

Modeling and Control of 6-axis Robot Arm

Ali Murtatha Shuman

Master of Science Thesis in Electrical Engineering

Modeling and Control of 6-axis Robot Arm

Ali Murtatha Shuman

LiTH-ISY-EX-20/5351-SE

Supervisor: **Hamed Haghshenas**
 ISY, Linköpings universitet
 Jonas Larsson
 ABB Corporate Research

Examiner: **Svante Gunnarsson**
 ISY, Linköpings universitet

*Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden*

Copyright © 2020 Ali Murtatha Shuman

Abstract

Robot manipulators are getting more and more attention nowadays. This is due to their high precision and the speed they provide while executing their tasks. The desires for such high standards are increasing exponentially due to the extended workspace that manipulators provide. Therefore, a safe controller is needed to make it possible for the robot to work alongside people considering the safety precautions. These safety preconditions are widely spread, even when the needs for better human-friendly robots are rising.

This thesis will introduce and explain a way to model a 6-axis robot by using its dynamical properties as well as the development of a joint space inverse dynamic controller. The controller will be tested in various different ways. Firstly by adding noise to the measured data. Then testing the robustness of the control model, while the simulated model includes properties different from those used for the controller itself. The different properties would for example be payloads and the inertia of the links. Thereafter, evaluating the precision of a followed path that is given by an operational space trajectory.

The outcome of these experiments show promising results. The results show that the controller is able to manage a noise in both the joint angle and joint velocity. It also shows that an error in the payload data will give a small error in the joint angles, sequentially that gives an acceptable error for the end-effector in the operational space. Furthermore, the controller manages to keep the maximum error in the joint angle low, while it is following a trajectory in the operational space.

Acknowledgments

I would like to start by showing my gratitude to the *ABB corporate research* group at ABB for welcoming me and giving me the opportunity to do the thesis. A special thanks to my supervisor at ABB Jonas larsson for all the feedback and the knowledge he shared with me. Furthermore I would like to thank my examiner and supervisor at Linköping University Svante Gunnarsson and Hamed Haghshenas for their support and advises through my thesis.

I would also like to thank Haitem Haider for crafting Figure 2.2. Lastly I would like to show a special and monumental appreciation towards my brother, Zain Al-abideen Shuman, for proofreading the thesis report.

Linköping, December 16, 2020
Ali Murtatha Shuman

Contents

List of Figures	ix
List of Tables	xvi
Notation	xvii
1 Introduction	1
1.1 Purpose	2
1.2 Problem Statement	2
1.3 Related Work	2
1.3.1 MO-MA Hybrid	2
1.3.2 Rising	3
1.3.3 RB-KAIROS	3
1.4 Resources	3
2 Theory: Dynamic Modeling of Manipulator Structures	5
2.1 Configurations of Rigid Body	5
2.2 Transformation	7
2.3 DH Parameters	9
2.4 Kinematics	10
2.5 Dynamics	11
2.5.1 Lagrange Formulation	12
2.5.2 Example: Two DoF Non-Planar Robot Arm	15
2.5.3 Example: Validation	17
3 Theory: Joint Space Inverse Dynamic Control	19
3.1 Simulated Model of the Manipulator	19
3.2 Inverse Dynamics Control	20
3.2.1 Nonlinear Compensation and Decoupling	21
3.2.2 PD-Controller	22
4 Application to the Robot	23
4.1 Direct Implementation	25

4.2	Robotics Toolbox RTB	26
4.3	Robotic System Toolbox RSTB	26
5	Simulation Results	29
5.1	Experiment Baseline Case	29
5.2	Measurement Disturbance Influence on the Controller	36
5.3	Robustness of Control Model	42
5.3.1	Robustness of Payload Influence Test	43
5.3.2	Robustness of Inertia Influence Test	54
5.4	Operational Space Time Path	64
6	Result	73
6.1	Discussion	73
6.2	Future Development	74
A	RTB Definition of The Links Chain	77
B	<i>Simulink</i> Model of the Robot Controller Using RSTB.	79
C	RSTB Definition of The Links Chain	81
D	Tests of the Noise Disturbance	83
E	Payload Tests	95
F	Inertia Test	107
	Bibliography	119

List of Figures

1.1	ABB prototype of 6-axis robot arm.	1
2.1	The red parts represent a revolute and prismatic joint in two different positions: A and B. The arrow shows the rotational and translational axis.	6
2.2	Robot arm, chain of links, with links reference frames.	6
2.3	A) Cartesian, B) Cylindrical and C) Spherical coordinate systems. .	7
2.4	Center of gravity of link in different reference frames.	9
2.5	Center of gravity of rigid-body in different reference frames. . . .	15
2.6	Test results of validation of the equation of motion for the two DoF non-planar robot arm.	18
3.1	Manipulator simulated model structure, A) using RSTB (Robotics system toolbox by <i>MATLAB</i>), B) using MFB (<i>MATLAB</i> function Block).	20
3.2	Joint space inverse dynamics control	21
3.3	Exact linearization of the system dynamics.	21
4.1	The robot and the reference frames for each body link.	24
4.2	The joint frames according to the DH convention.	25
4.3	<i>Simulink</i> model of the robot controller using RSTB, further described in Appendix B.	27
5.1	Ideal case. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1. . .	31
5.2	Ideal case. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	32
5.3	Ideal case. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	33

5.4	Ideal case. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	34
5.5	Ideal case. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	35
5.6	Noise generator structure.	36
5.7	Disturbance case: added noise on joint angle q and joint velocity \dot{q} . From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	37
5.8	Disturbance case: added noise on joint angle q and joint velocity \dot{q} . From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	38
5.9	Disturbance case: added noise on joint angle q and joint velocity \dot{q} . Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	39
5.10	Disturbance case: added noise on joint angle q and joint velocity \dot{q} . Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	40
5.11	Disturbance case: added noise on joint angle q and joint velocity \dot{q} . Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	41
5.12	Payload test: test weight is 10% higher. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	43
5.13	Payload test: test weight is 10% higher. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	44
5.14	Payload test: test weight is 10% higher. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	45
5.15	Payload test: test weight is 10% higher. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	46

5.16 Payload test: test weight is 10% higher. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	47
5.17 Payload test: test weight is 10% lower. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	48
5.18 Payload test: test weight is 10% lower. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	49
5.19 Payload test: test weight is 10% lower. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	50
5.20 Payload test: test weight is 10% lower. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	51
5.21 Payload test: test weight is 10% lower. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	52
5.22 Inertia test: inertia tensor 10% higher. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	54
5.23 Inertia test: inertia tensor 10% higher. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	55
5.24 Inertia test: inertia tensor 10% higher. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	56
5.25 Inertia test: inertia tensor 10% higher. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	57
5.26 Inertia test: inertia tensor 10% higher. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	58

5.27 Inertia test: inertia tensor 10% lower. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	59
5.28 Inertia test: inertia tensor 10% lower. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	60
5.29 Inertia test: inertia tensor 10% lower. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	61
5.30 Inertia test: inertia tensor 10% lower. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	62
5.31 Inertia test: inertia tensor 10% lower. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	63
5.32 Trajectory test: the time reference signal in the operational space.	64
5.33 Trajectory test: 3D representation of the given trajectory and the robot, in four different frames.	65
5.34 Operational space trajectory test. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	66
5.35 Operational space trajectory test. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	67
5.36 Operational space trajectory test. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	68
5.37 Operational space trajectory test. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	69
5.38 Operational space trajectory test. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	70
6.1 Robustness control, for the future development.	74
B.1 <i>Simulink</i> model of the Robot controller using RSTB.	80

D.1	Disturbance case: added noise on joint angle q . From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	84
D.2	Disturbance case: added noise on joint angle q . From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	85
D.3	Disturbance case: added noise on joint angle q . Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	86
D.4	Disturbance case: added noise on joint angle q . Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	87
D.5	Disturbance case: added noise on joint angle q . Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	88
D.6	Disturbance case: added noise on joint velocity \dot{q} . From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	89
D.7	Disturbance case: added noise on joint velocity \dot{q} . From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	90
D.8	Disturbance case: added noise on joint velocity \dot{q} . Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	91
D.9	Disturbance case: added noise on joint velocity \dot{q} . Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	92
D.10	Disturbance case: added noise on joint velocity \dot{q} . Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	93
E.1	Payload test: test weight 5% higher. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	96

E.2	Payload test: test weight 5% higher. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	97
E.3	Payload test: test weight 5% higher. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	98
E.4	Payload test: test weight 5% higher. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	99
E.5	Payload test: test weight 5% higher. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	100
E.6	Payload test: test weight 5% lower. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	101
E.7	Payload test: test weight 5% lower. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	102
E.8	Payload test: test weight 5% lower. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	103
E.9	Payload test: test weight 5% lower. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	104
E.10	Payload test: test weight 5% lower. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	105
F.1	Inertia test: inertia tensor 5% higher. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	108
F.2	Inertia test: inertia tensor 5% higher. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	109
F.3	Inertia test: inertia tensor 5% higher. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	110

F.4	Inertia test: inertia tensor 5% higher. Top graph: total saturated torques u_S . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	111
F.5	Inertia test: inertia tensor 5% higher. Top graph: total saturated control signal y_S . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	112
F.6	Inertia test: inertia tensor 5% lower. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	113
F.7	Inertia test: inertia tensor 5% lower. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	114
F.8	Inertia test: inertia tensor 5% lower. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	115
F.9	Inertia test: inertia tensor 5% lower. Top graph: total saturated torques u_S . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	116
F.10	Inertia test: inertia tensor 5% lower. Top graph: total saturated control signal y_S . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.	117

List of Tables

2.1	The parameters for two non-planer links-arm.	15
4.1	The DH parameters used for the direct implementation and RTB approaches.	25
5.1	Noise amplitude, depending on where the noise is added.	42

Notation

QUANTITIES

Notation	Meaning
n	Total number of Joints.
q_i	Joint (i) angle [rad].
\dot{q}_i	Joint (i) angular velocity [rad/s].
\ddot{q}_i	Joint (i) angular acceleration [rad/s ²].
q	Joints angle vector ($n \times 1$) [rad].
\dot{q}	Joints angular velocity vector ($n \times 1$) [rad/s].
\ddot{q}	Joints angular acceleration vector ($n \times 1$) [rad/s ²].
q_d	Desired joints angle vector ($n \times 1$) [rad].
\dot{q}_d	Desired joints angular velocity vector ($n \times 1$) [rad/s].
\ddot{q}_d	Desired joints angular acceleration vector ($n \times 1$) [rad/s ²].
\tilde{q}_d	Error in joints angle vector ($n \times 1$) [rad].
$\dot{\tilde{q}}_d$	Error in joints angular velocity vector ($n \times 1$) [rad/s].
$\ddot{\tilde{q}}_d$	Error in joints angular acceleration vector ($n \times 1$) [rad/s ²].
v	Linear velocity.
ω	Angular velocity.
τ	Torque.
ξ	Generalized force associated.
u	Torque signal to the simulated model.
u_s	Saturated torque signal to the simulated model.
\tilde{u}	Overshot of the saturated torque signal, $\tilde{u} = u - u_s$.
y	Acceleration signal to the nonlinear compensation and decoupling.
y_s	Saturated acceleration signal to the nonlinear compensation and decoupling.
\tilde{y}	Overshot of the saturated acceleration signal, $\tilde{y} = y - y_s$.

QUANTITIES

Notation	Meaning
\mathcal{L}	Lagrangian.
\mathcal{T}	Kinetic energy.
\mathcal{U}	Potential energy.
\mathcal{F}	Force.
H_e^w	Homogeneous transformation matrix (4×4) from end-effector frame (e) to the world frame (w), if nothing is mentioned in the exponent it is referred by default to the world frame.
H_i^{i-1}	Homogeneous transformation matrix (4×4) for joint (i) to frame ($i - 1$).
H_R	Homogeneous transformation matrix with only rotation quantities (4×4).
H_T	Homogeneous transformation matrix with only Transnational quantities (4×4).
R	Rotational matrix (3×3).
T	Transnational vector (3×1).
\widetilde{H}	Homogeneous transformation matrix (4×4), the error between the homogeneous transformation matrix of the desired angle q_d and the transformation matrix of actual joint angle q .
\widetilde{R}	Rotational matrix (4×4), the error between the rotational matrix of the desired angle q_d and the rotational matrix of actual joint angle q .
\widetilde{T}	Transnational vector (3×1), the error between the transnational vector of the desired angle q_d and the transnational vector of actual angle q .
$J(q)$	Geometric Jacobian matrix.
$J_P(q)$	Linear part of the Geometric Jacobian matrix ($3 \times n$).
$J_O(q)$	Angular part of the Geometric Jacobian matrix ($3 \times n$).
$B(q)$	Mass matrix ($n \times n$).
$C(q, \dot{q})$	Coriolis/Centrifugal matrix ($n \times n$).
$G(q)$	Gravity vector ($n \times 1$).
$n(q, \dot{q})$	Coriolis/Centrifugal and Gravity vector ($n \times 1$).

ABBREVIATIONS

Abbreviation	Meaning
DoF	Degree of Freedom
3D	Three-dimensional
CoG	Center of gravity
RTB	Robotics toolbox by Preter Corke
RSTB	Robotics system toolbox by <i>MATLAB</i>
MFB	<i>MATLAB</i> function Block
B	Base
J1	Joint 1
LA	Lower Arm
J2	Joint 2
UA	Upper Arm
J3	Joint 3
OS	Output shift

1

Introduction

The word “robot” itself, was popularized for the first time a hundred years ago in the 1920s, in *Karel Čapek’s* play *R.U.R. Rossum’s Universal Robots*. With that said, it took approximately 30 years until the invention of a "real" functional robot to come into an existence, it went by the name "Unimate #001". It was invented by *Joseph Engelberger* and *George Devol*. The task of this robot was to assist a hot die-casting machine [15]. Now, after a century from popularizing the word “robot” and development in robotics, this master thesis project, is one of the continued projects that aim to expand the development and understanding of robotics.

The goal of this thesis is to derive a joint space inverse dynamic controller for a six degree of freedom (DoF) robot arm and simulate the performance using *MATLAB*. The prototype of the robot can be seen in Figure 1.1. This project is the building block for future improvements, where there will be a possibility to integrate the dynamics as well as the controller for the robot using a mobile platform. This makes it possible for the manipulator and mobile platform to enact as a singular unit.



Figure 1.1: ABB prototype of 6-axis robot arm.

This chapter contains the purpose and the problem statement of this work as well as the related work. An introduction about how to derive the dynamic model that describes the dynamics of this robot or any manipulator can be found in chapter 2. Thereafter, a joint space inverse dynamic method will be introduced in chapter 3, which will also be implemented in *Simulink*. Chapter 4 includes the methods that are available and used to apply the dynamics and control algorithms of the robot. Lastly, the simulation experiment can be found in chapter 5, followed by the conclusion and results of this work in chapter 6.

1.1 Purpose

This master thesis was proposed at *ABB* in purpose of developing a dynamical model and the controller of a 6DoF robot arm. The purpose of this work is to study and develop a full body control of a manipulator robot, by using the dynamical properties of the manipulator. This is done in hopes of developing manipulators that are collaborative with humans at workplaces such as hospitals where both humans and robots coexist.

1.2 Problem Statement

The problems handled in this thesis are:

- Creating a kinematic and dynamic model that describes a 6DoF robot arm.
- Making a joint space inverse dynamics controller for a 6DoF robot arm.
- Testing and evaluating the model and control approach.

1.3 Related Work

The mobile manipulators have gotten the attention of many companies, due to their varied work possibilities and flexibility. The text below will present some mobile manipulators that are comparable to putting the robot on a mobile base. A similar project that derives the dynamic as well as the controller, for a manipulator arm with 6DoF, has been studied earlier in [1] and [5].

1.3.1 MO-MA Hybrid

The *MO-MA hybrid* is a two-part robot that has *Omron Cart Transporter* as a mobile platform and *TM Collaborative Robot* as the manipulator arm.

The Cart Transporter is mainly used for transporting payloads up to 130 kg and with 0.9m/s max speed. The mobile platform is intended to be used in warehouses and busy factories, as well as collaborating with other mobile platforms *Omron Cart Transporter* [11].

1.3.2 Rising

Rising is a relatively small robot with up to 7 DoF robotic arm. This platform is designed for interventions, reconnaissance, and EOD (Explosive ordnance disposal) missions. The mobile platform uses 2 tracks and 2 flippers to manoeuvre. This makes it possible for the robot to climb with a maximum of 80% climbing angle [13].

1.3.3 RB-KAIROS

RB-KAIROS is also a two part mobile manipulator, where the mobile platform has a Swedish wheel [3], and the manipulator has 6 DoF. This mobile manipulator has the most similarities to the robor, in both size and functionality [16]. The main distinction is the Swedish wheel.

1.4 Resources

To achieve the goal of this thesis work, the following resources were provided by *Linköping University* and *ABB*:

- Place to work (*ABB*)
- PC (*ABB*)
- The robor characteristic parameters (*ABB*)
- *MATLAB* license (LiU)
- Supervisor (*ABB* / LiU)

2

Theory: Dynamic Modeling of Manipulator Structures

This chapter will include the theory that was studied and used for the project. It will in other words consist of a brief introduction of how to describe a robot as well as homogeneous transformation to transform the representation of the reference frames from one point to another. Lastly how to derive the DH parameters (Denavit–Hartenberg), a convention to help finding the expressions for the transformation matrices. Thereafter, a proper introduction will be presented. This introduction will consist of the kinematics and dynamics concepts, as well as the differential kinematics and the inverse-dynamics [6]. There will lastly be an example of how to derive the dynamical equation of 2 DoF non-planar arm.

2.1 Configurations of Rigid Body

Describing a robot in 3-dimensions will require the configuration for that said robot. The configuration includes the specifics of all the position points. In robotics there are two types of bodies, a flexible body, which can be deformed like *RBO Hand 2* [2], and a rigid body, which will be the focus of this project, since the robot can be perceived as entirely rigid. The rigid bodies, also called links, are connected to each other by a revolute or prismatic joint, see Figure 2.1.

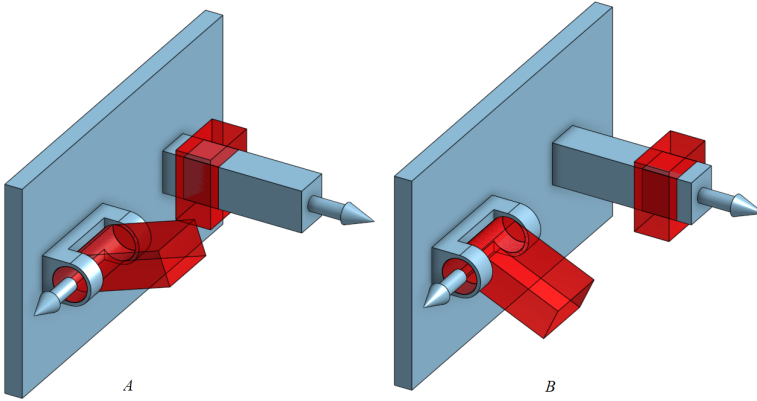


Figure 2.1: The red parts represent a revolute and prismatic joint in two different positions: A and B. The arrow shows the rotational and translational axis.

Robot manipulators, in general, are built by two or several links. Each link has its own mass center, moment of inertia, etc. It is easier to define a link in a fixed coordinate system for each link part, in the Figure 2.2 a fixed coordinate system (world frame) shown with the black axis. This coordinate system does not move its position nor the orientation with the change of robot joint variable. However, the links' fixed coordinate system (reference frames) which is shown with (red, green and blue) representing (x , y and z)-axis can change the position and orientation. The change occurs alongside the change of the joint variables, a definition of the joint variables will be introduced in section 2.3. Thereafter, to describe the position of the mass center of any link part, a transformation can be used between the frames.

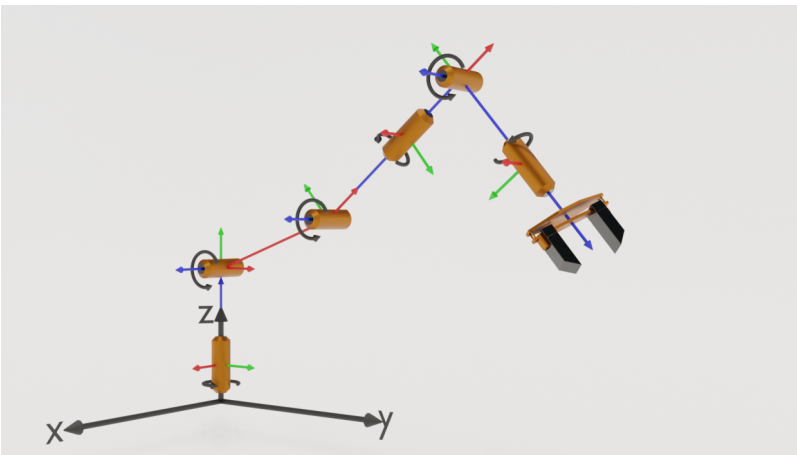


Figure 2.2: Robot arm, chain of links, with links reference frames.

There are many ways to describe the position of a point in the 3D space, such as using the Cartesian, cylindrical and/or spherical coordinate systems, see Figure 2.3. The Cartesian coordinate systems will be used all throughout this thesis.

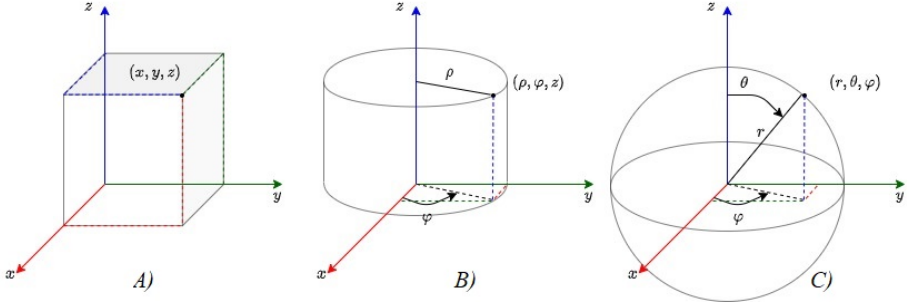


Figure 2.3: A) Cartesian, B) Cylindrical and C) Spherical coordinate systems.

2.2 Transformation

Using matrices to apply a translation and/or rotation to a point is beneficial due to the associative property of the matrix multiplication. The homogeneous transformation matrix, is a (4×4) matrix, which gives the information that changes the representation vector p^l of a certain point from one frame to another, which gives vector p^w . A general homogeneous transformation matrix looks as follows:

$$H_l^w = \begin{bmatrix} R & T \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x & p_x \\ a_y & b_y & c_y & p_y \\ a_z & b_z & c_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$p^w = H_l^w p^l$$

The first (3×3) elements are the rotation matrix R and the first 3 elements in the last column contains the translation T . The rotations around the axis (x, y, z) with angles (ϕ, γ, θ) are given by the following homogeneous transformations:

$$\begin{aligned}
H_{R_x}(\phi) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
H_{R_y}(\gamma) &= \begin{bmatrix} \cos \gamma & 0 & \sin \gamma & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \gamma & 0 & \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
H_{R_z}(\theta) &= \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{2.2}$$

This makes it easy to calculate the end-effectors position/rotation, by multiplying the transformation matrices of the joints frames with each other. With that, the final transformation matrix can be obtained from the link frame coordinate to the world coordinate system or vice versa by inverting it.

$$\begin{aligned}
H_e^0 &= H_1^0 H_2^1 \dots H_{i-1}^{i-2} H_e^{i-1} \\
H_0^e &= (H_e^0)^{-1}
\end{aligned} \tag{2.3}$$

Steiner Theorem

The inertia tensor of a link is often defined w.r.t. the mass center. After using the Steiner theorem (2.4), also called the parallel axis theorem, the inertia tensor can be redefined w.r.t. the reference frame [14].

$$I_f = I_c + m S(r_c^f)^T S(r_c^f) \tag{2.4}$$

The I_c is the inertia tensor w.r.t. the center of gravity, m is the mass of the link, and the $S(r_c^f)$ is the skew-symmetric matrix of the distance r_c^f from the reference frame to the center of gravity [14]. The skew-symmetric matrix is given by equation (2.5),

$$S(r) = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{2.5}$$

where $r = [r_x, r_y, r_z]^T$. With that said, to translate the inertia tensor of the link from one reference frame i to another reference frame $i-1$ as shown in Figure 2.4 without knowing I_c , the inertia tensor in the center of mass, the following calculation is used. By recasting the equation in (2.4) with the knowledge of the inertia tensor in the reference ($i=1$) to be I_1 , and with that deriving the inertia tensor I_0 in the reference ($i-1=0$).

$$\begin{aligned} I_1 &= I_c + mS(r_c^1)^T S(r_c^1) \Leftrightarrow \\ I_c &= I_1 - mS(r_c^1)^T S(r_c^1) \end{aligned} \quad (2.6)$$

$$\begin{aligned} I_0 &= I_1 - mS(r_c^1)^T S(r_c^1) + mS(r_c^0)^T S(r_c^0) \Leftrightarrow / r_c^0 = r_1^0 + r_c^1 / \\ I_0 &= I_1 + m\left(S(r_1^0 + r_c^1)^T S(r_1^0 + r_c^1) - S(r_c^1)^T S(r_c^1)\right) \end{aligned} \quad (2.7)$$

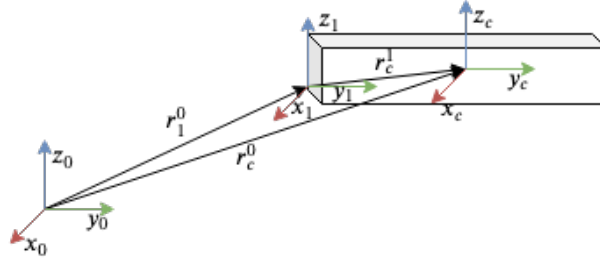


Figure 2.4: Center of gravity of link in different reference frames.

The inertia tensor also depends on the orientation of the reference frame, and if the origin of the reference frame rotates with rotation matrix R . The inertia tensor I'_o in the new frame is related to I_o by the following relationship:

$$I_o = RI'_oR^T \quad (2.8)$$

2.3 DH Parameters

DH parameters are four parameters that describe the convention for attaching a reference frame to the links of a robot manipulator [14], or in general, a kinematic chain, see Figure 2.2. This convention is based on four transformations, first rotation around \hat{z} followed by a translation in \hat{z} and then a rotation around \hat{x} and lastly a translation in \hat{x} . The resulting Homogeneous transformation of these operations are given by equation (2.9).

$$\begin{aligned} H_i &= H_{R_z}(\theta_i)H_{T_z}(d_i)H_{R_x}(\alpha_i)H_{T_x}(a_i) \\ &= \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.9)$$

The four quantities are θ_i , d_i , α_i and a_i , and indexing with the frame number i . These quantities represent the following:

- θ_i : Link twist angle from x_{i-1} to x_i measured along z_{i-1} .
- d_i : The distance from x_{i-1} to x_i measured along z_{i-1} .
- α_i : Link twist angle from z_{i-1} to z_i measured along x_i .
- a_i : The distance from z_{i-1} to z_i measured along x_i .

