
aux1(480,7) = 1;
aux1(480,20+t-6) = 1;
coef(480,1) = -1;
%481 a1*a4*a6*a8*eta(t-7)
aux1(481,1) = 1;
aux1(481,4) = 1;
aux1(481,6) = 1;
aux1(481,8) = 1;
aux1(481,20+t-7) = 1;
```

Appendix

```
coef(481,1) = -1;
%482 a2*a3*a6*a8*eta(t-7)
aux1(482,2) = 1;
aux1(482,3) = 1;
aux1(482,6) = 1;
aux1(482,8) = 1;
aux1(482,20+t-7) = 1;
coef(482,1) = -1;
%483 a2*a4*a5*a8*eta(t-7)
aux1(483,2) = 1;
aux1(483,4) = 1;
aux1(483,5) = 1;
aux1(483,8) = 1;
aux1(483,20+t-7) = 1;
coef(483,1) = -1;
%484 a2*a4*a6*a7*eta(t-7)
aux1(484,2) = 1;
aux1(484,4) = 1;
aux1(484,6) = 1;
aux1(484,7) = 1;
aux1(484,20+t-7) = 1;
coef(484,1) = -1;
%485 a2*a4*a6*a8*eta(t-8)
aux1(485,2) = 1;
aux1(485,4) = 1;
aux1(485,6) = 1;
aux1(485,8) = 1;
aux1(485,20+t-8) = 1;
coef(485,1) = -1;
%486 eta(t)
aux1(486,20+t) = 1;
coef(486,1) = -1;

ineqPolySys{t-8}.supports = sparse(aux1);
ineqPolySys{t-8}.coef = coef;

end
lbd = [-1e10*ones(1,20) -delta_eta*ones(1,N)];
ubd = -lbd;

param.relaxOrder = 1;
param.POPsolver = 'active-set';
[param,SDPobjValue,POP] = sparsePOP(objPoly,ineqPolySys,lbd,
ubd,param);
PUI(k,j) = POP.xVectL(k)
end
end
```

Bibliography and webography

- [1] Huang, W., Huang, X., Majidi, C. et al. Dynamic simulation of articulated soft robots. Nat Commun 11, 2020.
- [2] Jacob Jamey D., Simpson A., Smith S., Design and Flight Testing of Inflatable Wings with Wing Warping, 2005.
- [3] <https://dspace.mit.edu/bitstream/handle/1721.1/100772/SoftRoboticsReview-FinalAuthorVersion.pdf;jsessionid=26843CDDA70AECA5B83B92B3000A3003?sequence=1>
- [4] <https://startupitalia.eu/56650-20160830-octopus-polpo-soft-robot>
- [5] Tolley M., Shepherd R., A Resilient, Untethered Soft Robot, 2014.
- [6] Zion T., Yue C., Soft Robotics in Medical Applications, Journal of Medical Research, Vol. 03,2018.
- [7] <https://www.nasa.gov/centers/marshall/news/news/releases/2014/14-112.html>
- [8] <https://www.nasa.gov/content/bigelow-expandable-activity-module>
- [9] https://www.esa.int/ESA_Multimedia/Images/2020/09/Inflatable_Moon_base
- [10] Brown G., Haggard R., Inflatable structures for deployable wings, 2012.
- [11] Freeland R.E., Bilyeu G., In-step inflatable antenna experiment, Acta Astronautica, pp.29-40,1993.
- [12] https://www.researchgate.net/publication/322809624_Inflatable_structure_for_aerospace_application_Historical_perspective_and_future_outlook
- [13] <https://www.digitaltrends.com/cool-tech/nasa-robot-king-louie/>
- [14] Palmieri, P.; Gaidano, M.; Ruggeri, A.; Salamina, L.; Troise, M.; Mauro, S. An Inflatable Robotic Assistant for Onboard Applications. In Proceedings of the

International Astronautical Congress, IAC, Dubai, United Arab Emirates, 25–29 October 2021; Volume A5, pp. 25–29.

[15] P. Palmieri, M. Gaidano, M. Troise, L. Salamina, A. Ruggeri and S. Mauro, A deployable and inflatable robotic arm concept for aerospace applications, 2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace), 2021, pp. 453-458.

[16] Troise, M.; Gaidano, M.; Palmieri, P.; Mauro, S. Preliminary Analysis of a Lightweight and Deployable Soft Robot for Space Applications.

[17] Mohamed Z., Review of modelling and control of flexible-link manipulators, Proceedings of the Institution of Mechanical Engineers Part I Journal of System and Control Engineering, 2016, pp.861-873.

[18] Siciliano, Robotics Modelling Planning and Control, 2009.

[19] Veldman, S.L., Design and Analysis Methodologies for Inflated Beams; Delft University Press: Delft, The Netherlands, 2005.

[20] <https://www.orchina.com/webhelp/OrcaFlex/Content/html/Rayleighdamping,Guidance.htm#:~:text=Classical%20Rayleigh%20damping%20is%20viscous,%CE%BB%20are%20constants%20of%20proportionality>

[21] Waki H., Kim S., User Manual for SparsePOP, Research Reports on Mathematical and Computing Sciences, 2009.

[22] R. M. Mahamood and J. O. Pedro, "Hybrid PD/PID controller design for two-link flexible manipulators," 2011 8th Asian Control Conference (ASCC), 2011, pp. 1358-1363.

[23] Lochan K., Roy B. K., Subudhi B., A review on two-flexible manipulators, Annual Reviews in control, 2016, pp.346-367.

[24] Palmieri, P.; Melchiorre, M.; Mauro, S. Design of a Lightweight and Deployable Soft Robotic Arm. Robotics 2022, 11, 88.

[25] Lee, C., Kim, M., Kim, Y.J. et al. Soft robot review. Int. J. Control Autom. Syst. 15, 3–15, 2017.

- [26] Zahra S. N., Debao Z., A Novel Soft Actuator for Continuum Soft Robot Arm,2018 Proceedings of the 2nd International Conference on Biomedical Engineering and Bioinformatics, pp. 138-143,2018.