One of those quantities θ_i or d_i can either be a variable or a constant. If the joint i is prismatic then d_i is a variable and θ_i is the constant. If the joint is revolute then the variable is θ_i and the d_i is constant. The variable parameter θ will be referred to as a joint angle or q . In the occasion that both would turn out to be constants then the link i is a fixed link.

This convention applies a set of rules for how the frames coordinates are chosen and oriented.

1. Use right-hand frames for each joint.
2. Set z_i -axis along the revolution or translation axis for the Joint $i + 1$.
3. Select the origin O_i at the intersection point of axes z_i and the common normal with axis z_{i-1} .
4. Select the x_i -axes in the same direction of common normal from z_{i-1} and z_i .

2.4 Kinematics

The forward kinematics goal is to find the relationship between a robot/ manipulator's joint angles (q), and its end-effector's position [14]. The forward kinematics are used to compute the end-effector's position relatively to the base of the manipulator using the kinematics equations, where the base is referred to the link of the manipulator that is fixed in the room if the manipulator is mounted on a table or working station, otherwise where the mobile platform center reference-frame is, that is if the manipulator is mounted on a mobile platform . In other words, it provides the end-effector's position (p_e) in the room, when all the joint angles (q) are known.

These kinematic equations can easily be obtained by using the DH parameters to compute the transformation matrices (2.9) for each joint frame in the kinematic chain. Thereafter using (2.3) for computing the end-effector to base transformation H_e^0 , which gives the position of the end-effector, w.r.t the base frame, in the first three elements of the last column, as noted in (2.1).

$$p_e = \begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = H_e^0 p_0 = \begin{bmatrix} R_e(q) & T_e \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.10)$$

The p_e is a homogeneous vector of the end-effectors position.

Differential Kinematics

The differential kinematics goal is to get the relationship between the joint velocities and the end-effector angular as well as the linear velocities [6], [14], which is expressed in (2.11).

$$\begin{aligned}\dot{p}_e &= J_{P_e}(q)\dot{q} \\ \omega_e &= J_{O_e}(q)\dot{q}\end{aligned}\quad (2.11)$$

The J_p and J_o are $(3 \times n)$ matrices, where n is the number of joints, J_p is the contribution of joint velocities \dot{q} to end-effector linear velocities \dot{p}_e and the J_o for the angular velocities ω_e . This can be rewritten as

$$v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = \begin{bmatrix} J_{P_e}(q) \\ J_{O_e}(q) \end{bmatrix} \dot{q} = J_e(q)\dot{q} \quad (2.12)$$

where $J(q)$ expresses the system differential kinematics equation, with $(6 \times n)$ matrix that goes by the name "geometric Jacobian". The Jacobian matrix elements are computed as follows:

$$\begin{bmatrix} J_{P_i} \\ J_{O_i} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ \mathbf{0} \end{bmatrix} & \text{for a prismatic joint} \\ \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{for a revolute joint} \end{cases} \quad (2.13)$$

where p_e is given in (2.10) and z_{i-1} is derived from the rotation matrix R_{i-1}^0 , the third column of the rotation matrix.

$$z_{i-1} = R_1^0 \dots R_{i-1}^{i-2} z_0 \quad \text{where } z_0 \text{ is } \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (2.14)$$

Lastly the p_{i-1} is given by the following equation (2.15).

$$p_{i-1} = H_1^0 \dots H_{i-1}^{i-2} p_0 \quad \text{where } p_0 \text{ is } \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (2.15)$$

2.5 Dynamics

Generally, the mechanism of a robot is modelled as a rigid-body system. Therefore, robot dynamics can be used as an application of the rigid-body dynamics. This is followed by two main problems, the forward- and inverse-dynamics. The forward-dynamics are used to compute the accelerations \ddot{q} of the joint from the given joint angle q , joint velocity \dot{q} and the applied torque on the actuators τ . From the other side, the inverse-dynamics are given the joint angle q , joint velocity \dot{q} and joint accelerations \ddot{q} and works out the needed torque on the actuators τ [4].

The benefits of deriving a dynamical model of a manipulator are as follows: Having the ability of simulating a motion, analysis of the manipulator structure and the designing of the control algorithms.

Simulating the manipulator motion, allows testing for the control strategies as well as the motion planning, without the use of the physical system. This is beneficial in case the real system is not available, or if the test subjects, be it either the system or the user/developer, are exposed to some sort of risks.

2.5.1 Lagrange Formulation

One way to derive the dynamical model of the system is to use the Lagrange Formulation, a variation approach based on the kinetic and potential energy of the link system (robot)[14], [6]. The Lagrangian is given by:

$$\mathcal{L}(q, \dot{q}) = \mathcal{T}(q, \dot{q}) - \mathcal{U}(q) \quad (2.16)$$

where \mathcal{T} is the total kinetic energy and \mathcal{U} is the total potential energy of the system. The kinetic energy can be computed by:

$$\mathcal{T}_i = \sum_{i=1}^n \frac{1}{2} \dot{p}_i^T m_i \dot{p}_i + \frac{1}{2} \omega_i^T R_i I_{\ell_i}^i R_i^T \omega_i \quad (2.17)$$

where \dot{p} and ω are the linear and angular velocities as mentioned in 2.4. $I_{\ell_i}^i$ is the inertia tensor relative to the center of mass of link i when expressed in the base frame, and the R_i is the rotation matrix from link i frame to the base frame. This is mentioned in 2.2 equation (2.8), and can be rewritten as:

$$\mathcal{T}(q, \dot{q}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n b_{ij}(q) \dot{q}_i \dot{q}_j = \frac{1}{2} \dot{q}^T B(q) \dot{q} \quad (2.18)$$

where

$$\begin{aligned} B(q) &= \sum_{i=1}^n (m_i J_p^{(l_i)T} J_p^{(l_i)} + J_o^{(l_i)T} R_i I_{\ell_i}^i R_i^T J_o^{(l_i)} + m_{m_i} J_p^{(m_i)T} J_p^{(m_i)} + J_o^{(m_i)T} R_{m_i} I_{m_i}^i R_{m_i}^T J_o^{(m_i)}) \\ &= \sum_{i=1}^n (m_i J_p^{(i)T} J_p^{(i)} + J_o^{(i)T} R_i I_{\ell_i}^i R_i^T J_o^{(i)}) \end{aligned} \quad (2.19)$$

where $B(q)$ is the mass-matrix with $(n \times n)$ elements, which is *symmetric* and *positive definite* and in general configuration-dependent [14]. The J_p and J_o are the linear and angular parts of the geometric Jacobian matrix section 2.4, and the indexing (l) refers to the link mass and inertia and (m) refers to the motor mass and inertia. In this thesis, a combined version will be used where a motor and

a link is seen as a one-body-part. This is evident according to the data given by *ABB*. The potential energy can be computed by:

$$\mathcal{U}(q) = \sum_{i=1}^n m_i g_0^T p_{mi} \quad (2.20)$$

where g_0 is the gravity acceleration homogeneous vector in the base frame, e.g., $g_0 = [0 \ 0 \ g \ 0]^T$ if z is the vertical axis, and p_{mi} is the center of mass position of link (i) in the world frame.

Thereafter, the Lagrange equation is expressed by (2.21), or in vector form by (2.22).

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \quad i = 1, 2, \dots, n \quad (2.21)$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}} \right)^T - \left(\frac{\partial \mathcal{L}}{\partial q} \right)^T = \xi \quad (2.22)$$

where ξ_i is the generalized force associated with the generalized coordinate q_i . That is given by the contributions of the actuation torque τ_i at the joint i and of the viscous friction torques. This can be computed by equation (2.23).

$$\xi_i = \sum_{i=1}^{N_f} \left(\vec{F}_i \cdot \frac{\partial \vec{v}_i}{\partial \dot{q}_i} \right) + \sum_{i=1}^{N_\tau} \left(\vec{\tau}_i \cdot \frac{\partial \vec{\omega}_i}{\partial \dot{q}_i} \right) \quad (2.23)$$

where N_f and N_τ is the number of active non-conservative forces respectively torques. Lagrangian's derivatives in equation (2.22) gives the following:

$$B(q)\ddot{q} + n(q, \dot{q}) = \xi \quad (2.24)$$

ξ and $B(q)$ as mentioned above, and $n(q, \dot{q})$ is:

$$\begin{aligned} n(q, \dot{q}) &= \dot{B}(q)\dot{q} - \frac{1}{2} \left(\frac{\partial}{\partial \dot{q}} \left(\dot{q}^T B(q) \dot{q} \right) \right)^T + \left(\frac{\partial \mathcal{U}(q)}{\partial q} \right)^T \\ &= C(q, \dot{q})\dot{q} + G(q) \end{aligned} \quad (2.25)$$

The $C(q, \dot{q})$ is known as the coriolis-matrix with $(n \times n)$ elements. This can also be computed with equations (2.26) and (2.27), using the mass-matrix (2.19). c_{ij} are the elements of the coriolis-matrix, and b_{xx} are the mass-matrix elements.

$$c_{ij} = \sum_{k=1}^n c_{ijk} \dot{q}_k \quad (2.26)$$

$$c_{ijk} = \frac{1}{2} \left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right) \quad (2.27)$$

where $G(q)$ is known as the gravity term vector, under the assumption that the potential energy comes only from the gravity. But in case of the existence of springs at the joints, the springs terms will be added to $G(q)$, due to the contribution that the springs has to the potential energy. This can be obtained by equation (2.28).

$$G(q) = \frac{\partial \mathcal{U}(q)}{\partial q} \quad (2.28)$$

The final equation of motion, which describe the relationship between the desired torque and acceleration is given by the equation (2.29). This is obtained by combining both equation (2.23) and (2.24).

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_v\dot{q} + F_s\text{sgn}(\dot{q}) + G(q) = \tau - J^T(q)h_e \quad (2.29)$$

where

- τ is the actuation torques.
- $F_v\dot{q}$ is the viscous friction torques. F_v denotes the $(n \times n)$ diagonal matrix of viscous friction coefficients.
- $F_s\text{sgn}(\dot{q})$ is the Coulomb friction torques, where F_s is an $(n \times n)$ diagonal matrix and $\text{sgn}(\dot{q})$ denotes the $(n \times 1)$ vector whose components are given by the sign functions of the single joint velocities.
- h_e denotes the vector of force and moment exerted by the end-effector on the environment.

The friction torques occur from the motor shaft spinning in its bearings and the brushes sliding on the commutator, that may also depend on external loads. At no load the friction torques are given by:

$$\tau_{fric} = F_v\dot{q} + F_s\text{sgn}(\dot{q}) = K_t I_0 \quad (2.30)$$

to obtain F_v and F_s an estimation of each of F_v and F_s can be made by running the motor at two different voltages with no load.

This thesis consider all forces and torques that were mentioned above to be equal to zero, except for the actuation torques which is the desired input to the manipulator. This assumption is based on the lack of information for the quantities of these forces. They were therefore neglected for simplification proposes. That led to the following equation of motion:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (2.31)$$

Equation (2.31) is also called the inverse-dynamic equation, and the forward-dynamical equation is obtained by solving the joint acceleration (\ddot{q}) from the (2.31), which gives:

$$B(q)^{-1}(\tau - C(q, \dot{q})\dot{q} - G(q)) = \ddot{q} \quad (2.32)$$

that gives a linear function with respect to the actuators torques τ . The manipulation of this function is possible, due to the mass matrix $B(q)$ which is a full-rank matrix, that can be further inverted for any manipulator configuration. The benefit of the Lagrange-Formulation approach is to derive the dynamical equation in a systematic way, independently of the reference coordinate frame.

Another way to derive the dynamical model of the system is to use the Newton-Euler Formulation, which relies on $F = ma$ applied to each individual link of the robot. In other words, it describes the motion of the link based on a balance of all forces and moments acting on it. This leads to a set of equations whose structure is a recursive type of solution.

2.5.2 Example: Two DoF Non-Planar Robot Arm

As an example of the mentioned method, a computation of 2 DoF non-planar arm dynamical equation can be derived as shown below. The properties of this robot arm is shown in figure Figure 2.5 as well as table Table 2.1. This example is inspired by a tutorial from *MathWorks*® [10].

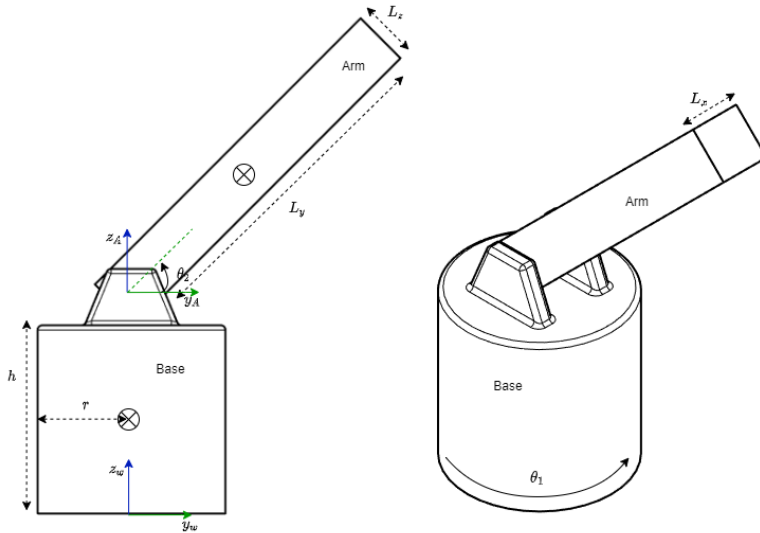


Figure 2.5: Center of gravity of rigid-body in different reference frames.

Table 2.1: The parameters for two non-planer links-arm.

Link	Mass	Dimensions	Angle
Base	m_B	h, r	θ_1
Arm	m_A	L_x, L_y, L_z	θ_2

The inertia tensors of the base and the arm can be computed as a simple cylinder respectively cube inertia tensor for simplicity by:

$$I_A = \begin{bmatrix} \frac{m_A (L_y^2 + L_z^2)}{12} & 0 & 0 \\ 0 & \frac{m_A (L_x^2 + L_z^2)}{12} & 0 \\ 0 & 0 & \frac{m_A (L_x^2 + L_y^2)}{12} \end{bmatrix} \quad (2.33)$$

$$I_B = \begin{bmatrix} \frac{m_B (h^2 + 3r^2)}{12} & 0 & 0 \\ 0 & \frac{m_B (h^2 + 3r^2)}{12} & 0 \\ 0 & 0 & \frac{r^2 m_B}{2} \end{bmatrix}$$

The positions of the mass-center in the world frame are given by:

$$p_B = \begin{bmatrix} 0 & 0 & h/2 \end{bmatrix}^T \quad (2.34)$$

$$p_A = \begin{bmatrix} -\frac{L_y \cos(\theta_2) \sin(\theta_1)}{2} & \frac{L_y \cos(\theta_1) \cos(\theta_2)}{2} & h + \frac{L_y \sin(\theta_2)}{2} \end{bmatrix}^T$$

Thereafter, the translation as well as the rotational velocity's can be obtained by:

$$\begin{bmatrix} \dot{p}_B \\ \omega_B \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}, \quad \begin{bmatrix} \dot{p}_A \\ \omega_A \end{bmatrix} = \begin{bmatrix} \frac{L_y \dot{\theta}_2 \sin(\theta_1) \sin(\theta_2)}{2} - \frac{L_y \dot{\theta}_1 \cos(\theta_1) \cos(\theta_2)}{2} \\ -\frac{L_y \dot{\theta}_1 \cos(\theta_2) \sin(\theta_1)}{2} - \frac{L_y \dot{\theta}_2 \cos(\theta_1) \sin(\theta_2)}{2} \\ \frac{L_y \dot{\theta}_2 \cos(\theta_2)}{2} \\ \dot{\theta}_2 \\ \dot{\theta}_1 \sin(\theta_2) \\ \dot{\theta}_1 \cos(\theta_2) \end{bmatrix} \quad (2.35)$$

With the expressions of the positions and the velocities of each link, the kinetic (T) as well as the potential (\mathcal{U}) energies can be computed as shown in 2.5.1, equation (2.18) and (2.20).

When the energy expressions are acquired, the Lagrangian as well as its generalized forces can be obtained by applying the derivations according to the Lagrangian equation (2.21) and (2.23).

$$\begin{aligned} \mathcal{L} = & \frac{m_A L_x^2 \dot{\theta}_1^2}{24} - \frac{m_A L_y^2 \dot{\theta}_1^2 \sin^2(\theta_2)}{6} + \frac{m_A L_y^2 \dot{\theta}_1^2}{6} + \frac{m_A L_y^2 \dot{\theta}_2^2}{6} - \frac{g m_A L_y \sin(\theta_2)}{2} \\ & + \frac{m_A L_z^2 \dot{\theta}_1^2 \sin^2(\theta_2)}{24} + \frac{m_A L_z^2 \dot{\theta}_2^2}{24} + \frac{m_B \dot{\theta}_1^2 r^2}{4} - g h m_A - \frac{g h m_B}{2} \end{aligned} \quad (2.36)$$

In this example the generalized forces will be considered equal to the actuation torque $\xi = \tau$. This is done by considering all the other forces to zero. This gives

the following equation of motion:

$$\begin{aligned}
 \tau_1 &= \ddot{\theta}_1 \frac{(m_A L_x^2 - 4 m_A L_y^2 \sin(\theta_2)^2 + 4 m_A L_y^2 + m_A L_z^2 \sin(\theta_2)^2 + 6 m_B r^2)}{12} \\
 &\quad - \frac{\dot{\theta}_1 \dot{\theta}_2 m_A \sin(2\theta_2) (4 L_y^2 - L_z^2)}{12} \\
 \tau_2 &= \ddot{\theta}_2 \frac{m_A (4 L_y^2 + L_z^2)}{12} + \\
 &\quad \frac{m_A (4 \sin(2\theta_2) L_y^2 \dot{\theta}_1^2 + 12g \cos(\theta_2) L_y - \sin(2\theta_2) L_z^2 \dot{\theta}_1^2)}{24}
 \end{aligned} \tag{2.37}$$

which can also be expressed in matrix form just as in equation (2.24).

$$\begin{aligned}
 B(q) &= \begin{bmatrix} \frac{m_A L_x^2 - 4 m_A L_y^2 \sin(\theta_2)^2 + 4 m_A L_y^2 + m_A L_z^2 \sin(\theta_2)^2 + 6 m_B r^2}{12} & 0 \\ 0 & \frac{m_A (4 L_y^2 + L_z^2)}{12} \end{bmatrix} \\
 n(q, \dot{q}) &= \begin{bmatrix} -\frac{\dot{\theta}_1 \dot{\theta}_2 m_A \sin(2\theta_2) (4 L_y^2 - L_z^2)}{12} \\ \frac{m_A (4 \sin(2\theta_2) L_y^2 \dot{\theta}_1^2 + 12g \cos(\theta_2) L_y - \sin(2\theta_2) L_z^2 \dot{\theta}_1^2)}{24} \end{bmatrix}
 \end{aligned} \tag{2.38}$$

Note that the expressions of the mass-matrix $B(q)$ as well as the coriolis/centrifugal and gravity vector $n(q, \dot{q})$ can be much longer, with consideration to the simplification that was used in this example. The simplifications that are being referred to are:

- The simplification of the inertia tensors, it has been simplified to be diagonal with the moments of inertia (I_{xz} , I_{yy} and I_{zz}), without the off-diagonal elements (I_{xy} , I_{xz} and I_{yz}), also known as the products of inertia", which is often included.
- The simplification in the generalized forces $\xi = \tau$.

Aside from that the example is only considering a two DoF, which will be gradually longer with each link (DoF), added to the system.

2.5.3 Example: Validation

To validate the equation in (2.37) and/or (2.38) three tests were made. First a test, testing what happens when no torque are applied to the system. Thereafter, a test to see what happens when only applying the torque required to keep the system in the steady state. Lastly, testing what happens when applying bigger torque than the torque that keeps the system in the steady state.

The first test gave that the second link behaved as a pendulum, which gave an oscillation between zero and $-\pi$ on the angle of the arm link, see the top graph

in Figure 2.6. This behavior is correct due to the gravity force and the lack of any other forces/torques that stops the oscillation. The second test showed that the required torques that needed to keep the system in the steady state on the start position are $\tau_1 = 0$ and $\tau_2 = \frac{gm_AL_y}{2}$. The result of this test can be seen in the middle graph on Figure 2.6, note that y -axis is multiplied by 10^{-13} , which means that this increase on θ_2 is due to numerical error in *MATLAB*. This test also proved that the equation of motion is correct, since the τ_2 is equal to the gravitation force multiplied by the distance between the mass center and the rotational axes. Finally the last test when applying torques bigger then 0 on τ_1 and $\frac{gm_AL_y}{2}$ on τ_2 the system started to rotate nonstop. The result of this test is shown on the bottom graph in Figure 2.6, both joint angles are increasing rapidly. With that, the elements of the equation of motion for the two DoF non-planar robot arm are given by (2.38).

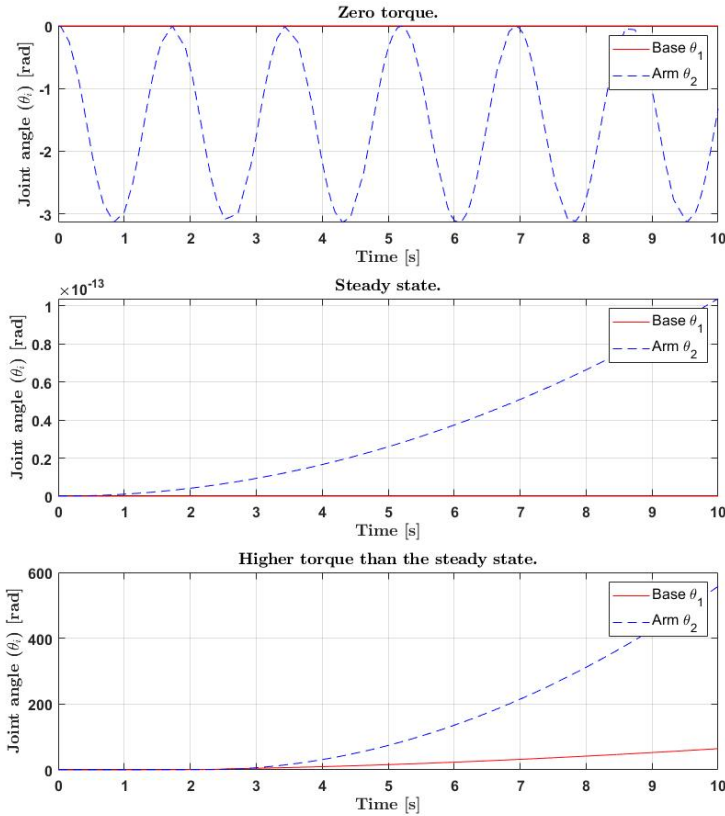


Figure 2.6: Test results of validation of the equation of motion for the two DoF non-planar robot arm.

3

Theory: Joint Space Inverse Dynamic Control

This chapter will cover how to create a model of the manipulator and the control method studied and used in this thesis [14].

3.1 Simulated Model of the Manipulator

To test and verify the controller approach and its results, it is necessary to start by creating a simulated representation of the robot/manipulator. Specially if there is not a physical robot available. It is also safer to use a simulated model of a robot/manipulator in the beginning of the tests of the controller, to minimize the risks of damaging the robot/manipulator or the developer.

The control input-signal (u) which is the torque to the actuators, is used to move the robot joints. By knowing the input signal representation, a simulated model can be created by deriving the dynamical model of the robot from the forward-dynamical equation (3.1). That will work out the joint accelerations (\ddot{q}).

$$\begin{aligned} B(q)^{-1}(\tau - C(q, \dot{q})\dot{q} - G(q)) &= \ddot{q} \\ B(q)^{-1}(\tau - n(q, \dot{q})) &= \ddot{q} \end{aligned} \tag{3.1}$$

By knowing the joint accelerations (\ddot{q}) of each link of the manipulator, a computation of the joint velocities and joint angles (\dot{q} , q) can easily be done by integrating the joint accelerations (\ddot{q}).

Feeding the computed joint velocity and joint angle back to the equation, that gives the following model structure, see Figure 3.1, which is a simulated model of the manipulator implemented in *MATLAB*.

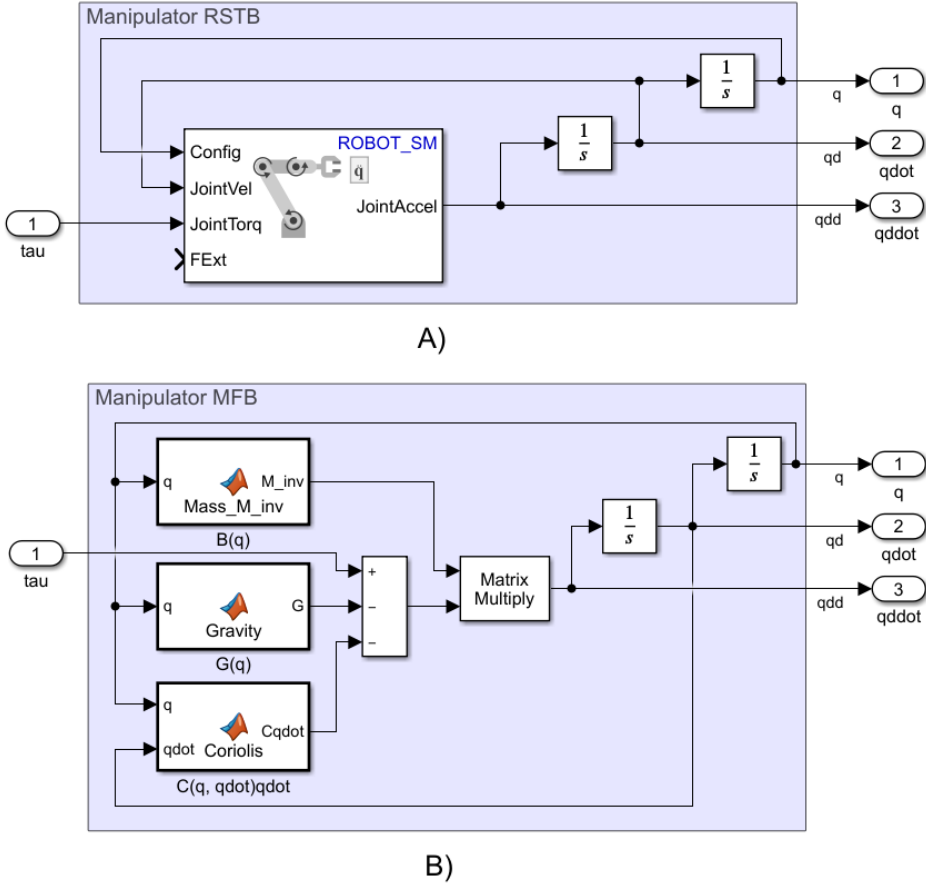


Figure 3.1: Manipulator simulated model structure, A) using RSTB (Robotics system toolbox by MATLAB), B) using MFB (MATLAB function Block).

3.2 Inverse Dynamics Control

The selection of this controller approach was made based on the future use of the robot, as well as the requirement from *ABB* which was to use a joint space controller, and not an operational space controller. In other words, the reference framework is to use a control of nonlinear multi-variable systems. The structure of this control method is shown in the figure below Figure 3.2.

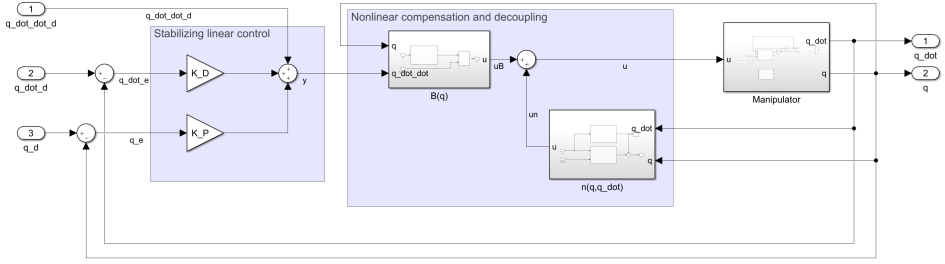


Figure 3.2: Joint space inverse dynamics control

The control structure can be divided into two parts, first the nonlinear compensation and decoupling, and the second part is the PD-controller.

3.2.1 Nonlinear Compensation and Decoupling

The purpose of the nonlinear compensation and decoupling is to make it possible to control each joint by separating the input/output-signals [14]. In other words, without any cross-connection between the signals. Additionally to that, to linearize the system in order for it to gain better and easier control over the variation in the input/output relationship. The desired performance of this part is not only an approximate linearization, but an exact linearization of system dynamics, obtained by means of a nonlinear state feedback [14]. This can be illustrated in figure Figure 3.3.

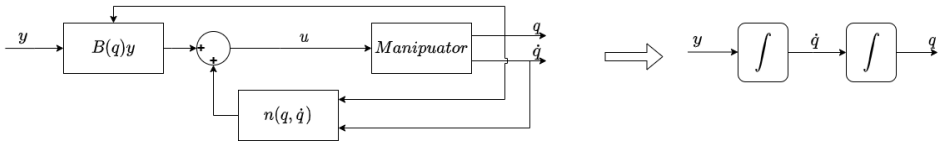


Figure 3.3: Exact linearization of the system dynamics.

The linearization is guaranteed to be found by the particular form of system dynamics. In fact, using the equation in (2.32) gives this relationship:

$$B(q)^{-1}(u - C(q, \dot{q})\dot{q} - G(q)) = y$$

$$\ddot{q} = y$$
(3.2)

This nonlinear control law in (3.2) is called the inverse dynamics control, since it is based on the computation of manipulator inverse dynamics. The system itself under the control of the equation (3.2) is linear and decoupled with respect to the new input y . In other words, the component y_i can only influence the joint angle q_i , with a double integrator relationship, independently of the motion of the other joints.

3.2.2 PD-Controller

The control problem, after the nonlinear compensation and decoupling, is reduced to that of finding a stabilizing control law y , which can be formulated as a PD-controller using the following expressions:

$$y = r - K_P q - K_D \dot{q} \quad (3.3)$$

with the knowledge of $y = \ddot{q}$, a reformulation of the equation can be done as a second-order equation:

$$\ddot{q} + K_D \dot{q} + K_P q = r \quad (3.4)$$

with the assumption of positive definite matrices K_P and K_D , is asymptotically stable. That means when the solutions of the controller, starts to get close enough to the equilibrium it does not only remain close enough, but also eventually converge to the equilibrium. That gives the following choice of K_P and K_D as diagonal matrices (3.5).

$$\begin{aligned} K_P &= \text{diag}\{\omega_{n1}^2, \dots, \omega_{nn}^2\} \\ K_D &= \text{diag}\{2\xi_1 \omega_{n1}, \dots, 2\xi_n \omega_{nn}\} \end{aligned} \quad (3.5)$$

The matrices K_P and K_D are a diagonal, because it is a decoupled system. In other words the reference signal r_i influences only the joint angle q_i with a second-order input/output relationship. This relationship is characterized by a natural frequency ω_{ni} and a damping ratio ξ_i . With that said, to track a path given by a desired trajectory q_d and the output q is ensured by choosing:

$$r = \ddot{q}_d + K_D \dot{q}_d + K_P q_d \quad (3.6)$$

and by substituting (3.6) into (3.4) and using $\tilde{q} = q_d - q$ gives the homogeneous second-order differential equation (3.7):

$$\ddot{\tilde{q}} + K_D \dot{\tilde{q}} + K_P \tilde{q} = 0 \quad (3.7)$$

where \tilde{q} is the error between the desired angle q_d and actual q angle. This kind of error only occurs if $\tilde{q}(0)$ and/or $\dot{\tilde{q}}(0)$ are different from zero and converges to zero with a speed depending on the chosen matrices K_P and K_D .

Lastly the implementation of this kind of control scheme requires computation of the inertia matrix $B(q)$ as well as the gravitational, damping, Coriolis and centrifugal terms vector $n(q, \dot{q})$ to be computed on-line. This is because the control is based on nonlinear feedback of the current angle, therefor it is not possible to be pre-computed.

4

Application to the Robot

This chapter includes the methods that were tested to implement the dynamical model, as well as the control algorithms to control and simulate the robot according to the given parameters from *ABB*. All the following implementation methods for the controller as well as the simulated model, use *MATLAB* as a programming platform [9]. The first method uses *MATLAB* symbolic toolbox to implement all the kinematic and dynamical functions by direct implementation [8]. The second method makes use of *Robotics toolbox* (RTB) [12], which is an open-source toolbox for *MATLAB* created by *Peter Corke*, for the study and simulation of robotics, such as arm-type robot manipulators and mobile robots. The last method uses the *Robotics system toolbox* (RSTB) [7], which is created by *MathWorks* as an additional toolbox for *MATLAB* that requires a license. The Robotics toolbox and Robotics system toolbox will be referred to as RTB and RSTB in this chapter.

The first two methods were studied but not implemented. Therefore, those two methods will only be introduced briefly with the essential weaknesses and benefits.

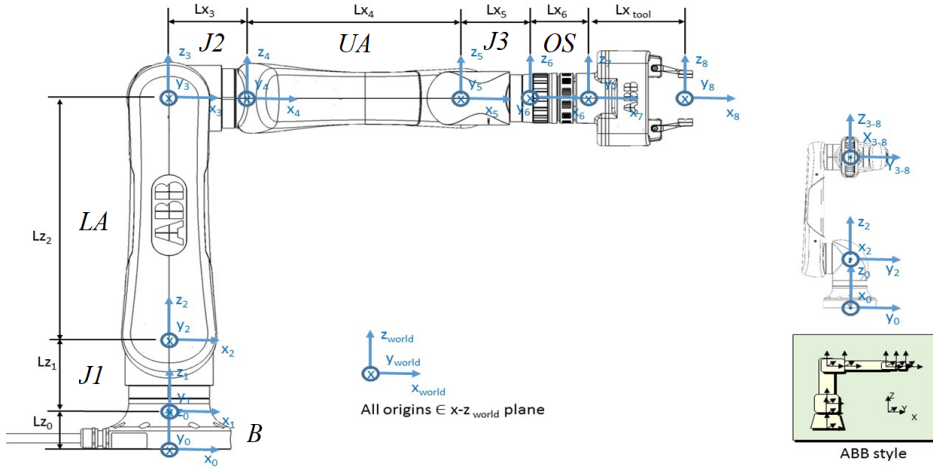


Figure 4.1: The robot and the reference frames for each body link.

The parameters that were available for the robot are the masses of each links, rigid-body mass center position, links lengths as well as the inertia of the links. All the characteristic parameters for the robot are given in the link reference frame, as seen in Figure 4.1 above.

4.1 Direct Implementation

The first approach that was considered and tested to be implemented was the direct implementation using the symbolic toolbox. It is because of the convenience of using the symbolic toolbox, which also provides the ability to manipulate the system definition more freely. That was done by firstly identifying the DH parameters to be as shown in Figure 4.2, according to the convention mentioned in section 2.3.

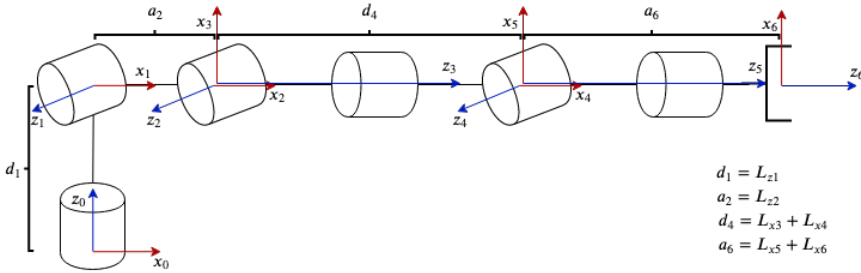


Figure 4.2: The joint frames according to the DH convention.

By using the parameters shown in Table 4.1, the expression of the homogeneous transformation matrix for each link can be obtained by using equation (2.9).

Table 4.1: The DH parameters used for the direct implementation and RTB approaches.

Link (i)	θ_i [rad]	d_i [m]	a_i [m]	α_i [rad]
1	θ_1	0.185	0	$\frac{\pi}{2}$
2	θ_2	0	0.380	0
3	θ_3	0	0	$\frac{\pi}{2}$
4	θ_4	0.420	0	$-\frac{\pi}{2}$
5	θ_5	0	0	$\frac{\pi}{2}$
6	θ_6	0	0.318	0

Thereafter computing the Jacobian matrix using (2.13), which gives all the variables needed to compute the mass-matrix $B(q)$ and the Coriolis matrix $C(q, \dot{q})$ using (2.19), (2.26) and (2.27). Lastly computing $G(q)$ using the equation for the potential energy (2.20).

The problem with this method was the number of links, number of DoF, which is six, which makes the expression of the mass matrix and Coriolis-matrix very long. That resulted in difficulty for the computer to compile the expression into a MFB (Matlab Function Block), requiring a lot of computing power and time. The decision to select the MFB was made due to the ability of the MFB to be code generated to C-language for the installation on the hardware of robot. Therefore,

this approach was discarded, and the focus was instead put on selecting a toolbox without this weakness. That led to chose RTB, since it is an open source toolbox.

4.2 Robotics Toolbox RTB

In the RTB the kinematic and dynamic functions are predefined, but it requires the definitions of links according to the DH parameters convention, with all other data of the links such as mass, inertia and center of gravity etc. See Appendix A for the code of the link declaration in *MATLAB*. Thereafter, a *Simulink* "Interpreted MATLAB Function-Block" was created for the computation of the Mass-matrix $B(q)$, coriolis-matrix $C(q, \dot{q})$, gravity-vector $G(q)$ as well as one block for the simulated model of the robot.

The problem with this approach is the predefined functions and "Interpreted MATLAB Function-Block" are not compatible to *MATLAB* code-generation function. That means it has to be rewritten into a *MATLAB Function block* or in *C*-language to be able to install it into the hardware of the robot. Another problem in RTB is that it gives comparatively slow simulations. This problem worsens the more links the system has. But the main problem is that it did not have any validation of the given data. In other words, it is not known if the input data is appropriate or not. This imperfection was mainly noticed in the input data of the inertia. But it has very educational and user friendly 3D representation of the link configuration. That was used to confirm the identification of DH parameters, but due to its imperfection and weaknesses and the slow response, this approach was also discarded. That led to testing the *Robotic System Toolbox* (RSTB). RSTB is *MATLAB*'s own toolbox, which required a license to use, that is why it was the last choice.

4.3 Robotic System Toolbox RSTB

The RSTB also has the kinematic and dynamic functions predefined, however it does not necessarily require the definition of the links according to the DH convention. Additionally to that it has also the *Simulink* block with those function predefined. The benefits of the RSTB is that it checks the given data to the link, which makes it more trustworthy than the RTB. The RSTB is also well optimized, that results in faster simulations than RTB. The model below, Figure 4.3 shows how the controller was built in the *Simulink* using the RSTB.

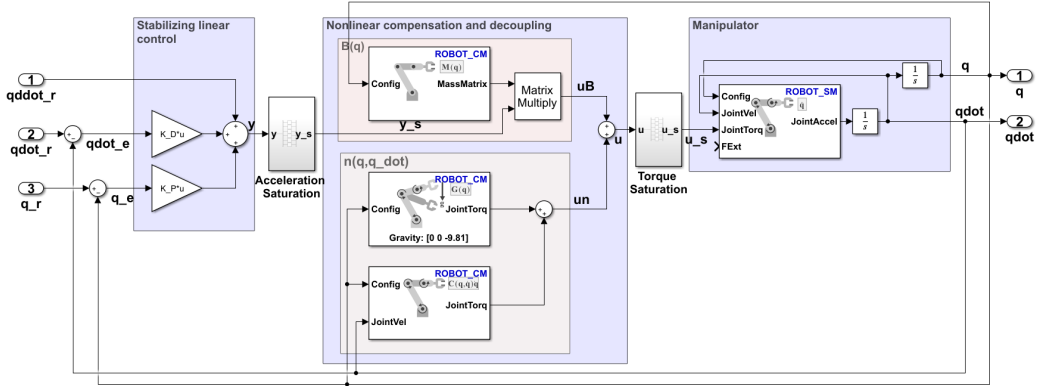


Figure 4.3: Simulinkmodel of the robot controller using RSTB, further described in Appendix B.

In addition to the control approach, which was mentioned in section 3.2, Figure 3.2, two saturation blocks were added. This was done considering the hardware of the robot and the required behaviors. One saturation block was added to give a limitation to the output signal, torque signal, of the controller and another one was added to limit the control signal to the "Nonlinear Compensation and Decoupling". Furthermore the definition of the links was made according to the given data from ABB, which is not following the DH convention as shown in Figure 4.1. The most noticeable difference to the DH convention, is that not all the reference frames pointing in the same direction of the links rotation axis etc. see Appendix C for the code of the link declaration in *MATLAB*. This approach seen to be the most trustworthy one of all the other three approaches. With that said, a validation of the model as well as the controller are required to justify the results of this thesis.

5

Simulation Results

To evaluate the result of the controller method as well as the implementation approach that was chosen using RSTB, there were four tests considered. First, to test the capability of the output signal to converge to the reference signal. This test will be repeated while facing a noise in the measured output signal, which is fed back to the controller. Thereafter, testing the robustness of the system, if the given data is not fully accurate. That is done by modifying the model parameters of the simulated model. Finally, testing the accuracy of a path following in the operational space.

5.1 Experiment Baseline Case

Every single robot that will be set into a work-space requires thorough testing. Specially if it will be in close contact with fragile components or even humans for that matter. They are tested for their capability of performing the task they were intended for. Therefore, the simulated model as well as the controller approach will be tested. The tests are divided into four cases:

- Ideal case,
- Measurement disturbance,
- Robustness,
- Operational space time path.

The first case will test the capability of the controller to follow the reference signal. This case will also work as a baseline case for the other cases. Therefore, a comparison will be made between the first case and each one of the other cases.

That will give a good estimate of how the system will be affected by those different case scenarios.

The ideal case was chosen to be a sequence of steps in the joint angles. The steps sequence starts at second one with the first joint (J1) and each second a new joint takes a step until the last joint (OS) makes the last step in the sixth second. For more clarification of which link is connected to which joint see Figure 4.1. The amplitudes of the steps are different in some of the joints. These differences in the steps amplitude are due to the maximum allowed acceleration that each joint can have. The chosen amplitude of the joints steps are $q_{i,amp} = (15, 30, 30, 90, 90, 90)$ degree, which is $q_{i,amp} = (0.2618, 0.5236, 0.5236, 1.5708, 1.5708, 1.5708)$ rad.

To give a better reference velocity and acceleration to the controller a filter was added to the angle reference signal and thereafter the filtered step first and second derivative was used as the reference velocity and acceleration. This filter $H(s)$ was chosen to have a rise time less than one second and without any overshoot.

$$H(s) = \frac{4}{0.05s^2 + s + 4} \quad (5.1)$$

The outcome of the ideal case is shown in the graphs below:

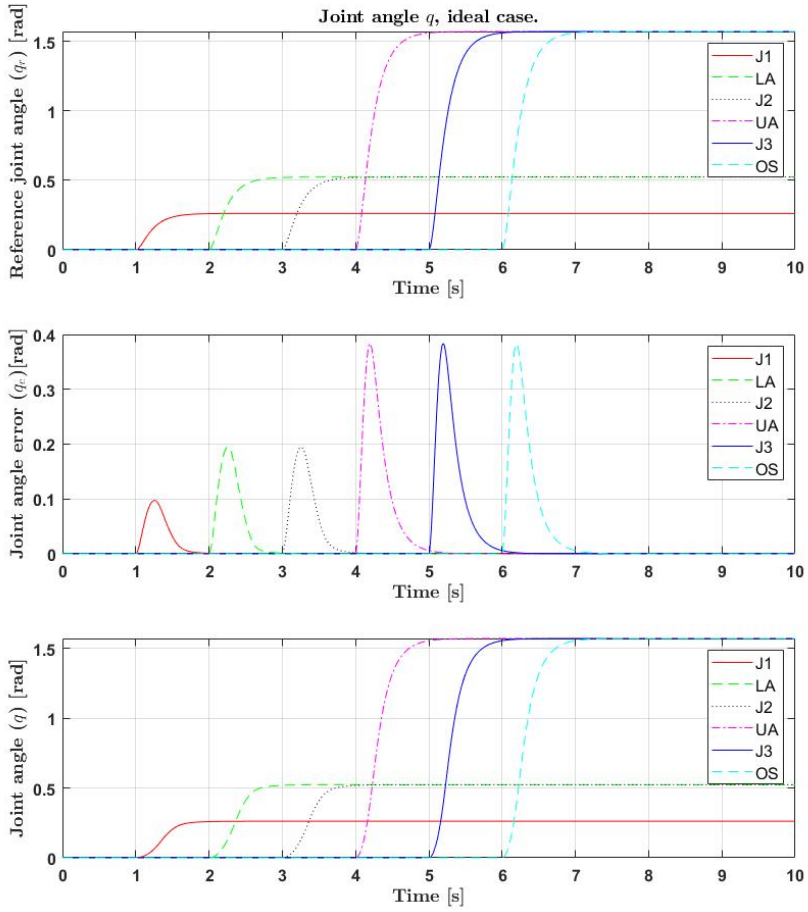


Figure 5.1: Ideal case. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

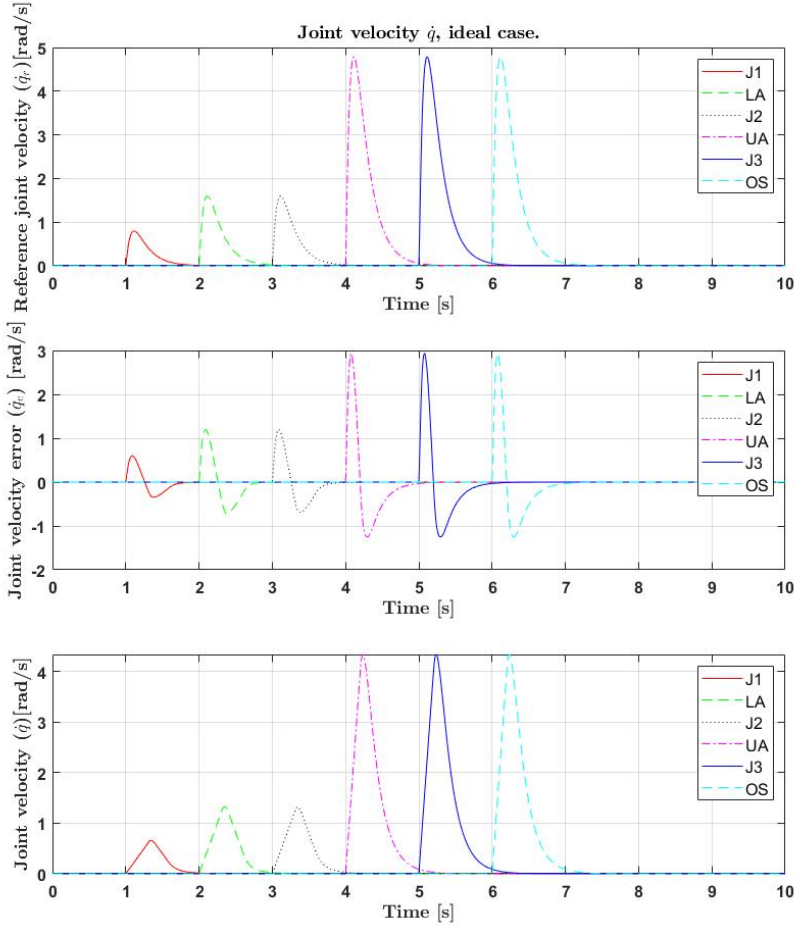


Figure 5.2: Ideal case. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

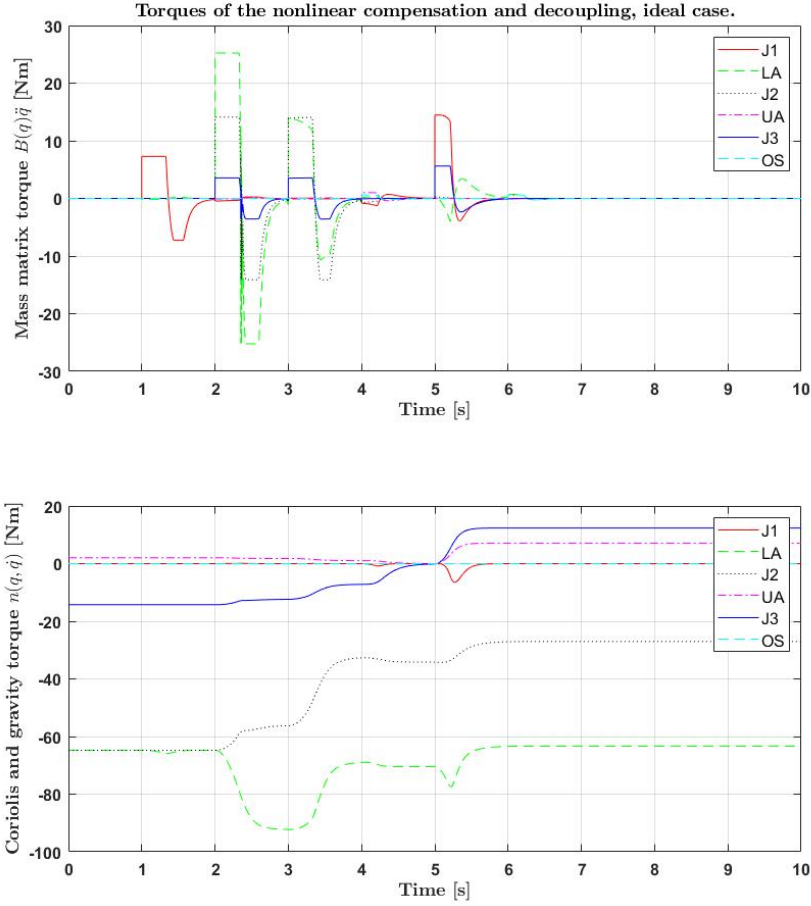


Figure 5.3: Ideal case. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

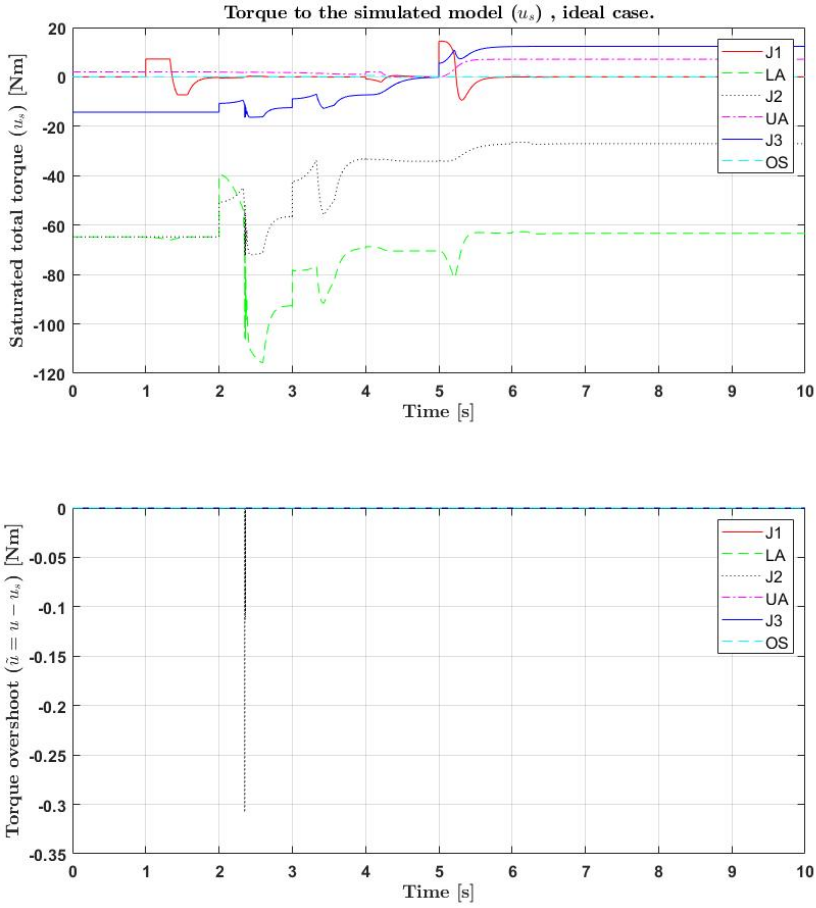


Figure 5.4: Ideal case. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

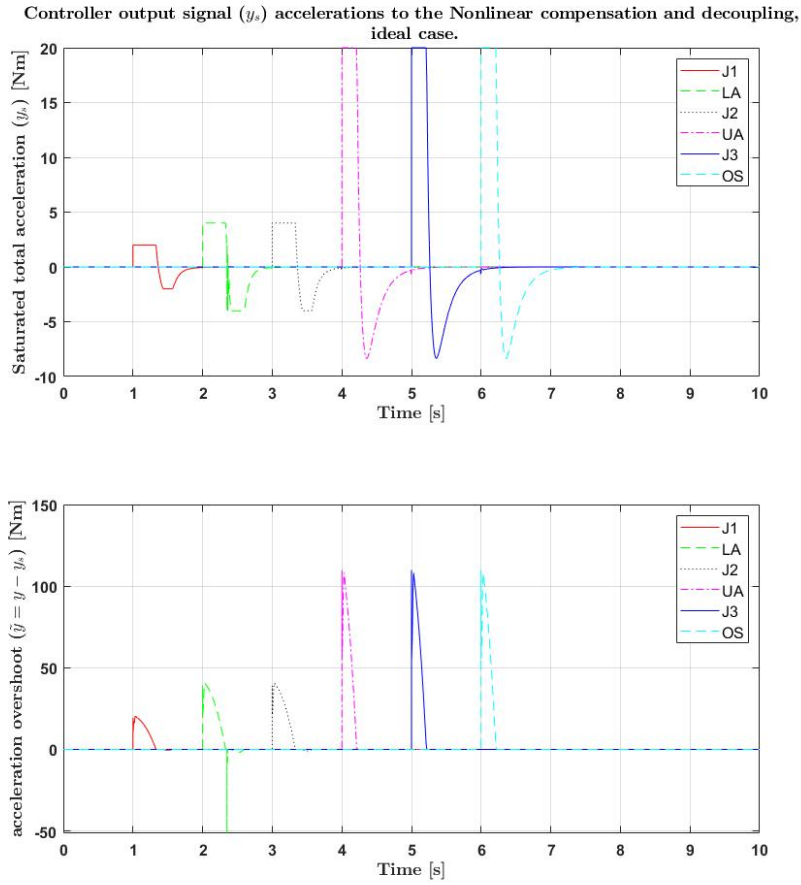


Figure 5.5: Ideal case. Top graph: total saturated control signal y_s . Second graph: saturated values of y .

The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

The ideal case scenario shows that this control method is capable of following the reference angles, see Figure 5.1. The occurrence of error in the angle is related to the saturation of both the acceleration and torque, and also the response time for the controller to react to the error. These occurrences can be seen when comparing the error graphs in Figure 5.1 and the acceleration limitation on Figure 5.5, also when comparing the acceleration limitation in Figure 5.5 and torque limitation in Figure 5.4. Although the error is less than the 40% of the step amplitude, at the start of the step. That is acceptable with the knowledge that the error converges to zero after one second.

5.2 Measurement Disturbance Influence on the Controller

Testing the performance of the controller while facing noise to the output signals is necessary. This is due to the fact that there will be noise distortion occurring in the output signal. Therefore, it is necessary to know the impact that this disturbance has, on the system and its behaviour.

To test how the noise might affect the performance of the controller, a normal distributed noise generator *Band-Limited White Noise* block was added to each one of the output signals, which are the joint angles q and also to the joint velocities \dot{q} . The maximum amplitudes of the noise to the joint angles q were chosen to be $1.5 \times 10^{-3} \text{ rad} \approx 0.085 \text{ degrees}$, and $0.05 \text{ rad/s} \approx 2.9 \text{ degree/s}$ for the joint velocity. These values are anticipated to occur in the real system and thus chosen to be simulated. Figure 5.6 shows the *Simulink* structure of how the noise was generated. The output of this structure is added to the joint angle, in the same way the noise was generated for the joint velocity.

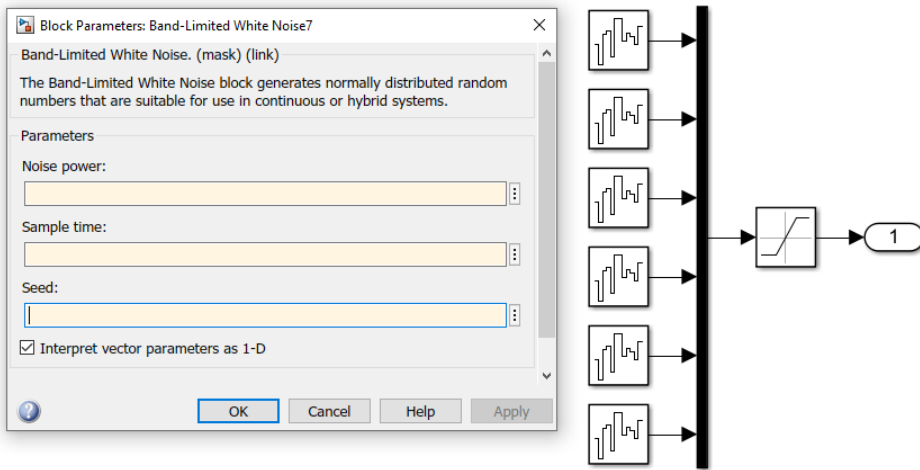


Figure 5.6: Noise generator structure.

The "Noise power" parameter is set to 5×10^{-9} to get a noise amplitude around $1.5 \times 10^{-3} \text{ rad}$ for the joint angle. The "Sample time" is set to 0.01 s for both joint angle and joint velocity. The "Seed" parameter is set to different values in each noise generator to give different values for each noise generator in the same time stamp. To ensure that the amplitude does not exceed the maximum value for joint angle and joint velocity, a saturation block was added after the noise generator with the limits $\pm 1.5 \times 10^{-3} \text{ rad}$ and $\pm 0.05 \text{ rad/s}$ respectively.

The graphs below show the test results when the noise is added in both the joint angle q and joint velocity \dot{q} . The graphs for the test results, where the noise is only added to the joint angle q or only to the joint velocity's \dot{q} are included in the Appendix D. The reference signal is the same in all these cases, which is also similar to the ideal case.

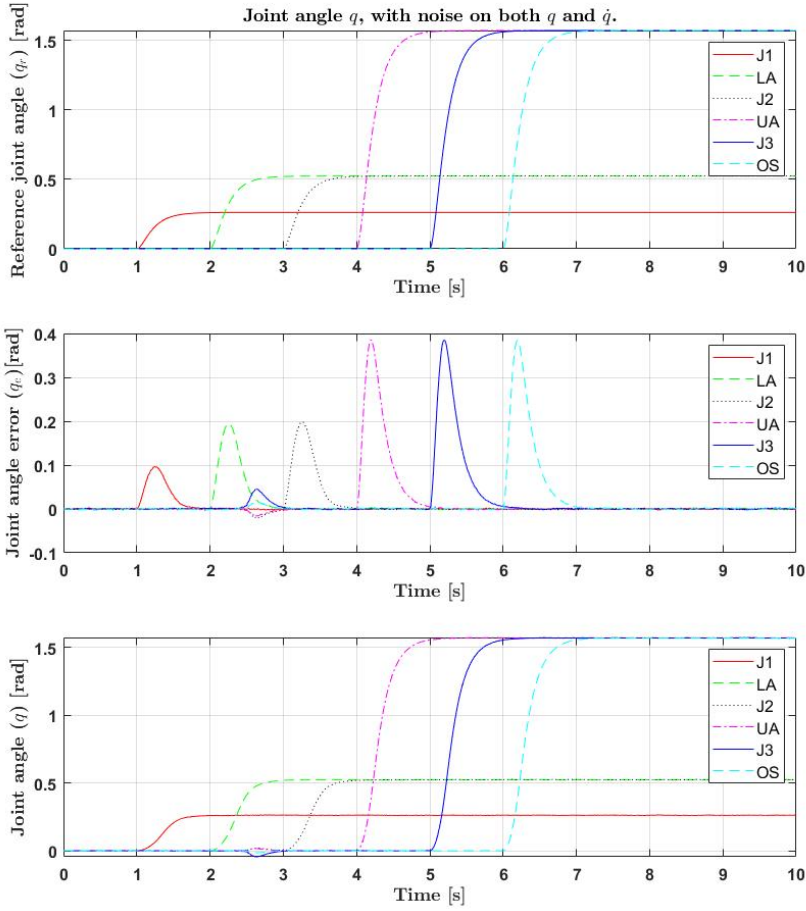


Figure 5.7: Disturbance case: added noise on joint angle q and joint velocity \dot{q} . From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q .

The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

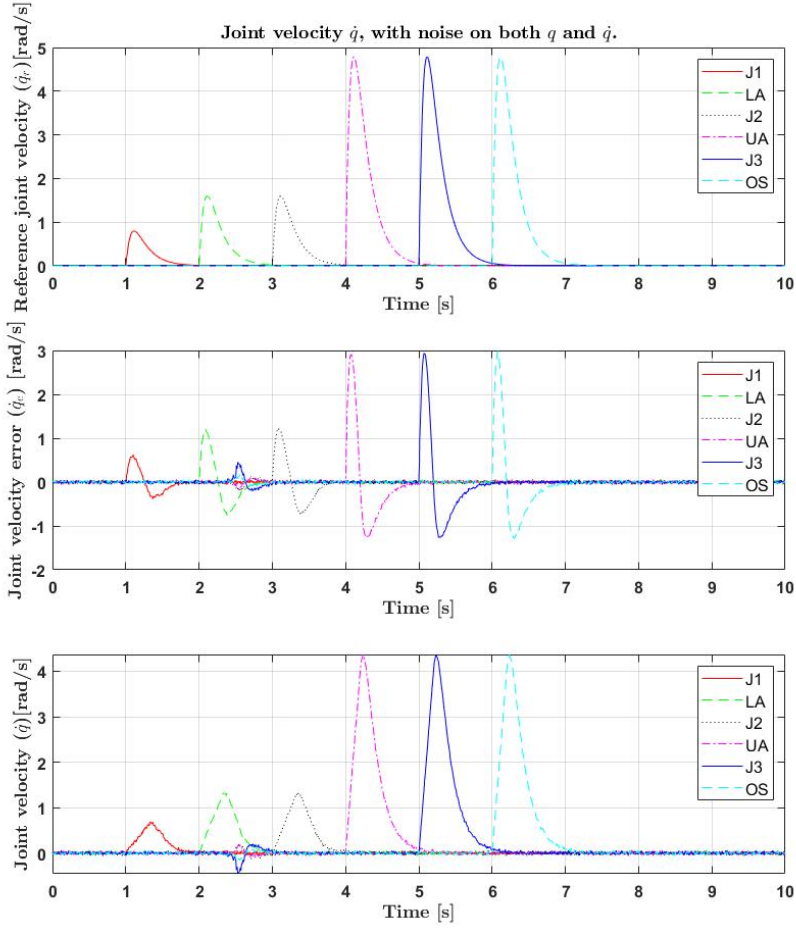


Figure 5.8: Disturbance case: added noise on joint angle q and joint velocity \dot{q} . From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

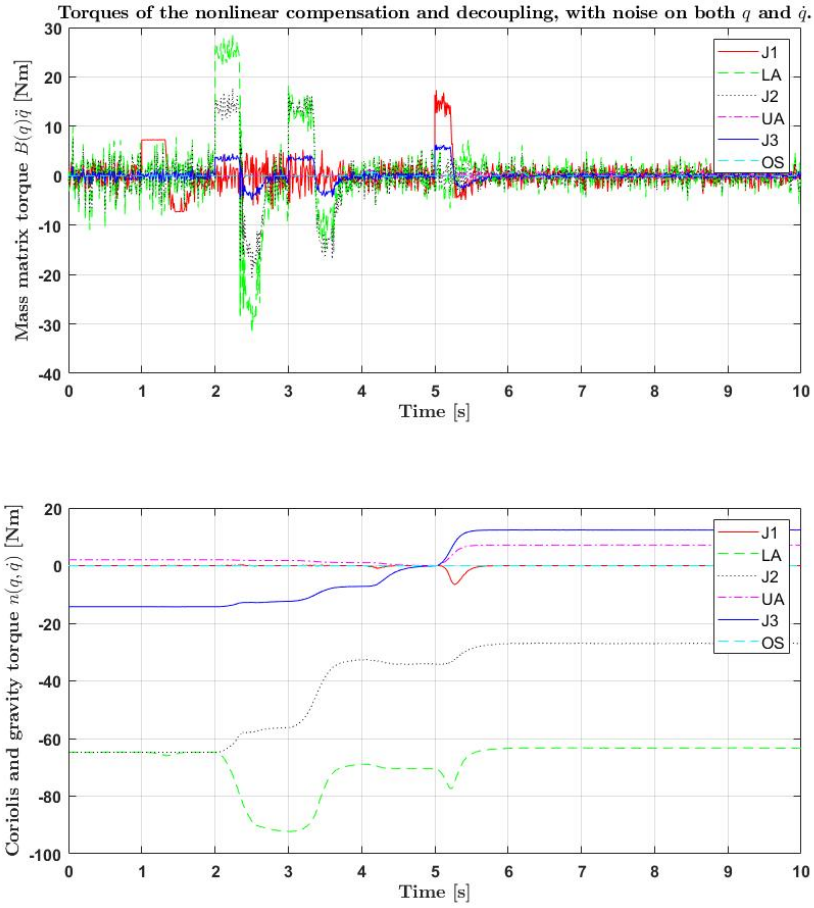


Figure 5.9: Disturbance case: added noise on joint angle q and joint velocity \dot{q} . Top graph: torques from the $B(q)\dot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

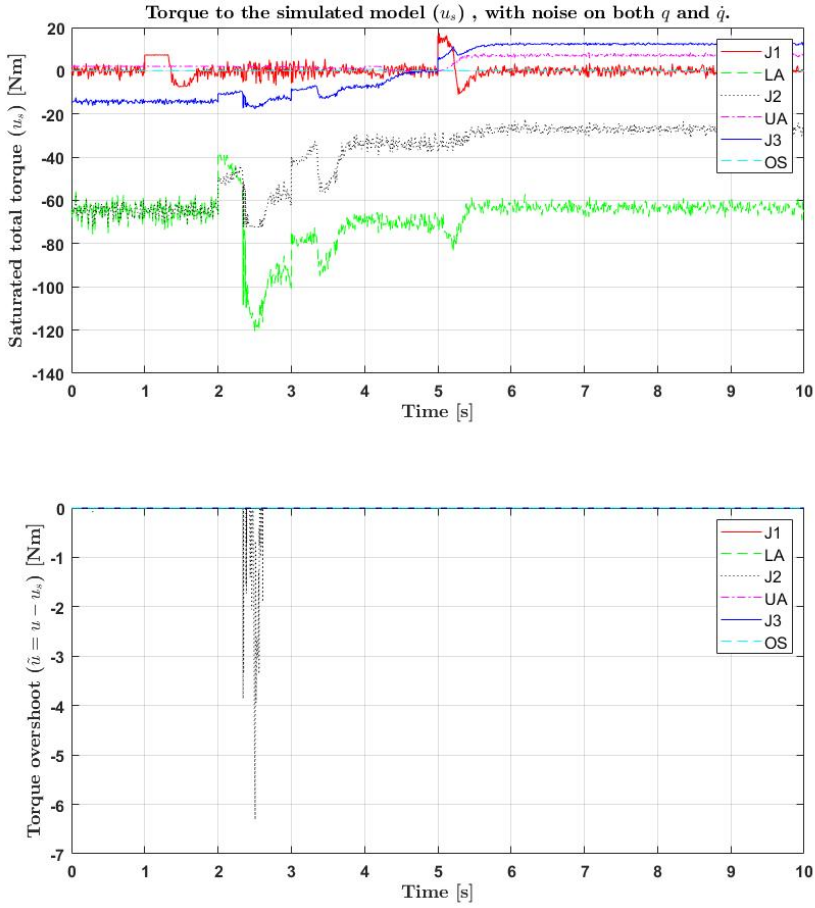


Figure 5.10: Disturbance case: added noise on joint angle q and joint velocity \dot{q} . Top graph: total saturated torques u_s . Second graph: saturated values of u .

The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

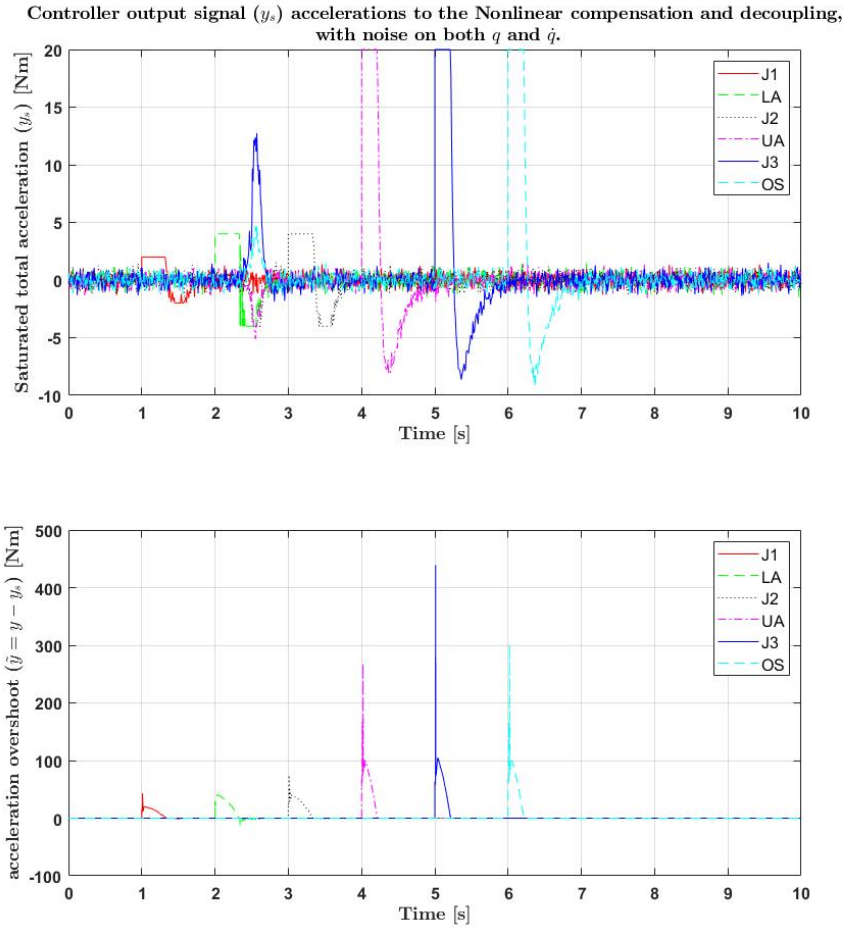


Figure 5.11: Disturbance case: added noise on joint angle q and joint velocity \dot{q} . Top graph: total saturated control signal y_s . Second graph: saturated values of \hat{y} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

The result shows that the error converging to zero, with some minor noise. Otherwise, the error kept to the same as the noise amplitude except for some curves, see Figure 5.7 and Figure 5.8 between the 2nd and 3rd second. Those curves on the joint angle and velocity depend on the control signal, the acceleration y , which gets disturbed by the noise and gives some peaks in the same time period. With that said, the graphs indicate that the system is more sensitive to the noise on the joint velocity \dot{q} than the noise for the joint angle q , see Appendix D. Specially

when comparing the behaviors of the system when a noise signal was added to either a joint angle or joint velocity, see Appendix D. The amplitude of the error that is influenced by the noise is the same as the noise amplitude, when the noise is added only to the joint angle. However, the error gets bigger when the noise is added on both the joint angle and joint velocity, see Table 5.1.

Table 5.1: Noise amplitude, depending on where the noise is added.

Added noise on	Error noise amplitude on q	Error noise amplitude on \dot{q}
q	$1.5 \times 10^{-3} \text{rad}$	$4 \times 10^{-3} \text{rad/s}$
\dot{q}	$< 1.5 \times 10^{-3} \text{rad}$	$50 \times 10^{-3} \text{rad/s}$
q and \dot{q}	$3 \times 10^{-3} \text{rad}$	$50 \times 10^{-3} \text{rad/s}$

The mass-matrix torque $B(q)\ddot{q}$ tends to oscillate, in other words adapt to the noise. Whilst the coriolis and gravity torque $n(q, \dot{q})$ is not affected at the same grade by the noise, even if the noise is added to the joint angle, joint velocity or both. This is evident in all three cases, where the signal of the Coriolis and gravity torque $n(q, \dot{q})$ is smoother than the signals of the mass-matrix $B(q)\ddot{q}$. Overall, the controller follows the reference signal and keeps the error 40% less than the step signal, in the same way as the ideal case. This leads to the investigation of the next case scenario, which is the robustness of the model.

5.3 Robustness of Control Model

In the most cases there may occur some errors in the given data. Therefore, the robustness of the controller has to be tested. First to be considered is which parameter, from the given data such as mass, inertia, center of gravity (CoG), payload and links lengths, can for the most part be assumed to be wrong or not precisely correct. The obvious one is the payload mass that can vary from task to task. The second thing to consider is the inertia parameters, which is difficult to achieve with good precision. Last but not least is the CoG, which can also be difficult to estimate with good precision. The masses and the lengths of the links are mostly measured with great precision. Therefore, two tests had to be run, where the controller keeps the given data from the specification of the robot according to ABB and changing the simulated model values. The tests are made by changing the values on:

- The payload by $\pm 5\%$ and $\pm 10\%$.
- The inertia tensor by $\pm 5\%$ and $\pm 10\%$.

The results of those tests, where the variation is only by $\pm 5\%$, are in Appendix E and Appendix F. Hence, the result of the test, where the variation is $\pm 10\%$, are shown below.

5.3.1 Robustness of Payload Influence Test

The test is made by multiplying 0.9, 0.95, 1, 05 and 1.10 to the mass of the last link (OS), which is including the mass of the test weight. Thereafter, a simulation of the behavior is done by giving the same reference signal as the ideal case scenarios. The result of those tests, where the variation is only by $\pm 5\%$, are in Appendix E. Otherwise, the result of the test, where the variation is $\pm 10\%$, are shown below.

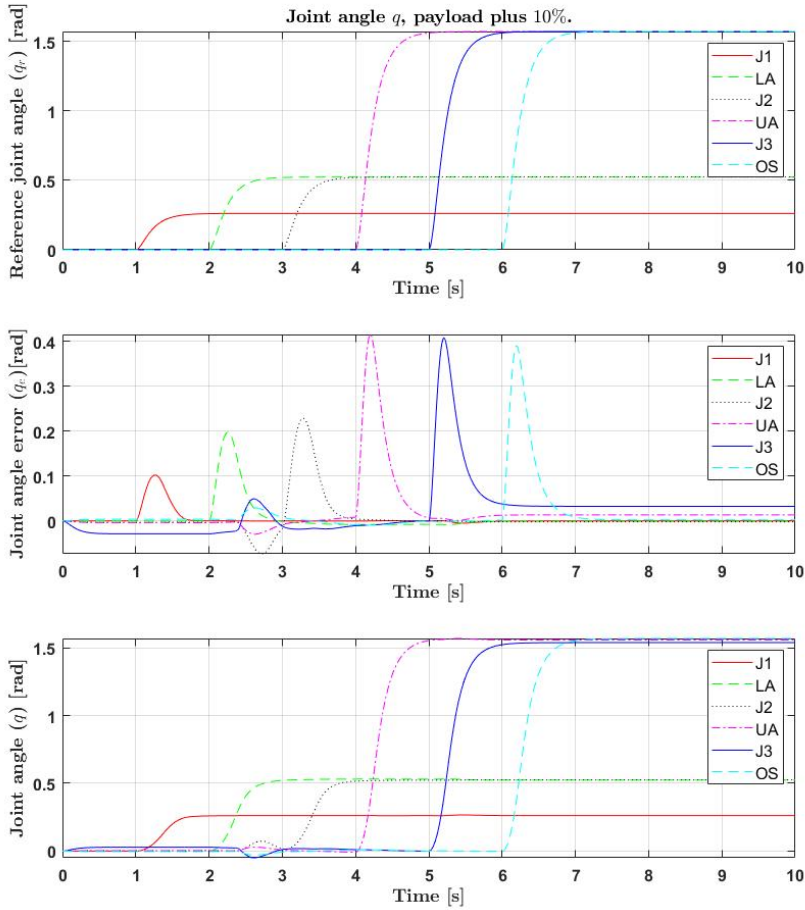


Figure 5.12: Payload test: test weight is 10% higher. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

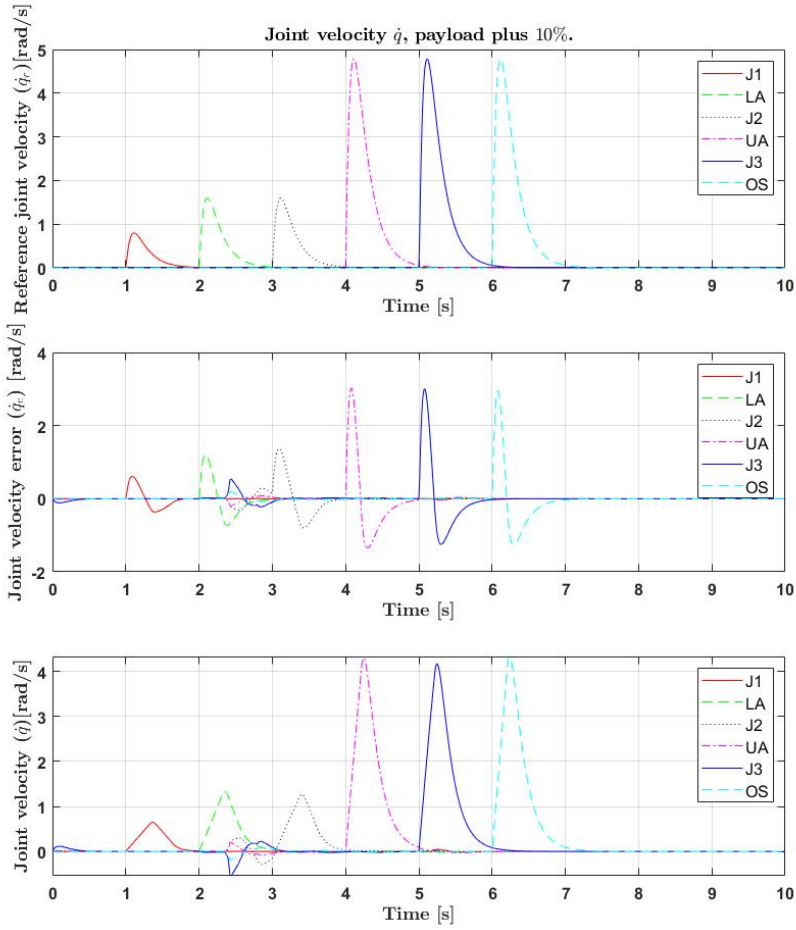


Figure 5.13: Payload test: test weight is 10% higher. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

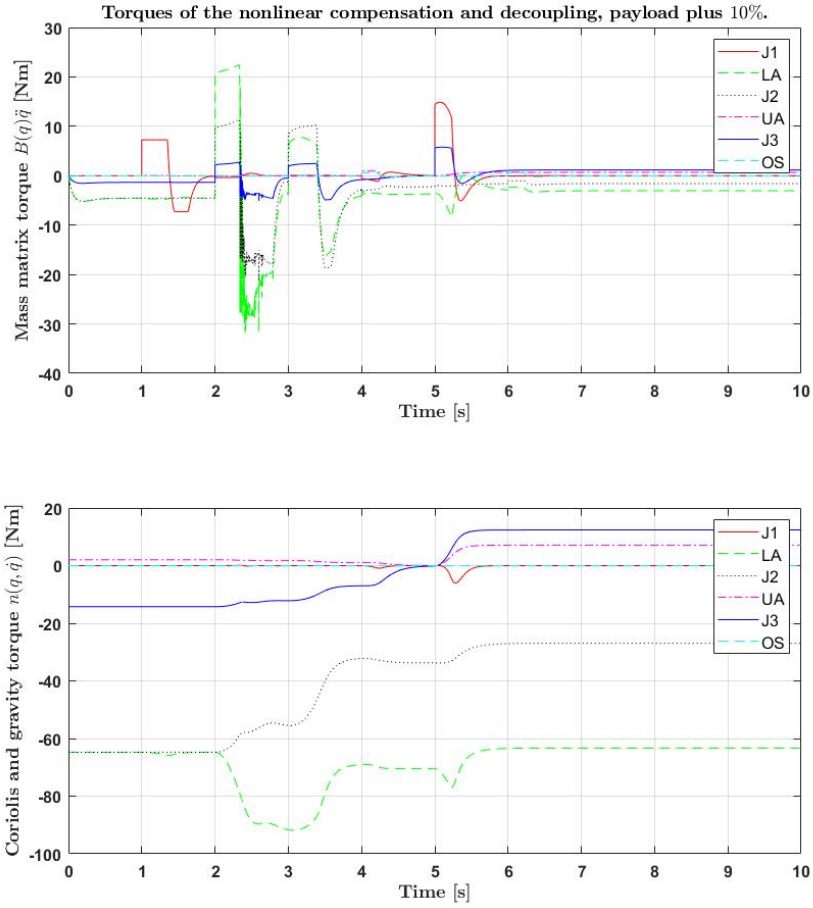


Figure 5.14: Payload test: test weight is 10% higher. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

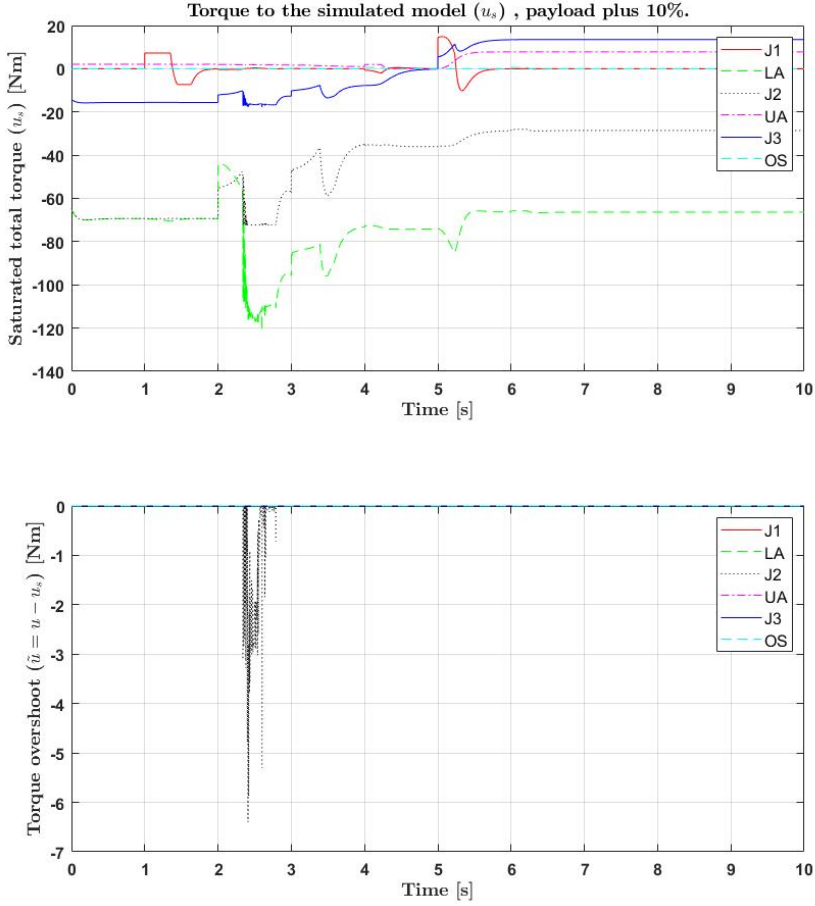


Figure 5.15: Payload test: test weight is 10% higher. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

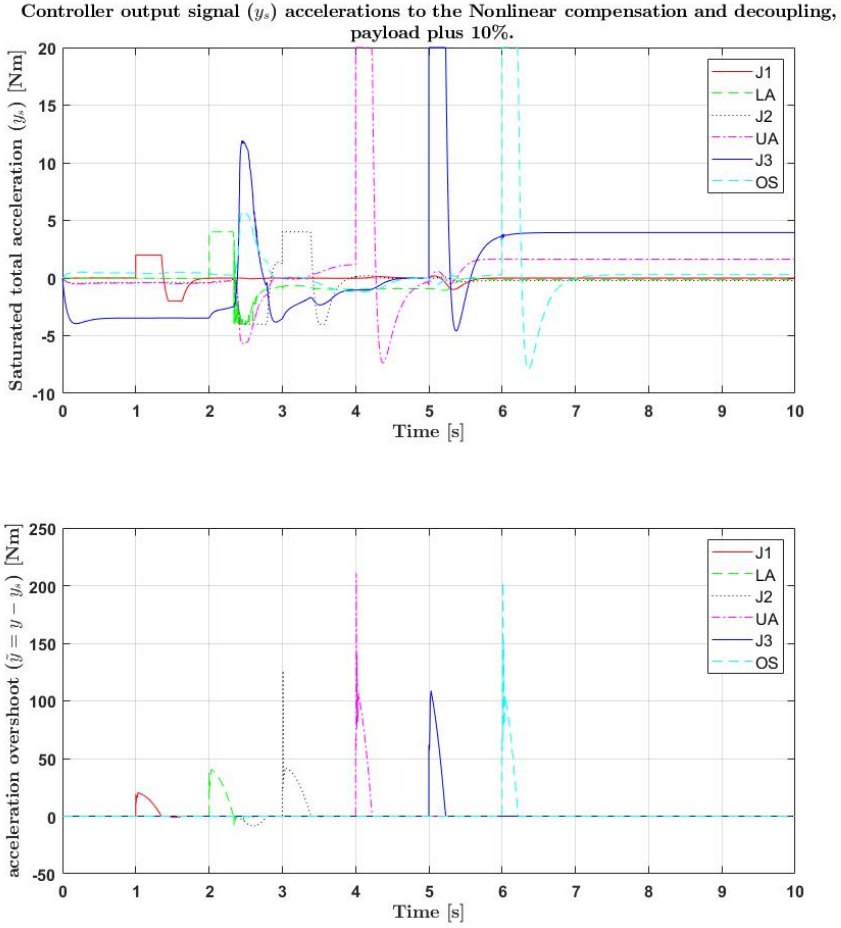


Figure 5.16: Payload test: test weight is 10% higher. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

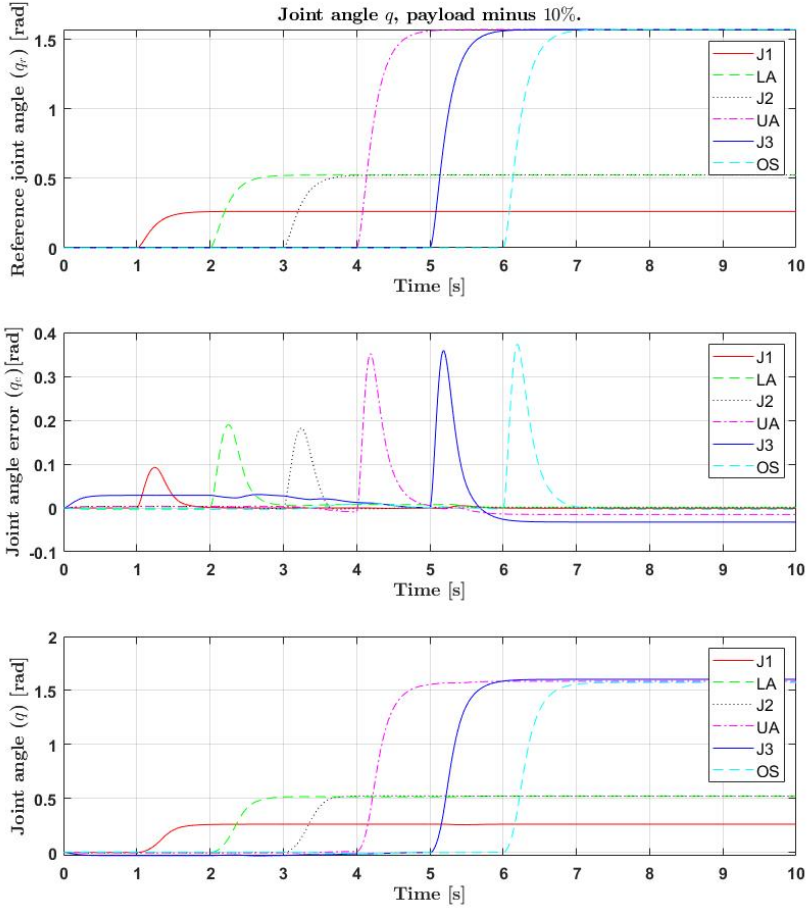


Figure 5.17: Payload test: test weight is 10% lower. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

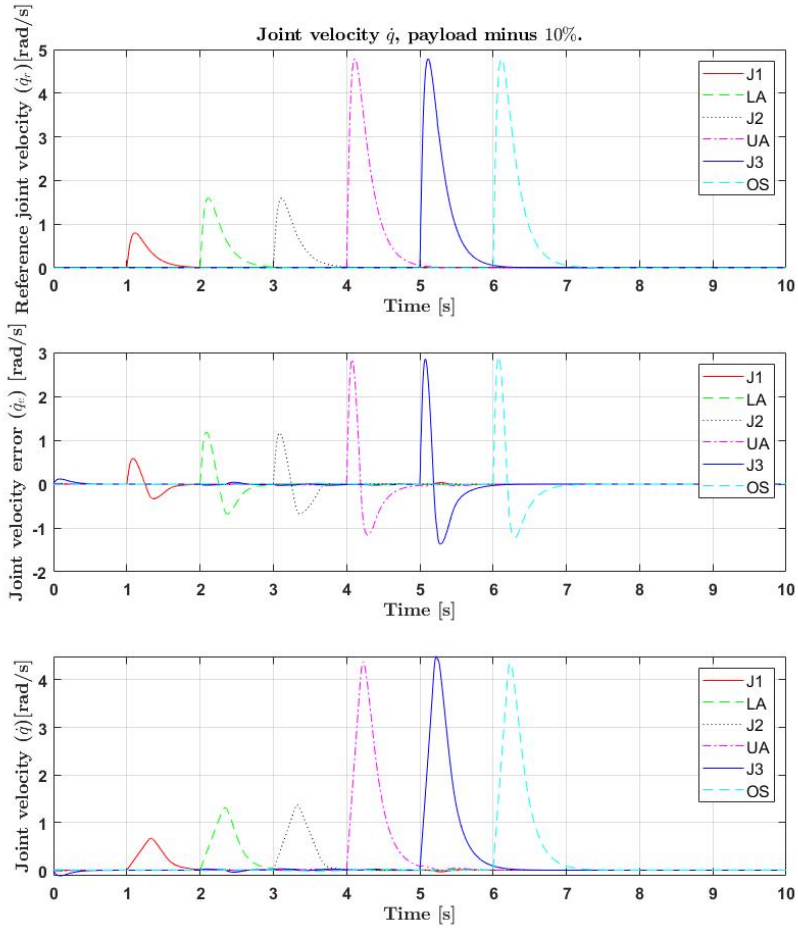


Figure 5.18: Payload test: test weight is 10% lower. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

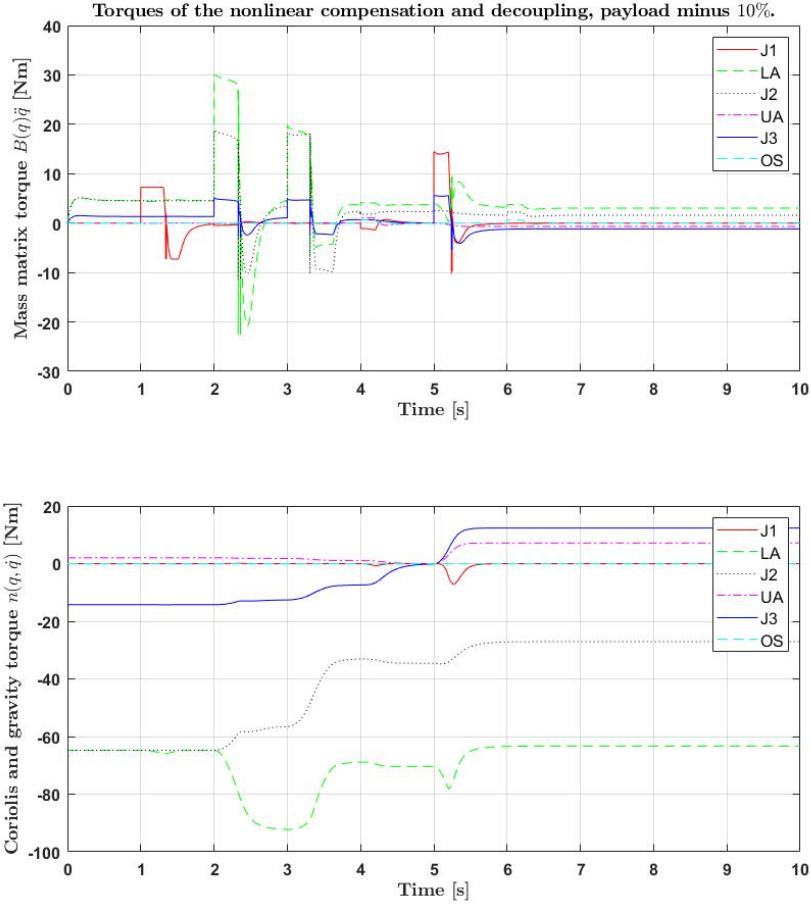


Figure 5.19: Payload test: test weight is 10% lower. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

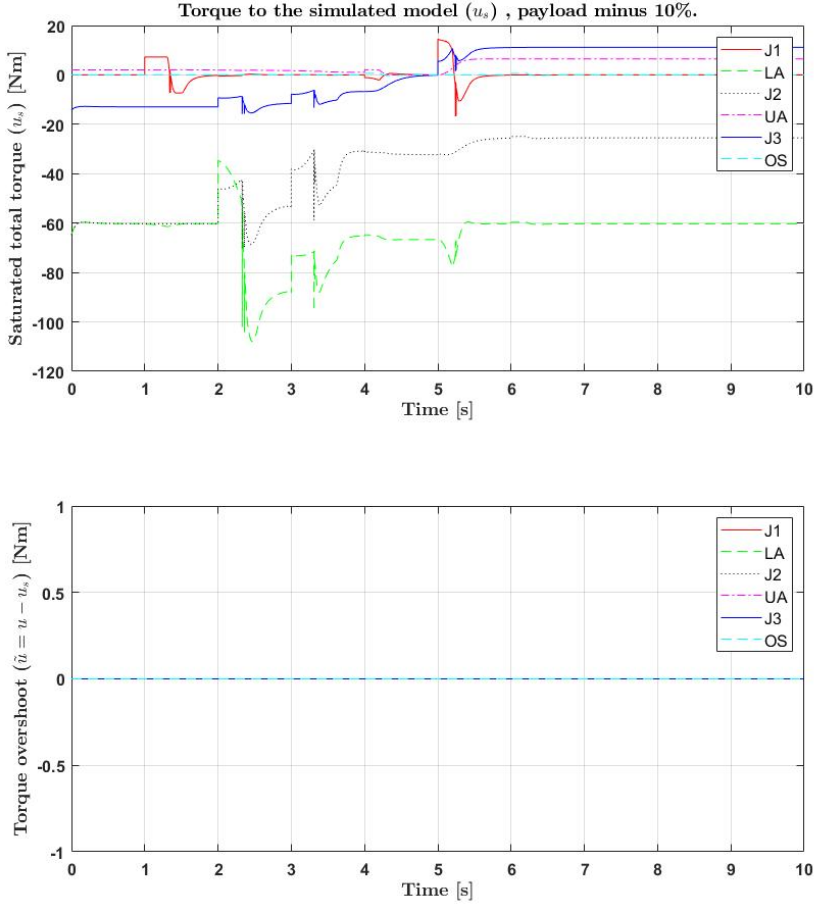


Figure 5.20: Payload test: test weight is 10% lower. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

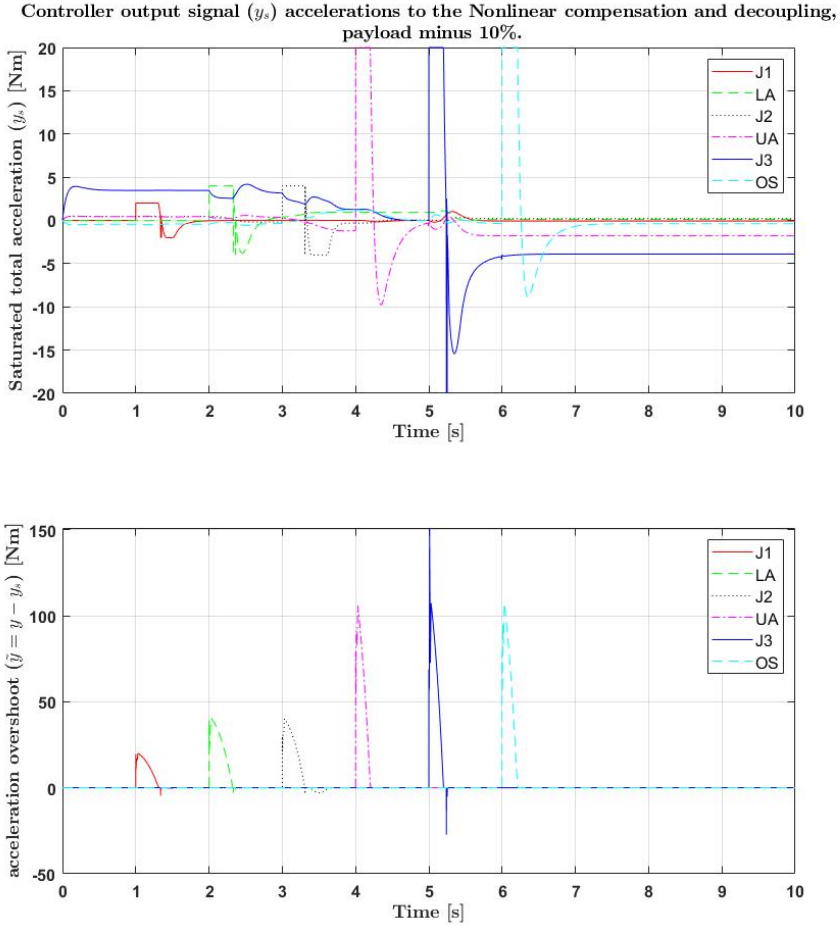


Figure 5.21: Payload test: test weight is 10% lower. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

The results show that a change in payload gives a static error in the joint angle. The error is linearly dependent on the error in the payload, that can be seen when comparing the ideal with all other payload tests (+5%, +10%, -5% and -10%). With that said, the amplitude of the highest static error is $\approx 0.016\text{rad} \approx 0.9167$ degree when the error in mass is $\pm 5\%$ and $\approx 0.032\text{rad} \approx 1.8335$ degree when the error in mass is $\pm 10\%$. With the knowledge that this error in the joint space yields an error in the operational space equal to $0.0047m$, it get acceptable. This error is calculated by (5.2)

$$\begin{aligned} \tilde{H}_e &= H_{e,q} - H_{e,q_r} = \begin{bmatrix} \tilde{R}_e & \tilde{T}_e \\ \mathbf{0} & 1 \end{bmatrix} \Rightarrow \\ \left\{ \begin{array}{l} \tilde{T}_e = \begin{bmatrix} 0.0002 & 0.0002 & 0.0047 \end{bmatrix}^T \Rightarrow \|\tilde{T}_e\| = 0.0047 \\ \tilde{R}_e = \begin{bmatrix} -0.0037 & -0.0087 & -0.0038 \\ -0.0017 & 0.0314 & 0.0136 \\ 0.0352 & 0.0005 & 0.0004 \end{bmatrix} \end{array} \right. \quad (5.2) \end{aligned}$$

where the \tilde{H}_e is the error between the homogeneous transformation matrix of the desired angle q_d and the actual angle q . The homogeneous transformation matrix is for the end-effector w.r.t the base. The \tilde{T}_e and \tilde{R}_e are the rotational and translational component of the \tilde{H}_e . The data in the equation above are from the test where 10% is subtracted to the payload, which has the largest error amplitude in joint angle $\approx 0.032\text{rad} \approx 1.8335$. The difference in the payload also has an impact on how much of the torque is saturated, which is reasonable. The lower the weight is, the less required torque is needed to move and vice versa.

5.3.2 Robustness of Inertia Influence Test

This test was made in the same way as the payload test, where the change was made in the inertia tensors of all the links. The result of those tests, where the variation is only by $\pm 5\%$, are in Appendix F. Otherwise, the result of the test, where the variation is $\pm 10\%$, are shown below.

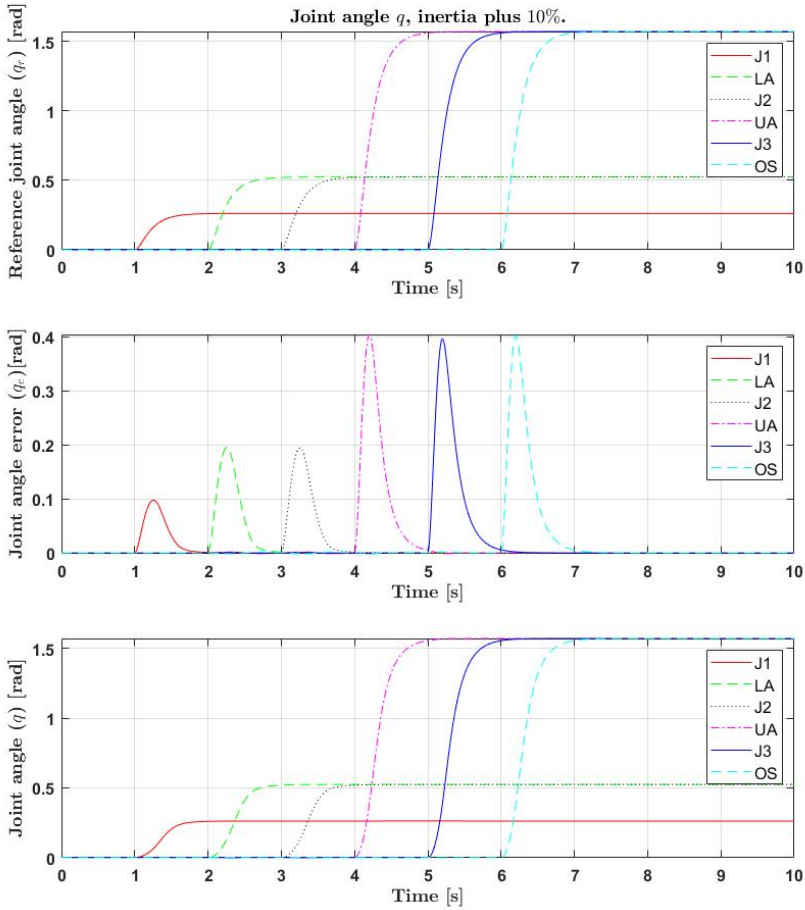


Figure 5.22: Inertia test: inertia tensor 10% higher. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

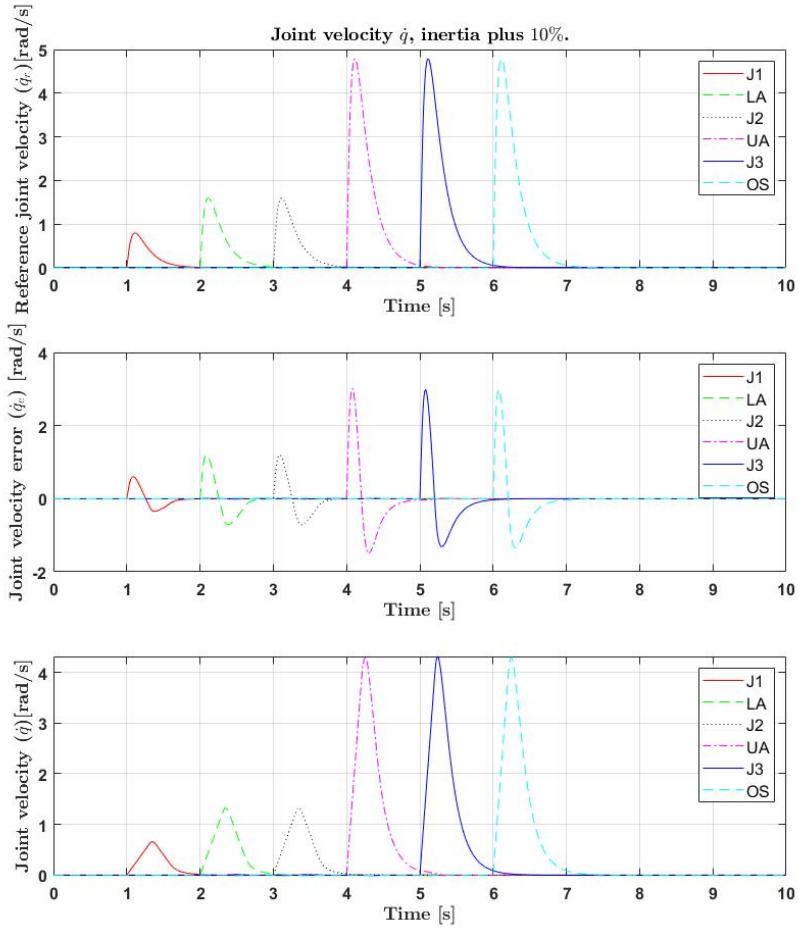


Figure 5.23: Inertia test: inertia tensor 10% higher. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

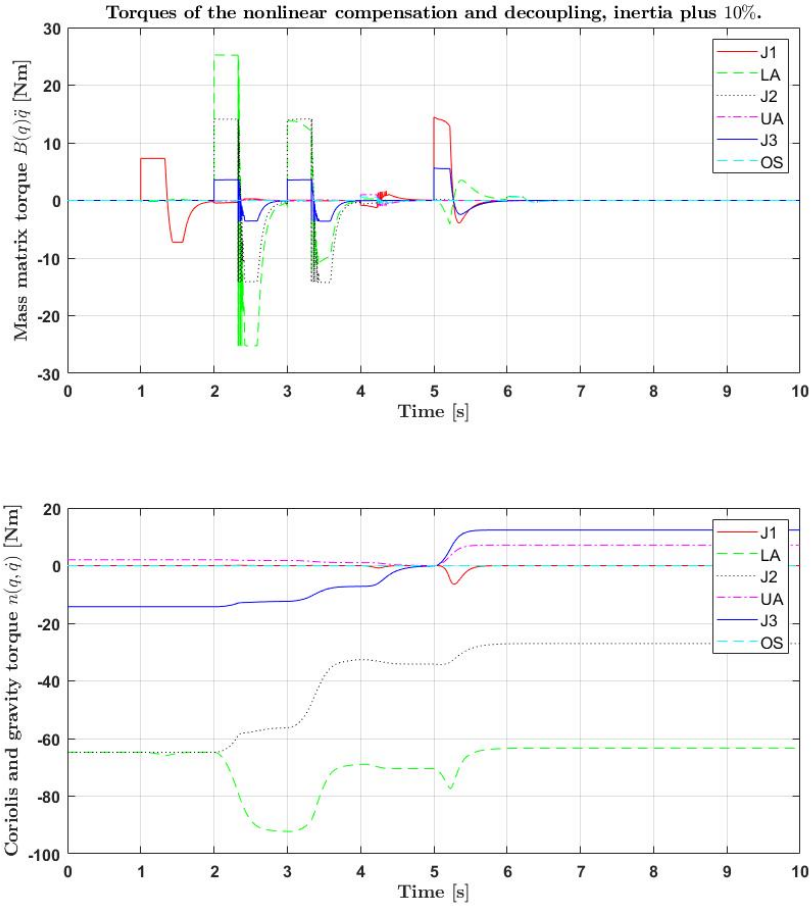


Figure 5.24: Inertia test: inertia tensor 10% higher. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

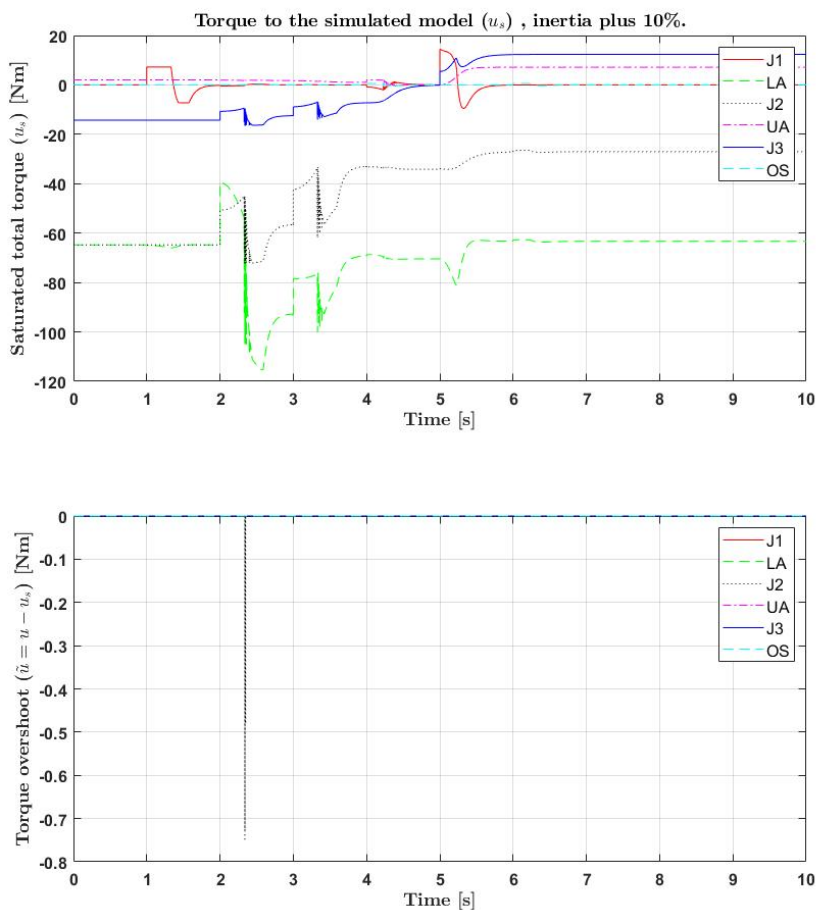


Figure 5.25: Inertia test: inertia tensor 10% higher. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

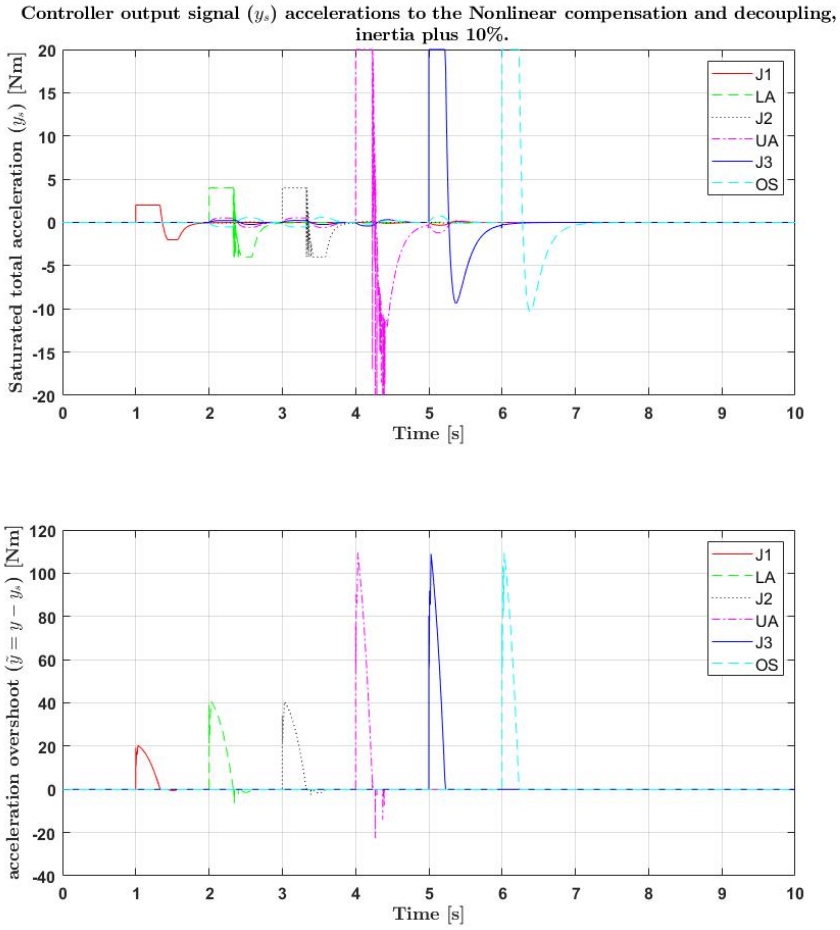


Figure 5.26: Inertia test: inertia tensor 10% higher. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

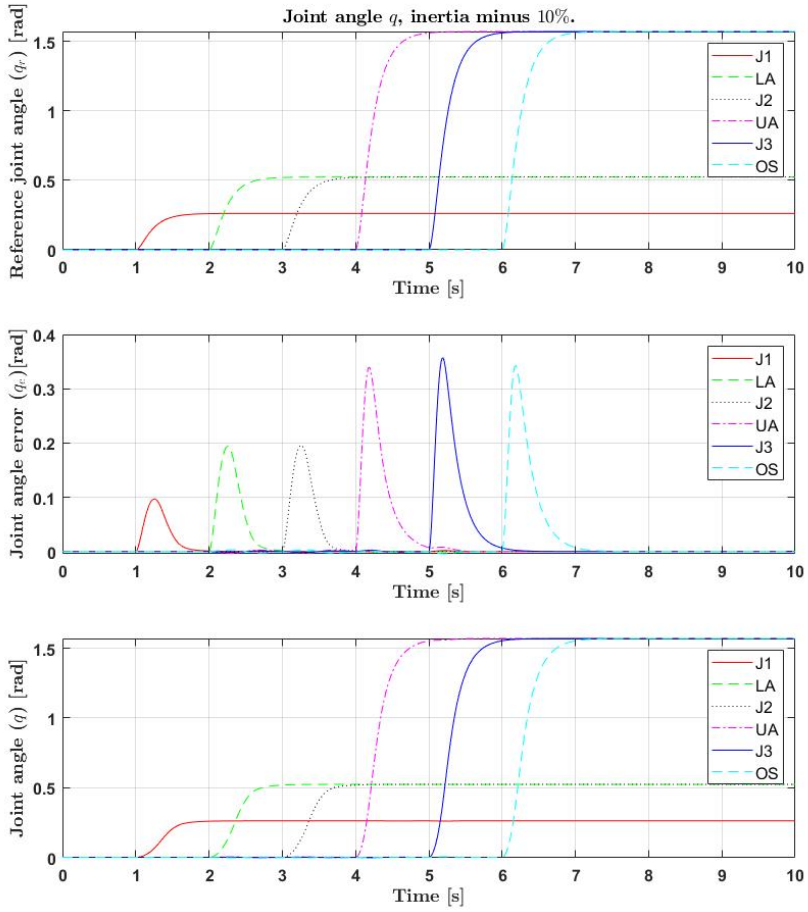


Figure 5.27: Inertia test: inertia tensor 10% lower. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

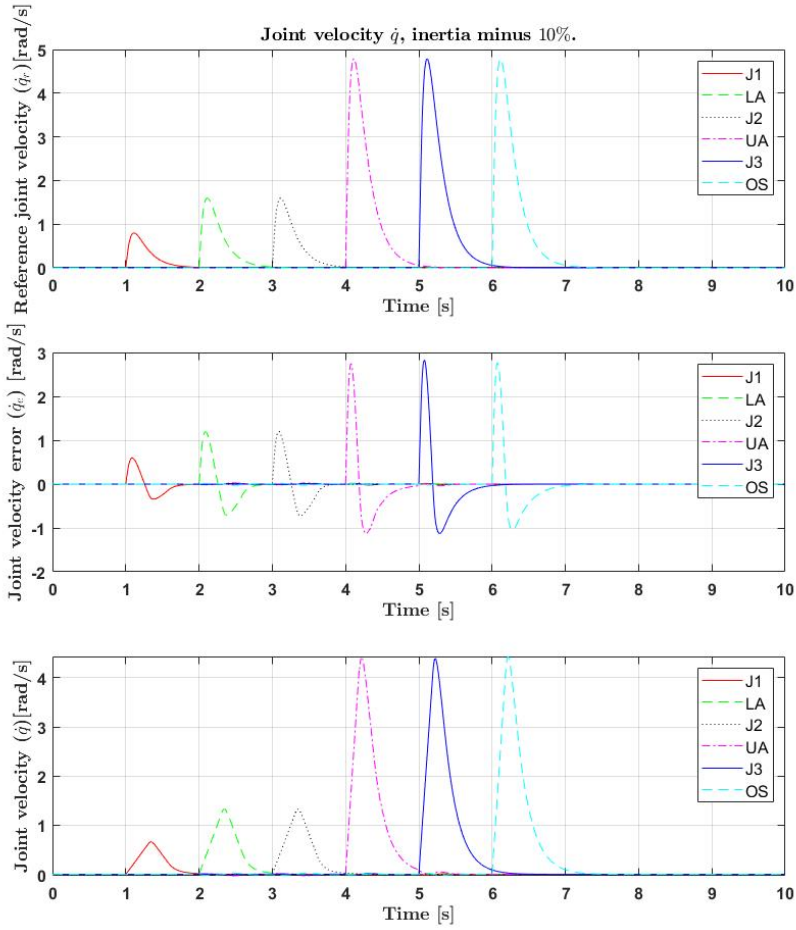


Figure 5.28: Inertia test: inertia tensor 10% lower. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

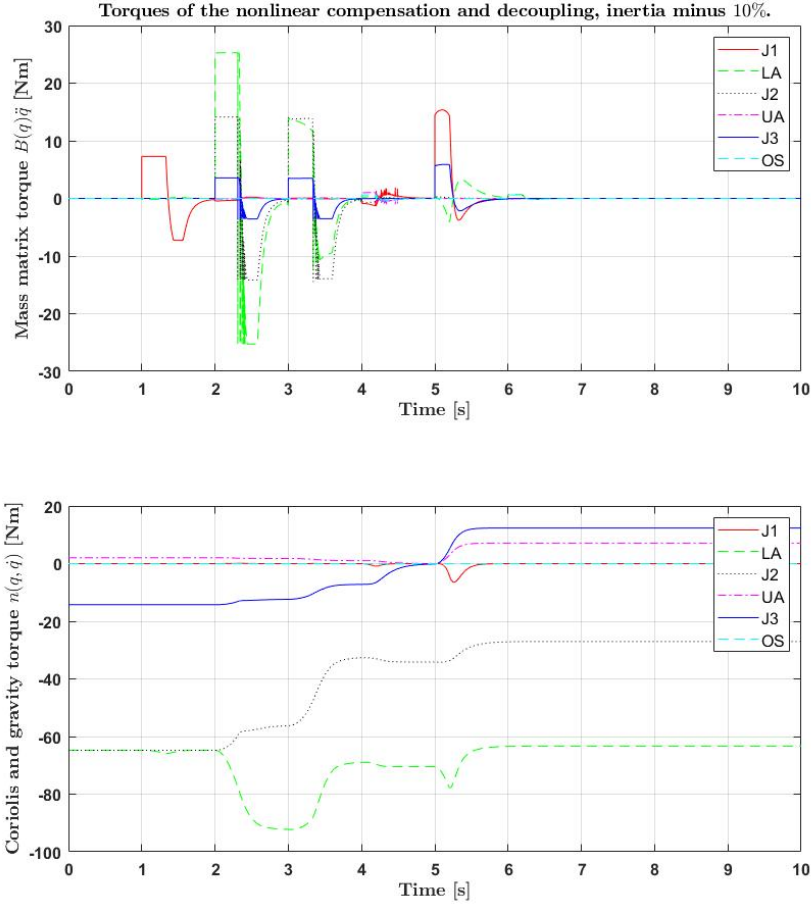


Figure 5.29: Inertia test: inertia tensor 10% lower. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

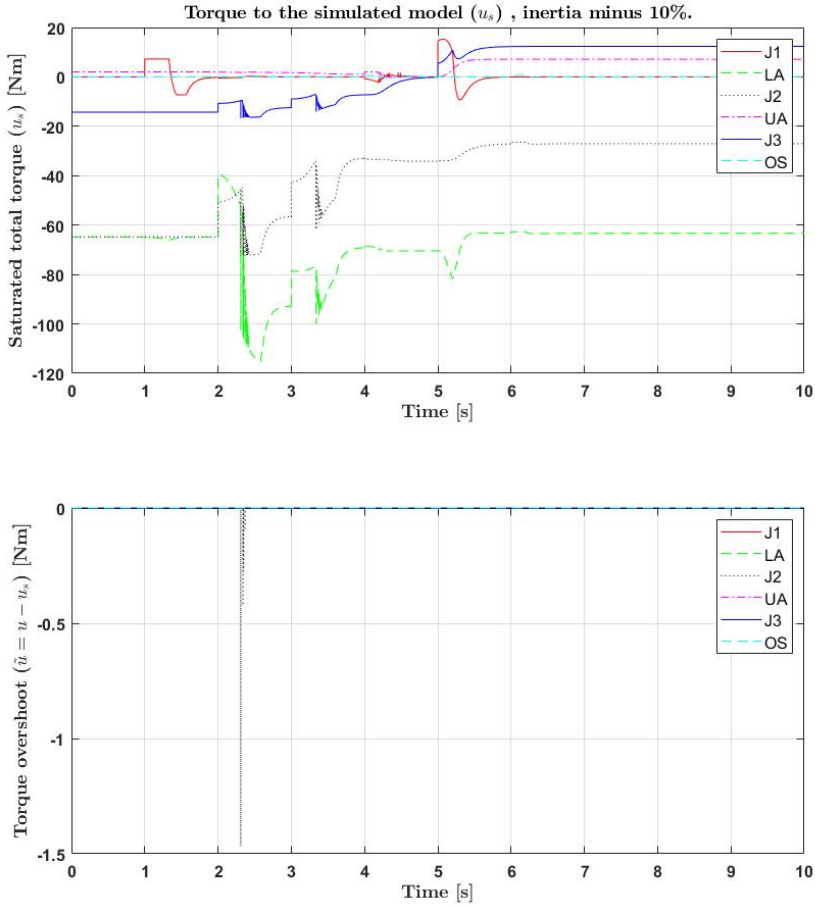


Figure 5.30: Inertia test: inertia tensor 10% lower. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

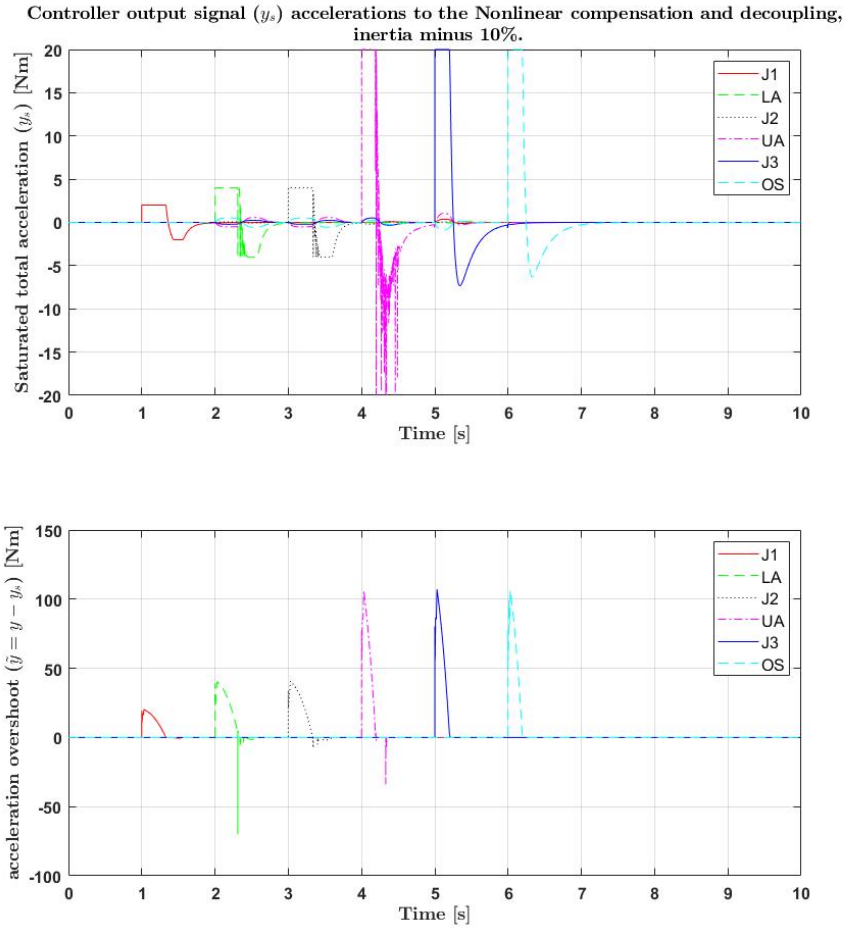


Figure 5.31: Inertia test: inertia tensor 10% lower. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

This test shows that the error in the inertia has a minimal affect on the output joint angles, only the error max amplitude change with a small margin. The difference is in the maximum error amplitude, it gets lower the lower the inertia value is and vice versa. There are also some minimal oscillations on the joint that does not make the step, which means the nonlinear compensation and decoupling get affected. On the other hand, the torque gets unstable and get some occasional peaks. The same goes for the acceleration.

5.4 Operational Space Time Path

In this test, a trajectory was created in the operational space, in the shape of a half moon as shown in Figure 5.32 and 5.33. This trajectory was given to a *Inverse Kinematics*-block from the RSTB. This block calculate the joint angle q from the given operational space position. Thereafter, these joint angles are given to the controller through a filter, $H(s)$ Equation 5.1, as a reference joint angle q_r , as well as its derivative \dot{q}_r and the second derivative \ddot{q}_r . The outcome of this test is shown in the graphs below:

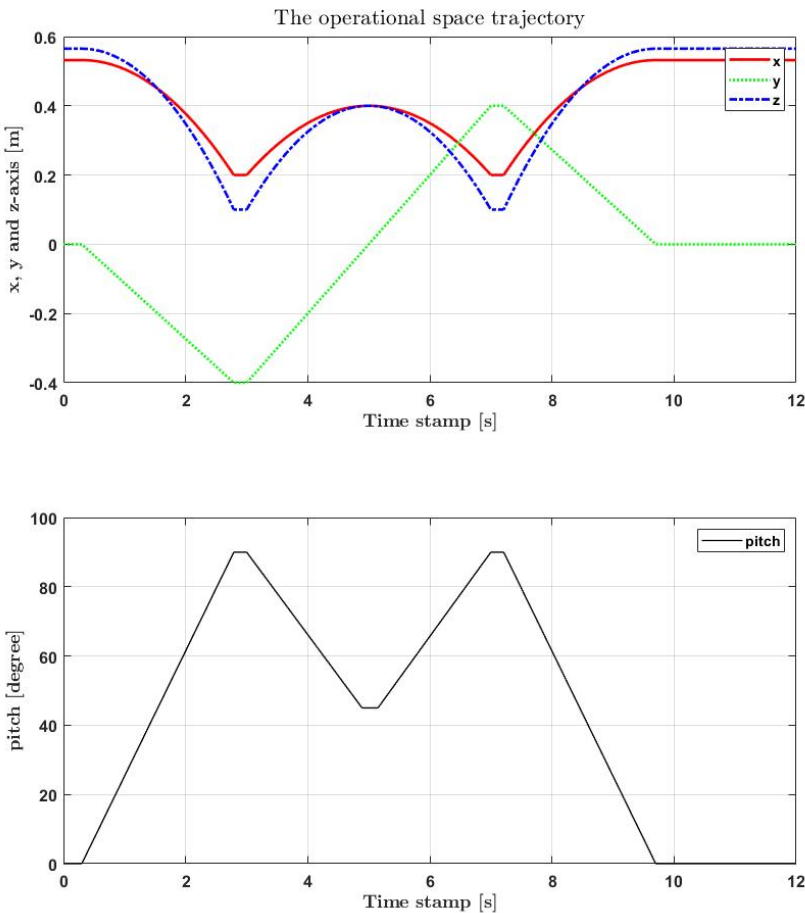


Figure 5.32: Trajectory test: the time reference signal in the operational space.

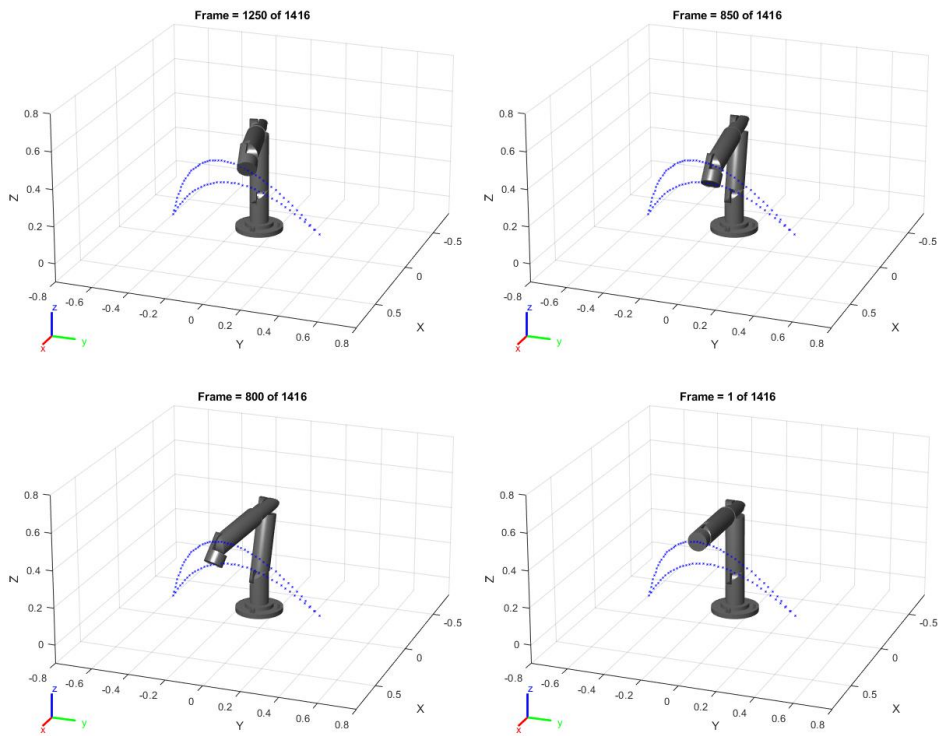


Figure 5.33: Trajectory test: 3D representation of the given trajectory and the robot, in four different frames.

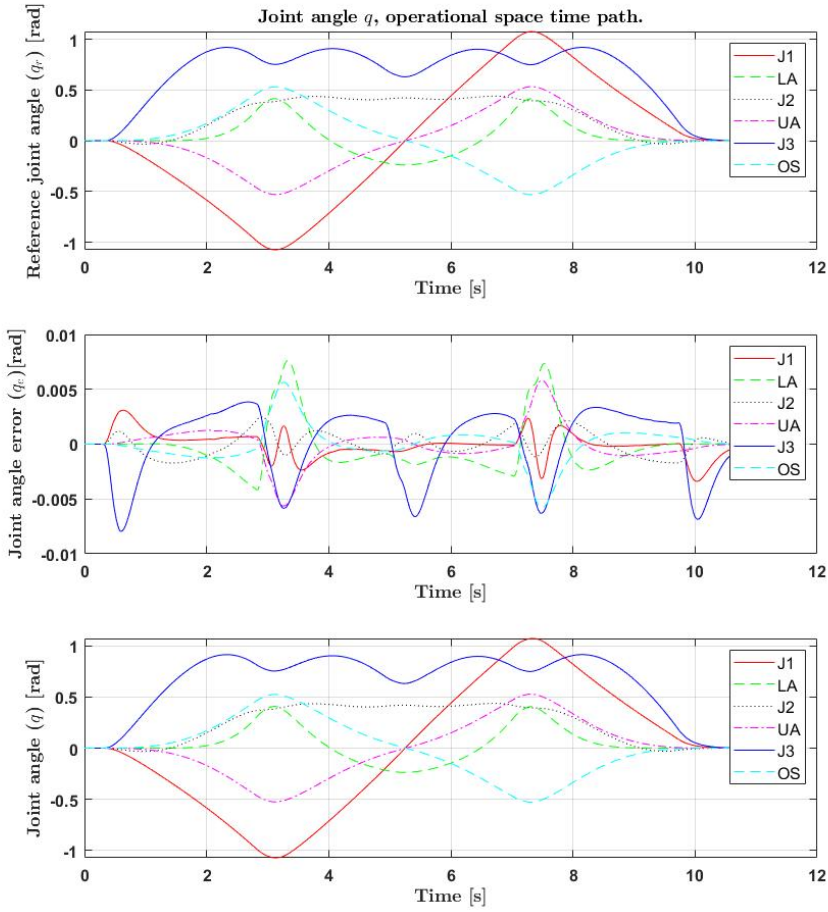


Figure 5.34: Operational space trajectory test. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

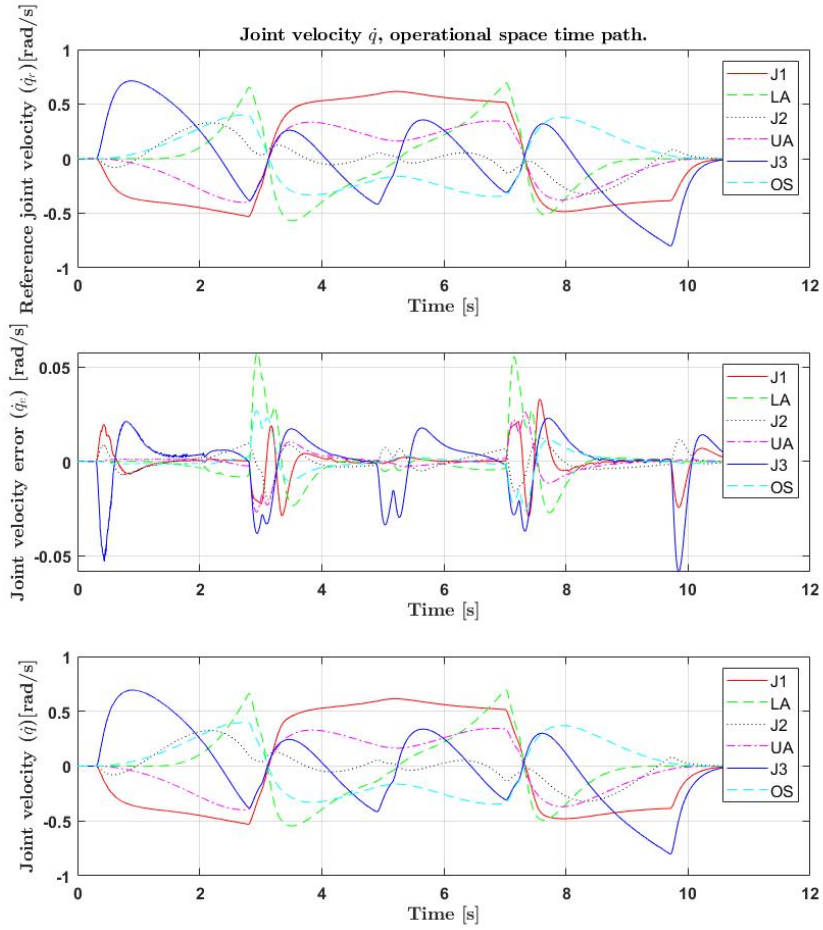


Figure 5.35: Operational space trajectory test. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

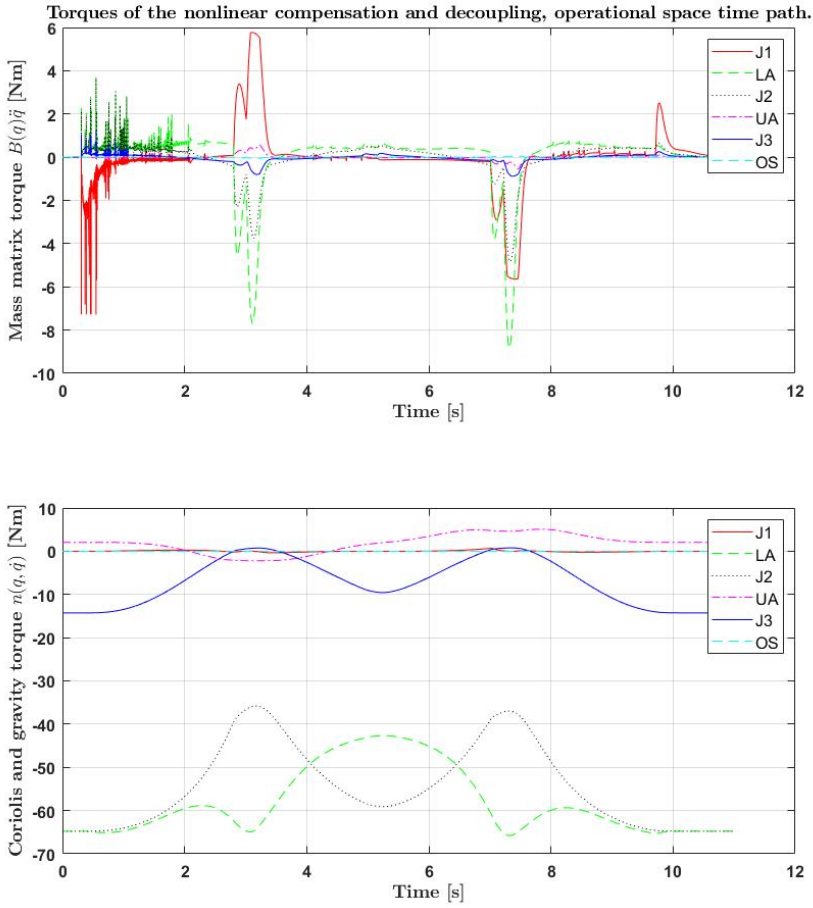


Figure 5.36: Operational space trajectory test. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

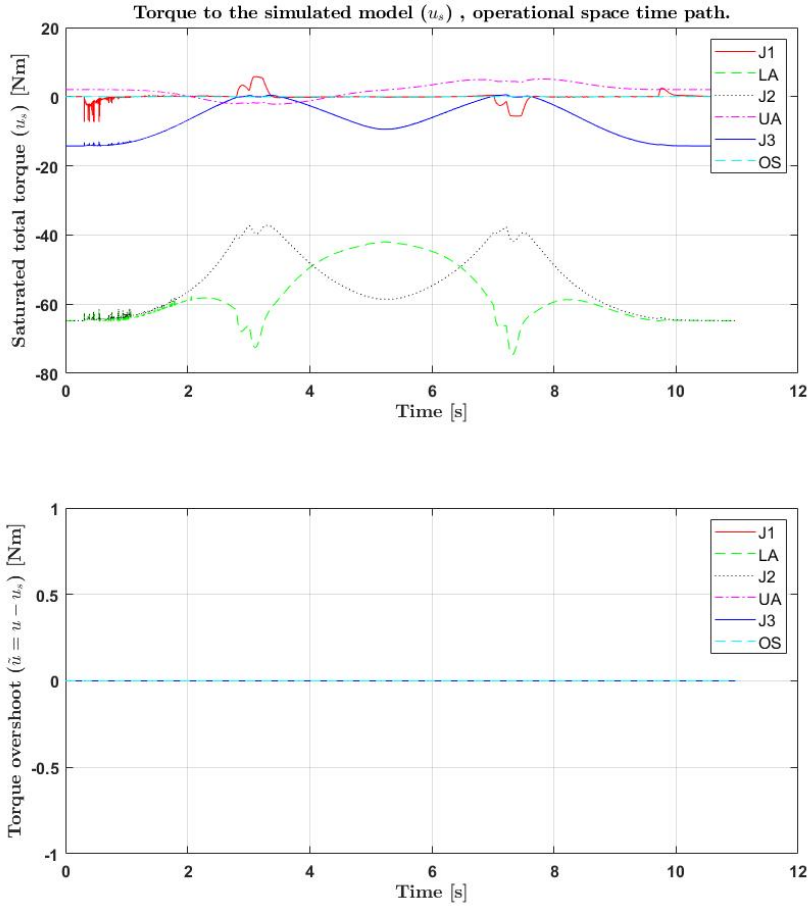


Figure 5.37: Operational space trajectory test. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

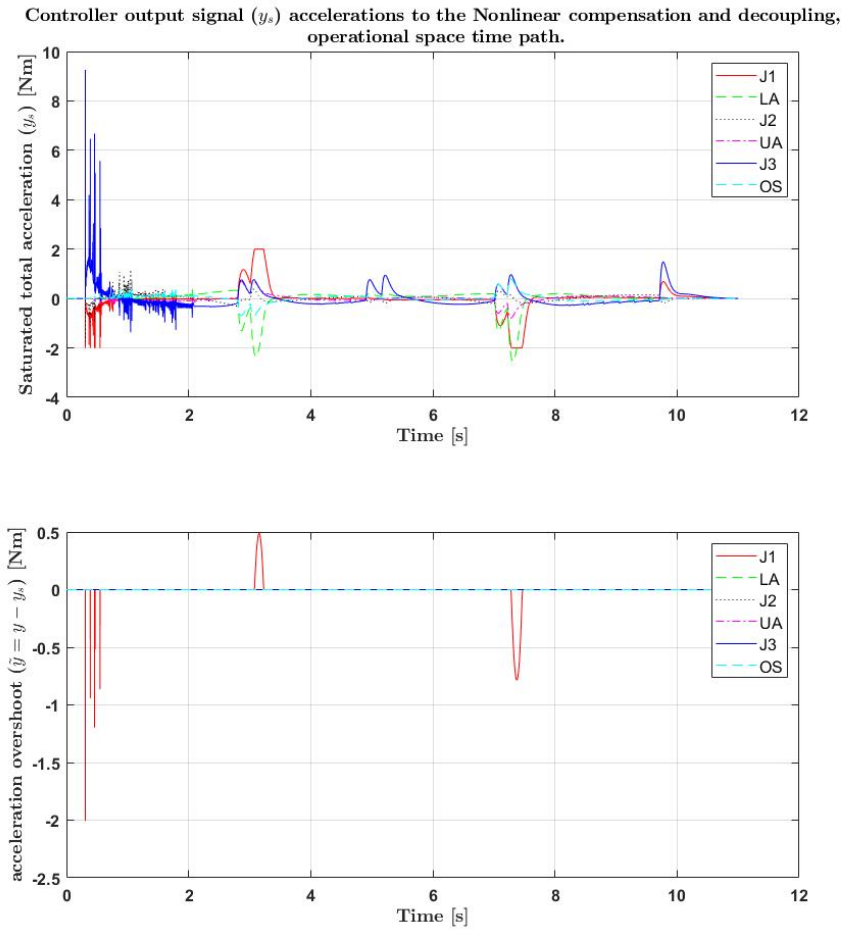


Figure 5.38: Operational space trajectory test. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

The result of this test shows that the controller manages to keep the error below $0.01\text{rad} \approx 0.5730$ degree. The error gets bigger only in the change of the path direction which happens in the following time stamps 0.30s, 3s, 5s, 7s and 10s, which is understandable. The torque graph shows that some peaks and rapid oscillations happens in the first 2 seconds, see Figure 5.36. Although, these changes do not result in a big amplitude, which is even lower than the amplitude of the ideal case. These peaks also appear in the acceleration graph.

To conclude this test, the results are promising and the controller shows a good response time and it is also able to follow the reference signal with a low error amplitude.

6

Result

This chapter conclude the results of this thesis and some ideas for the future development on this thesis.

6.1 Discussion

To conclude the result, a dynamical model of the robot as well as a joint space inverse dynamic controller have been developed. The control model as well as the simulated model of the robot was developed with the help of the *RSTB Robotic System Toolbox*. To justify the approach of the development of the control and the simulated model several experiments were made. The result of the controller was promising, specially in the ideal case scenario. The controller also managed to converge the output signal to the reference signal with a minor error while encountering a noise disturbance. The error which is influenced by the noise is unavoidable, this is due to the fact that there will always be distortions in the measured signals. The only solution is to add a filter to suppress the amplitude of the noise. Operational space path following experiment indicates that the error is mainly related to the step size. With that said, a softer trajectory can keep the error to the minimum. Although, more experiments are needed to be made, such as a combination of the noise and the other experiment also testing other maneuvers. Some of the combination experiments were made but were chosen not to be presented, that is mainly for the limited size of the report and simplicity for the reader.

However, the results only show the behavior of a simulated model, which does not guarantee the same results for the real robot. Therefore, more tests are needed to be done on the real robot to confirm the results of the controller as well as the simulated model.

6.2 Future Development

The most important thing to develop in the future is to add a filter on the measured signals, due to the impact that the noise has on the oscillation in the torque. These rapid changes in the torque may shorten the lifetime of the material used in the robot, it also requires more power (ampere). Another thing to keep in mind, is to add a robustness controller. That is to minimize the impact of the inaccuracy of the properties given to the controller, see Figure 6.1.

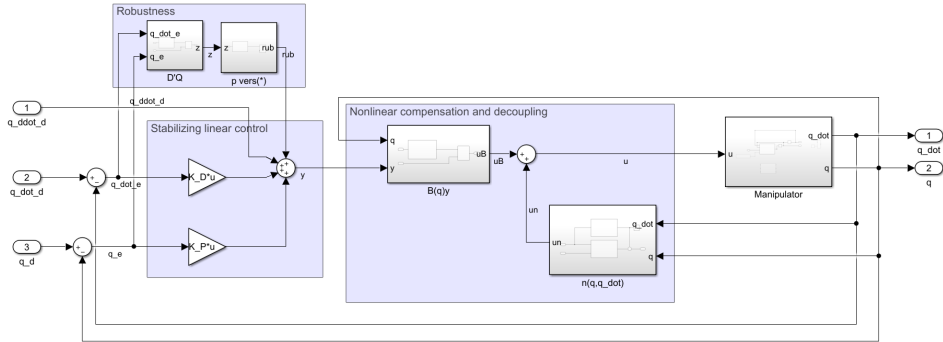


Figure 6.1: Robustness control, for the future development.

Appendix

A

RTB Definition of The Links Chain

```
1  n = 6; % number of links (rigid-body)
2  L (1:n) = Link();
3  for i = 1:n
4      L(i) = Revolute('d', d(i), ...           % link length ...
                    (Dennavit-Hartenberg notation)
5      'a', a(i), ...                           % link offset ...
                    (Dennavit-Hartenberg notation)
6      'alpha', alpha(i), ...                   % link twist ...
                    (Dennavit-Hartenberg notation)
7      'I', Body(i+1).I, ...                     % inertia tensor of ...
                    link with respect to center of mass I = [L_xx, L_yy, ...
                    L_zz, L_xy, L_yz, L_xz]
8      'r', Body(i).COG, ...                     % distance of i:th ...
                    origin to center of mass [x,y,z] in link reference frame
9      'm', Body(i).m, ...                       % mass of link
10     'Jm', 0, ...                             % actuator inertia
11     'G', 0, ...                             % gear ratio
12     'B', 0, ...                             % actuator viscous ...
                    friction coefficient (measured at the motor)
13     'Tc', [0 0];%, ...                       % actuator Coulomb ...
                    friction coefficient for direction [-,+] (measured at ...
                    the motor)
14 end
15 Robot_CM = SerialLink(L, 'name', 'ROBOT_CM');
```


B

***Simulink* Model of the Robot Controller Using RSTB.**

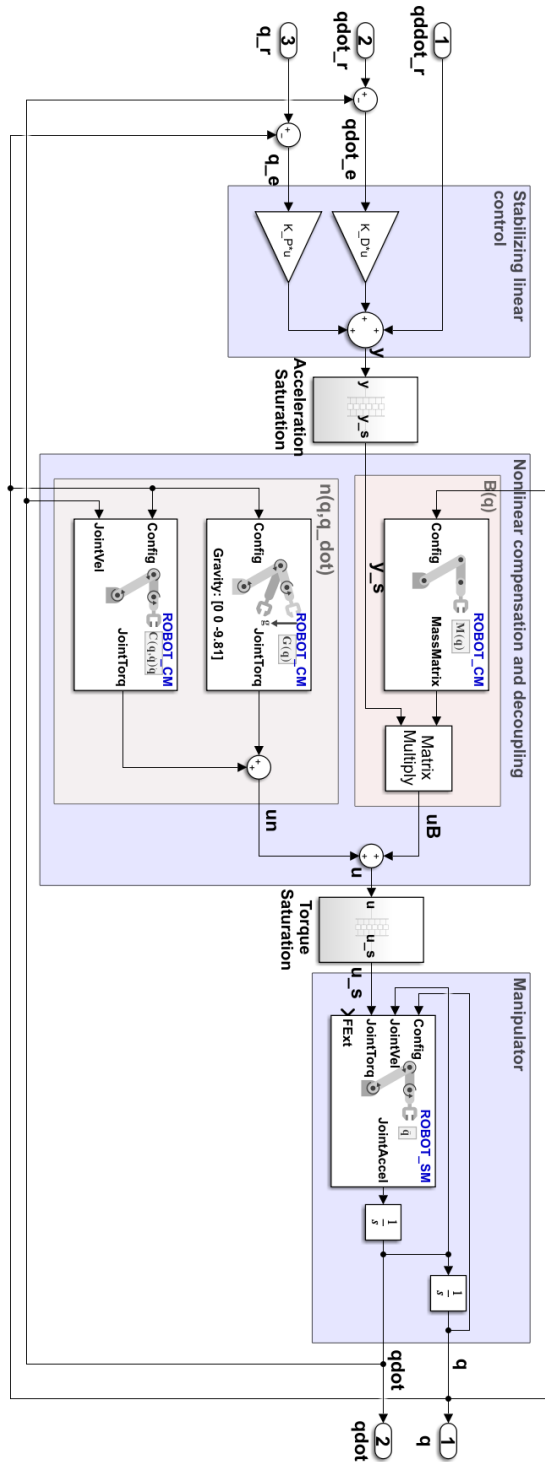


Figure B.1: Simulinkmodel of the Robot controller using RSTB.

C

RSTB Definition of The Links Chain

```
1  % joint      1 2 3 4 5 6
2  Rot_axis = [0 0 0 1 0 1; % x
3              0 1 1 0 1 0; % y
4              1 0 0 0 0 0]; % z
5  Transform = eye(4);
6  %Rigid-body tree declaration (Joint and body names)
7  joints_name = "jnt" + (1:6);
8  Visual_name = "3D/" + ["Base.stl", "Joint1.stl", "LowerArm.stl", ...
9                        "Joint2.stl", "UpperArm.stl", "Joint3.stl", "OS.stl"];
10
11 ROBOT_CM = rigidBodyTree('DataFormat','row');
12 ROBOT_CM.Base.addVisual("Mesh", Visual_name(1));
13 n = 6; % number of links (rigid-body)
14 Link(1:n) = rigidBody(' ');
15 Joint(1:n) = rigidBodyJoint(' ');
16 %Rigid-body tree definitions from the Body data
17 for i = 1:n
18     Link(i) = rigidBody(Body(i+1).name);           % ...
19     Rigit-body name
20     % adding properties to the joint
21     Joint(i) = rigidBodyJoint(joints_name(i), 'revolute'); % Joint ...
22     object that defines how a rigid body moves relative to ...
23     another.
24     Joint(i).JointAxis = Rot_axis(:,i);           % ...
25     JointAxis represents the rotation axis of a revolute ...
26     joint, vector with one in the desired position ([0 0 1] ...
27     rotation in z).
28     Transform(1:3,4) = Body(i).L;                 % Body ...
29     length
30     setFixedTransform(Joint(i), Transform);        % Sets ...
31     JointToParent transform to the user-supplied 4x4 ...
32     homogeneous transform matrix T and ChildToJointTransform ...
33     to an identity matrix.
```

```

24     % adding properties to the link
25     Link(i).Joint = Joint(i);
26     Link(i).Mass = Body(i+1).m;                % Mass of the ...
           rigid body, Body mass cannot be negative. Unit: kilogram (kg).
27     Link(i).CenterOfMass = Body(i+1).COG;        % Center of ...
           mass vector of the rigid body, relative to the body frame. ...
           Unit: meter (m).
28     Link(i).Inertia = Body(i+1).I;              % Inertia of ...
           the rigid body relative to the body frame. I = [Ixx Iyy ...
           Izz Iyz Ixz Ixy] Unit: kilogram-meter-squared (kg*m^2).
29     Link(i).addVisual("Mesh", Visual_name(i+1)); % Add visual ...
           geometry data to rigid body
30     addBody(ROBOT_CM, Link(i), Body(i).name);    % Add a body to ...
           the robot
31 end
32
33 ROBOT_CM.Gravity = [0,0,-9.81];
34 clear Link Joint i n Visual_name joints_name Transform Rot_axis

```

D

Tests of the Noise Disturbance

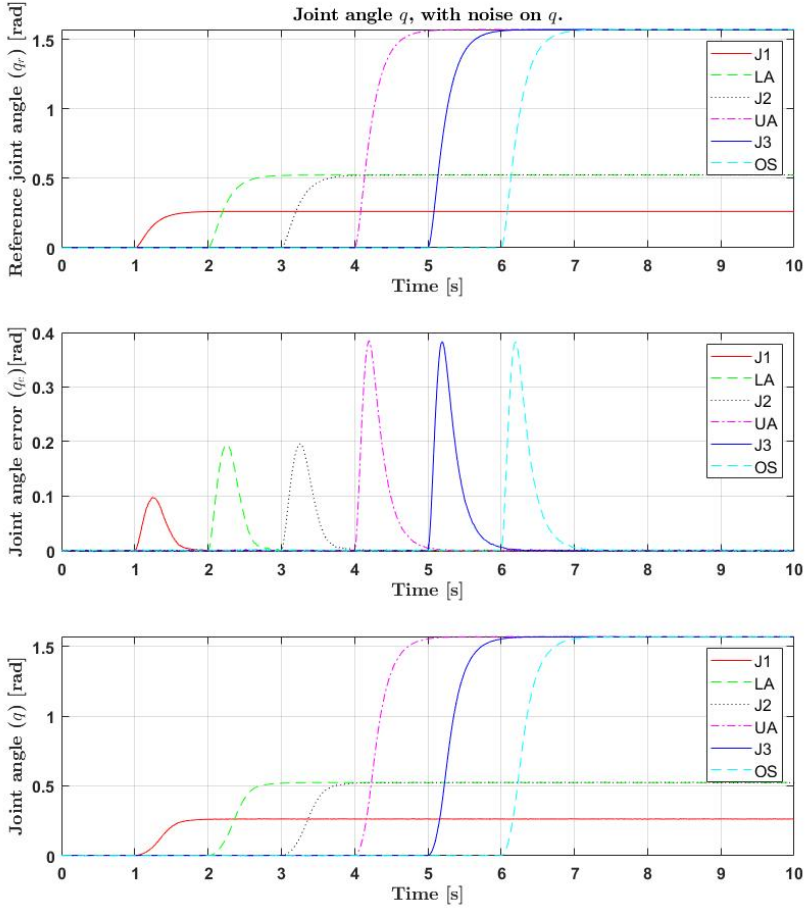


Figure D.1: Disturbance case: added noise on joint angle q . From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q .

The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

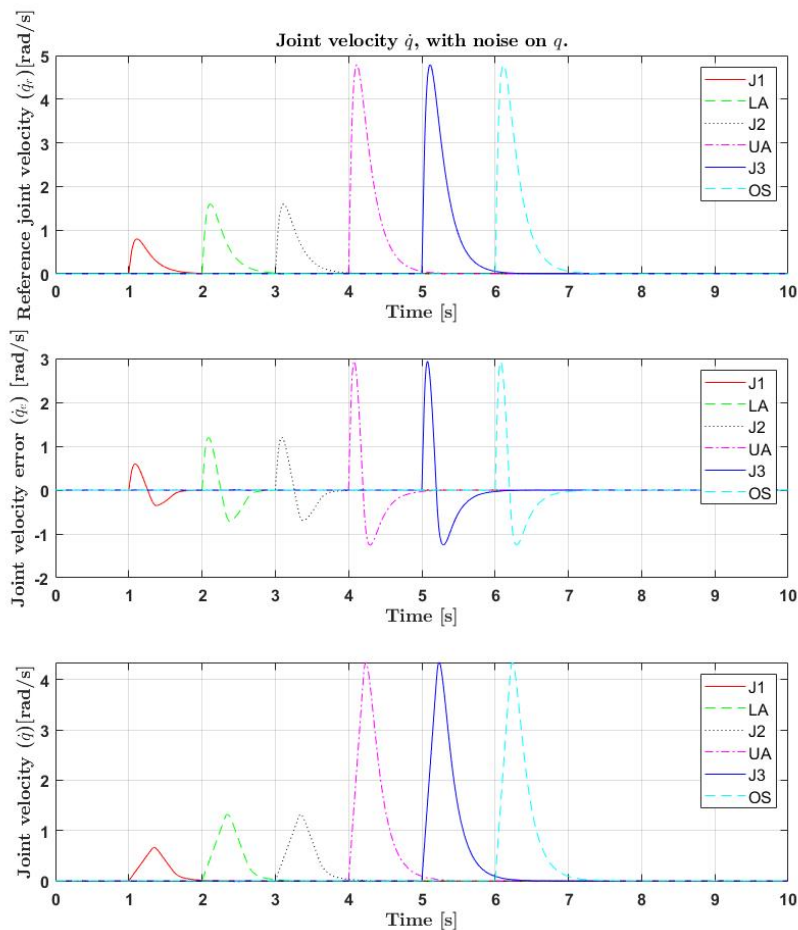


Figure D.2: Disturbance case: added noise on joint angle q . From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

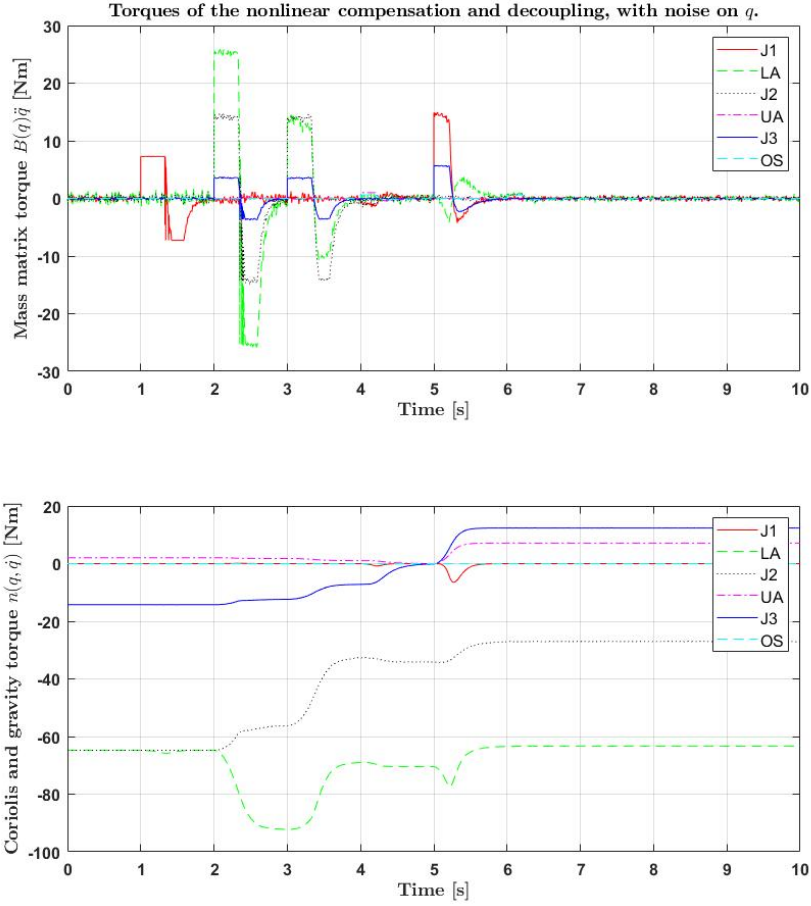


Figure D.3: Disturbance case: added noise on joint angle q . Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

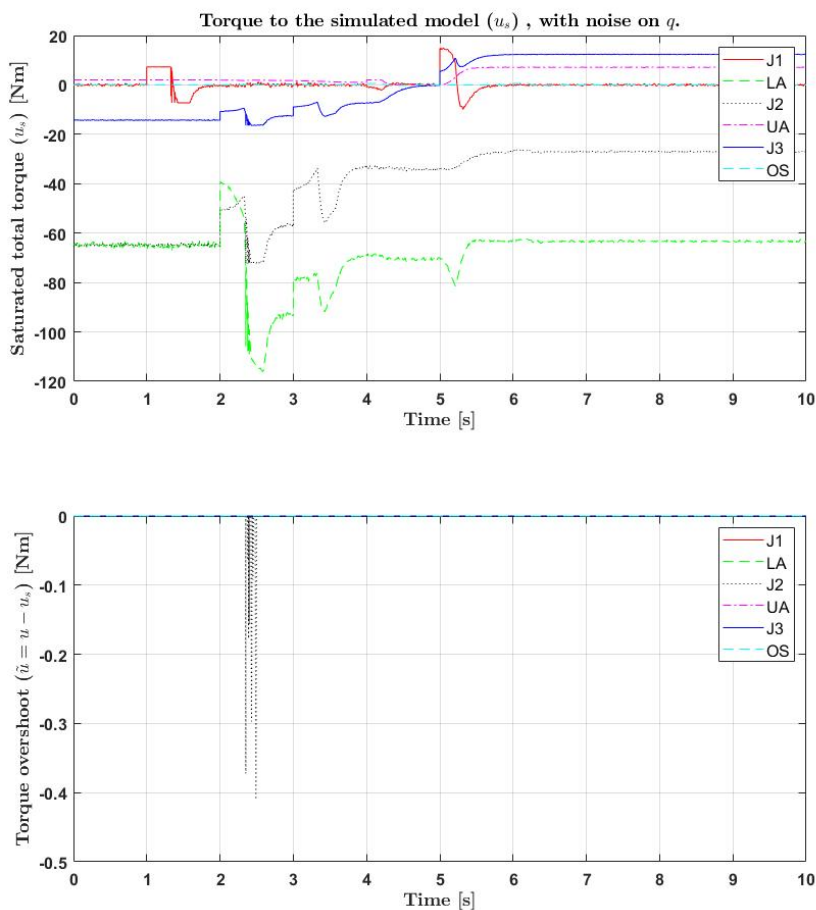


Figure D.4: Disturbance case: added noise on joint angle q . Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

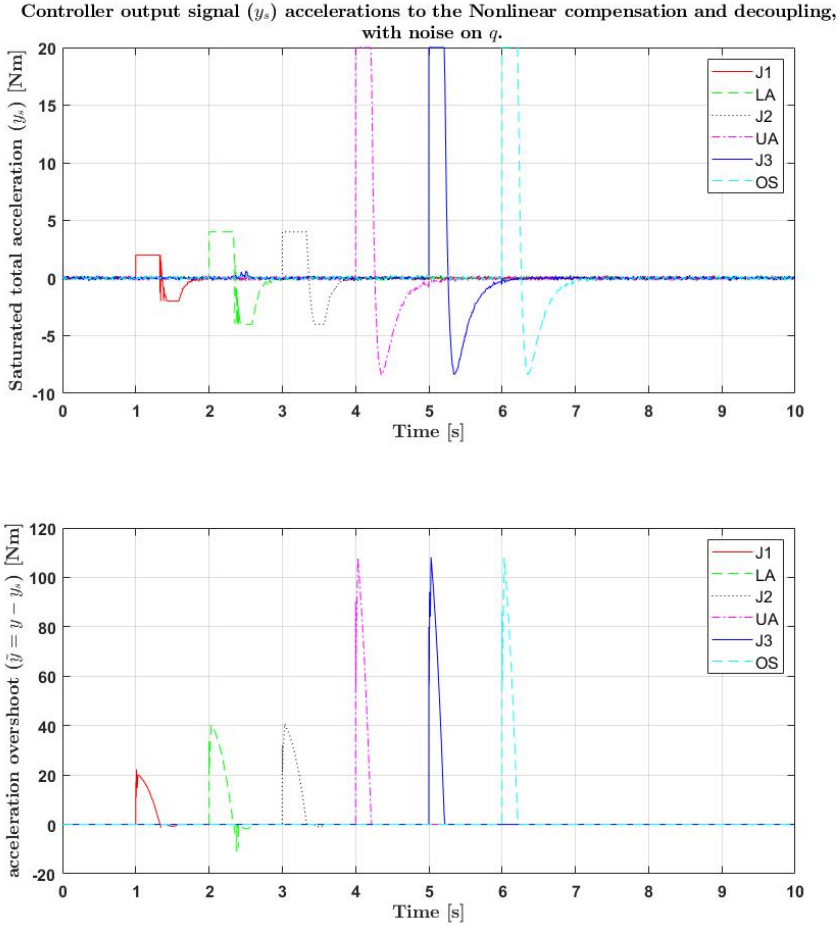


Figure D.5: Disturbance case: added noise on joint angle q . Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

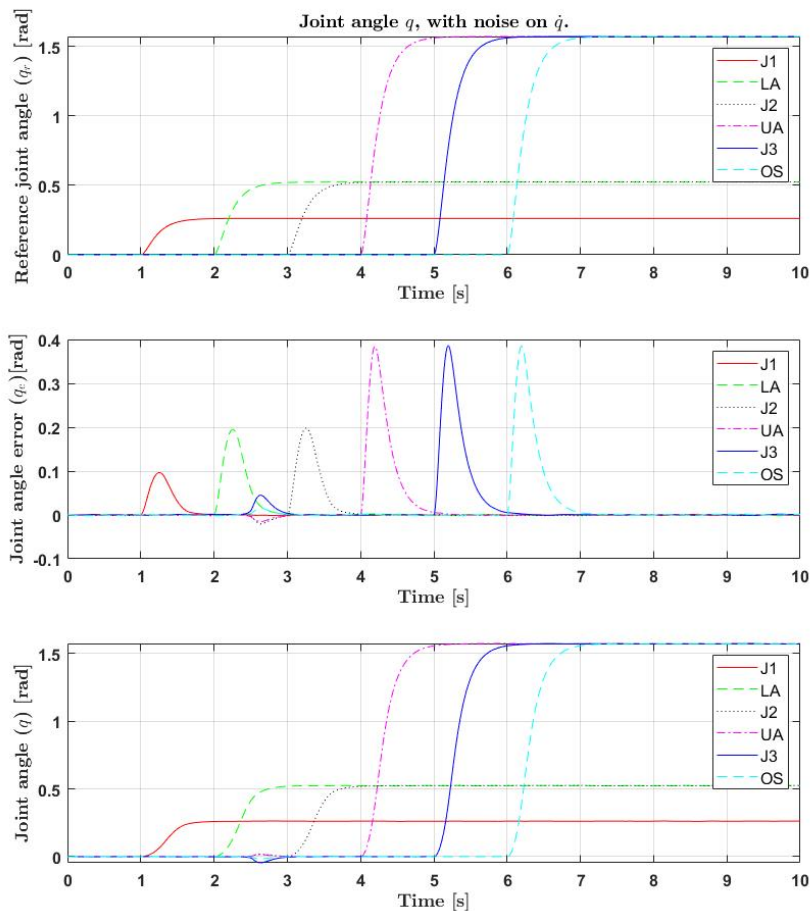


Figure D.6: Disturbance case: added noise on joint velocity \dot{q} . From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q .

The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

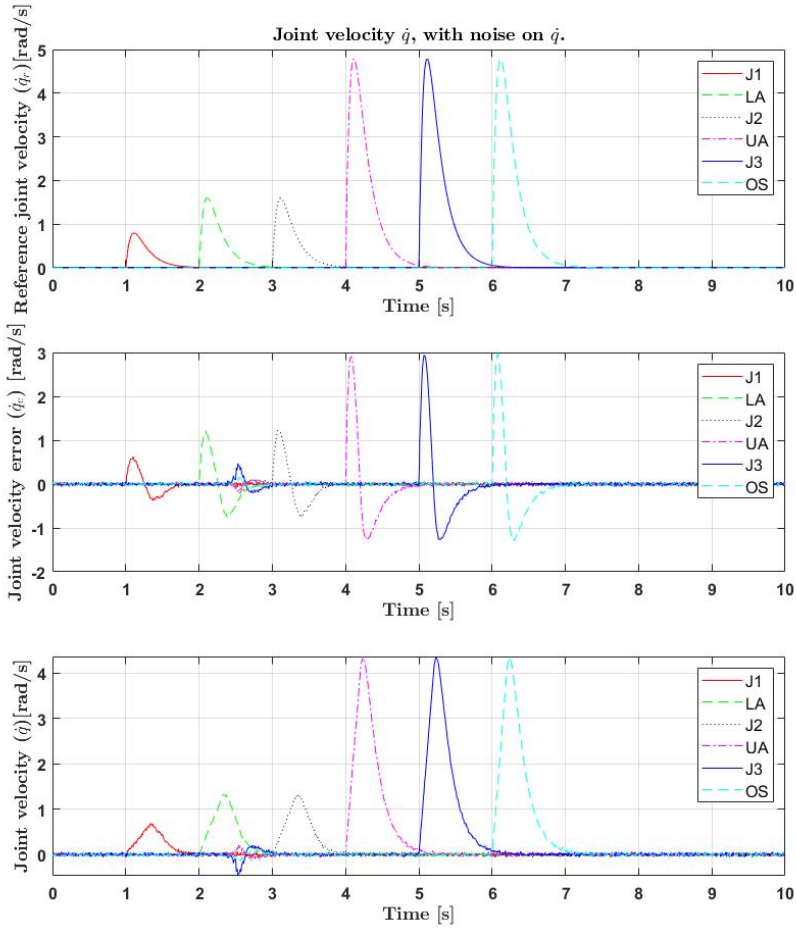


Figure D.7: Disturbance case: added noise on joint velocity \dot{q} . From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} .

The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

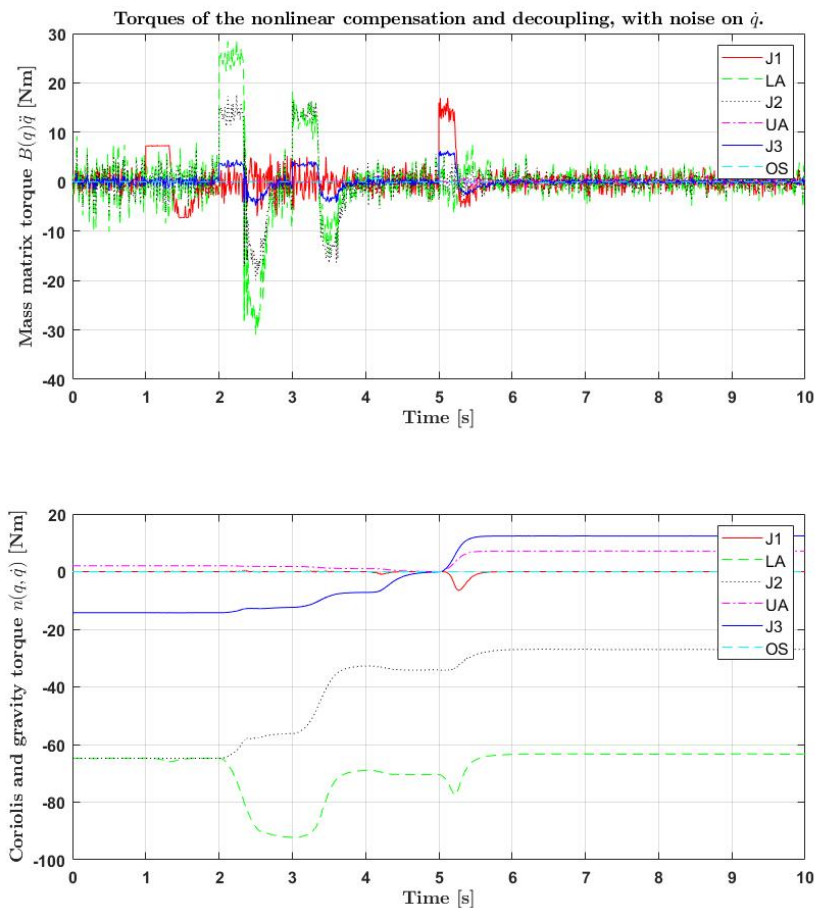


Figure D.8: Disturbance case: added noise on joint velocity \dot{q} . Top graph: torques from the $B(q)\dot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

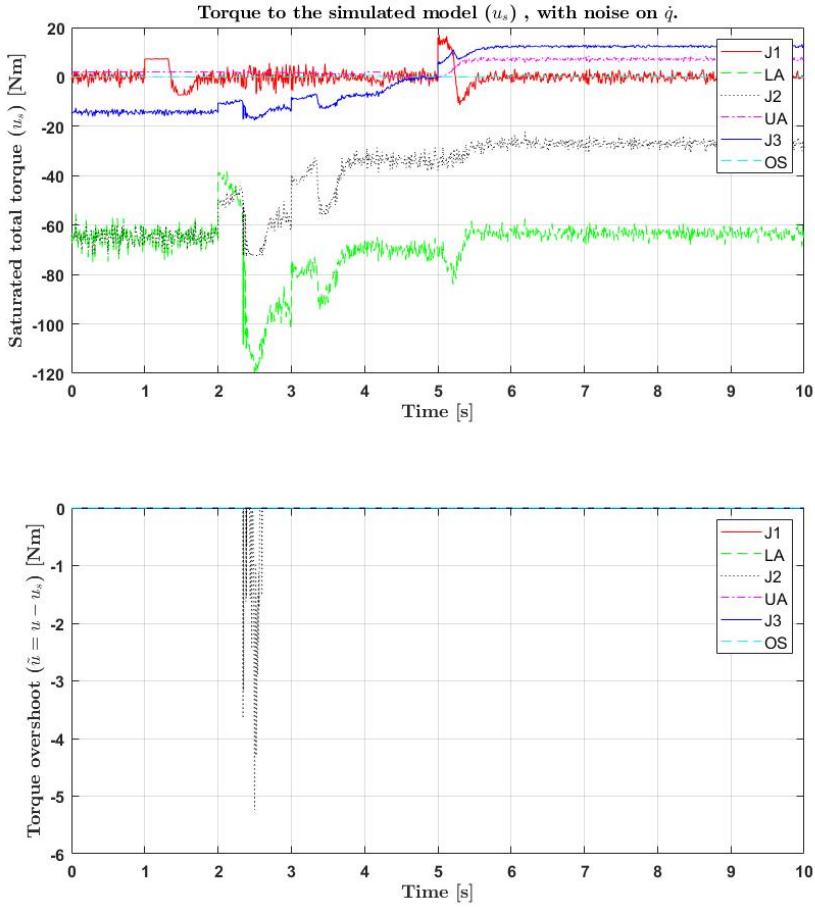


Figure D.9: Disturbance case: added noise on joint velocity \dot{q} . Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

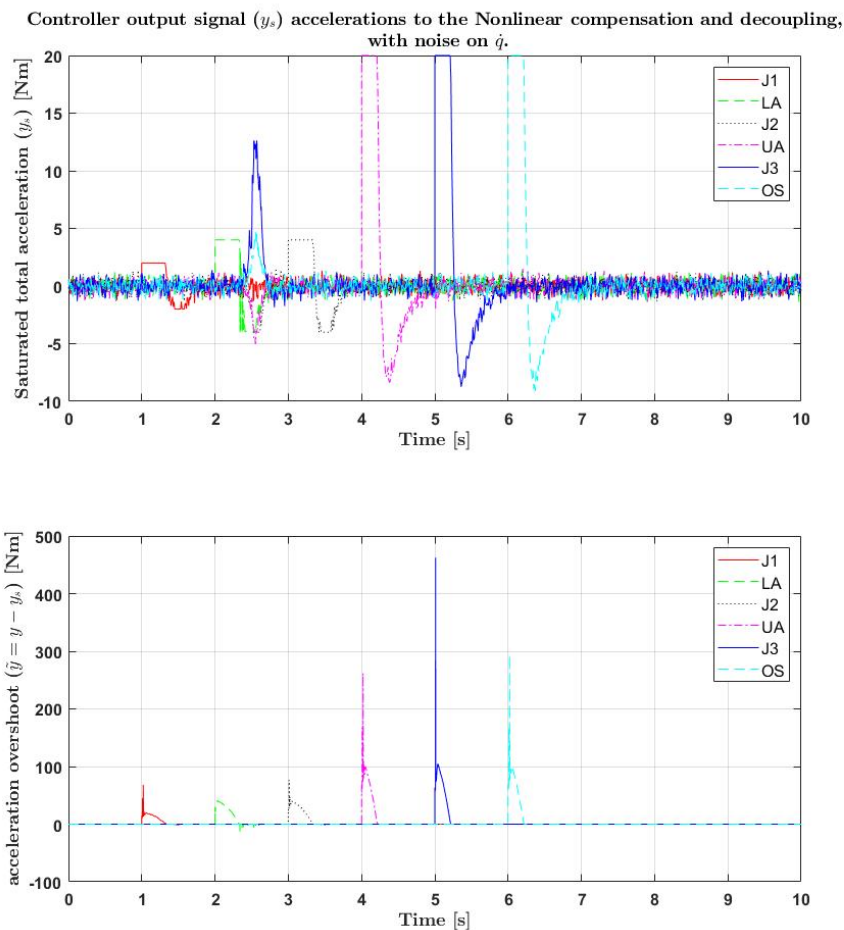


Figure D.10: Disturbance case: added noise on joint velocity \dot{q} . Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

E

Payload Tests

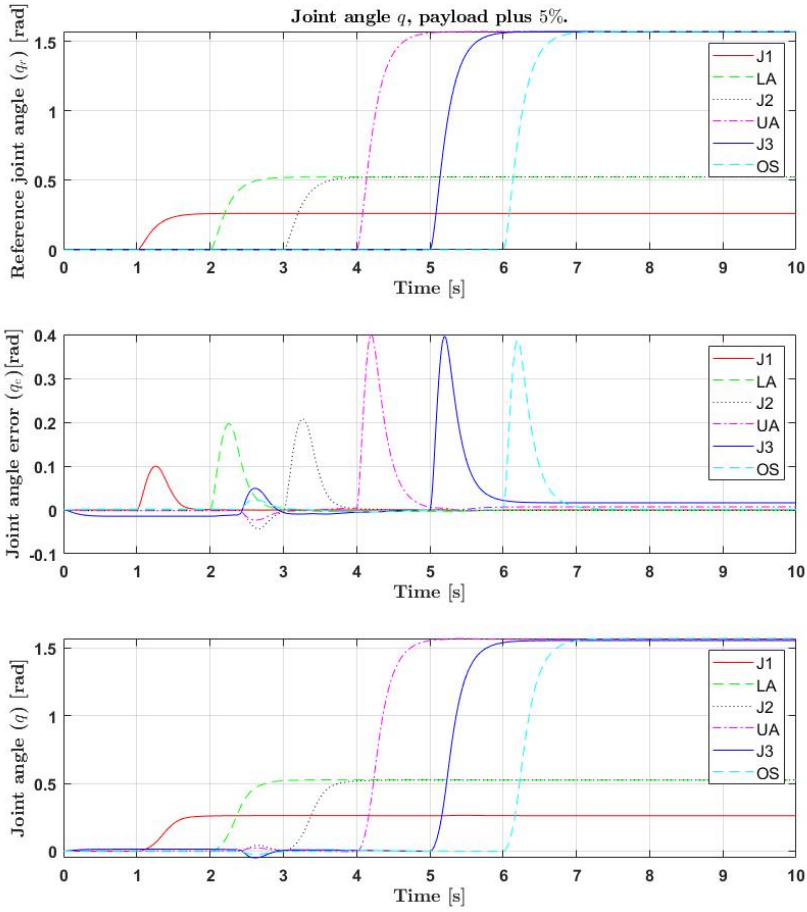


Figure E.1: Payload test: test weight 5% higher. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

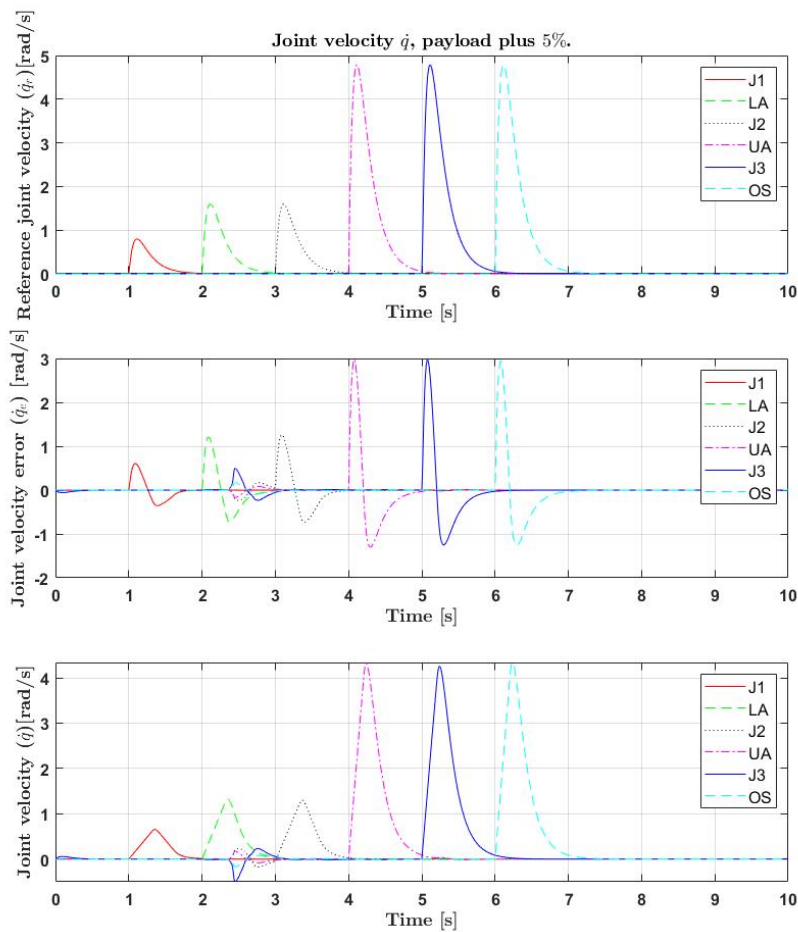


Figure E.2: Payload test: test weight 5% higher. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

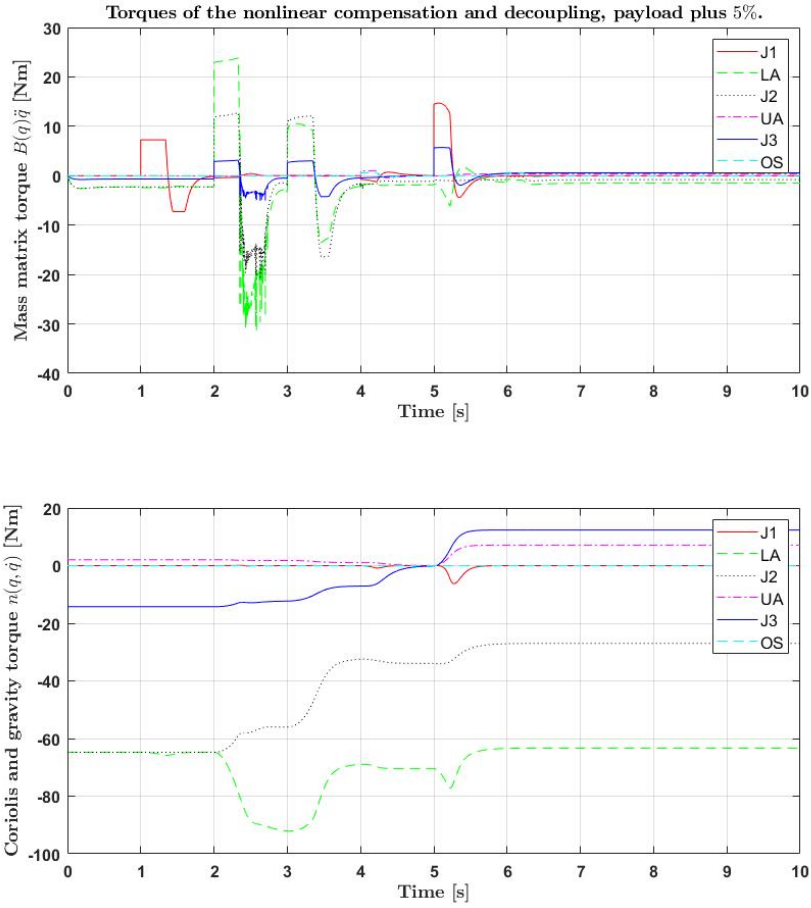


Figure E.3: Payload test: test weight 5% higher. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

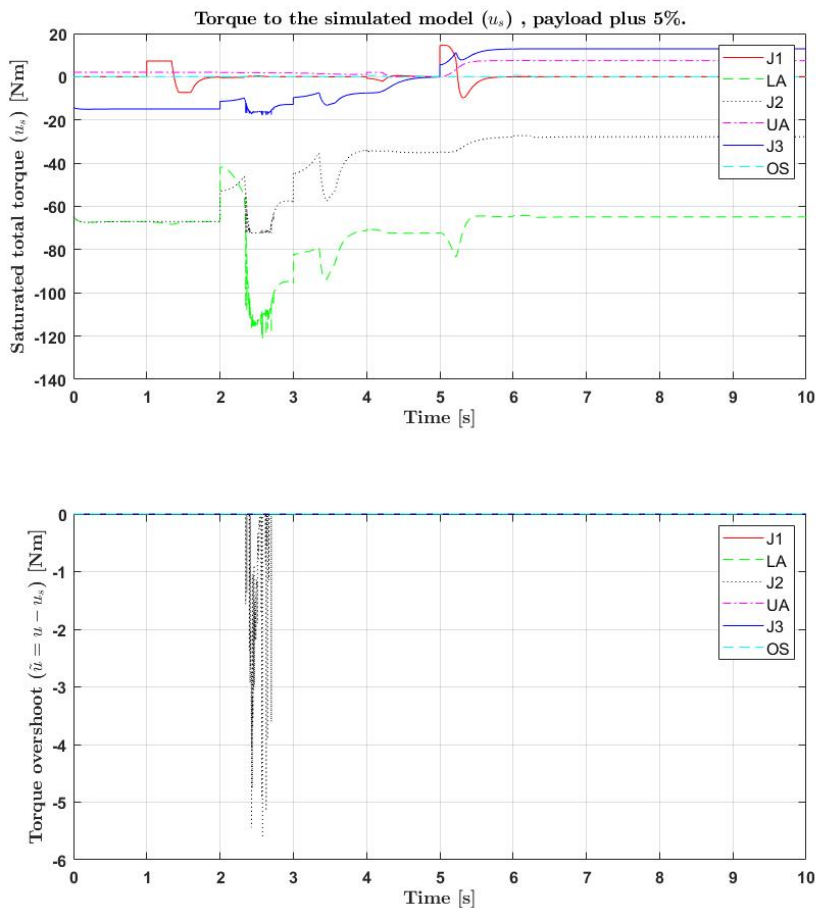


Figure E.4: Payload test: test weight 5% higher. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

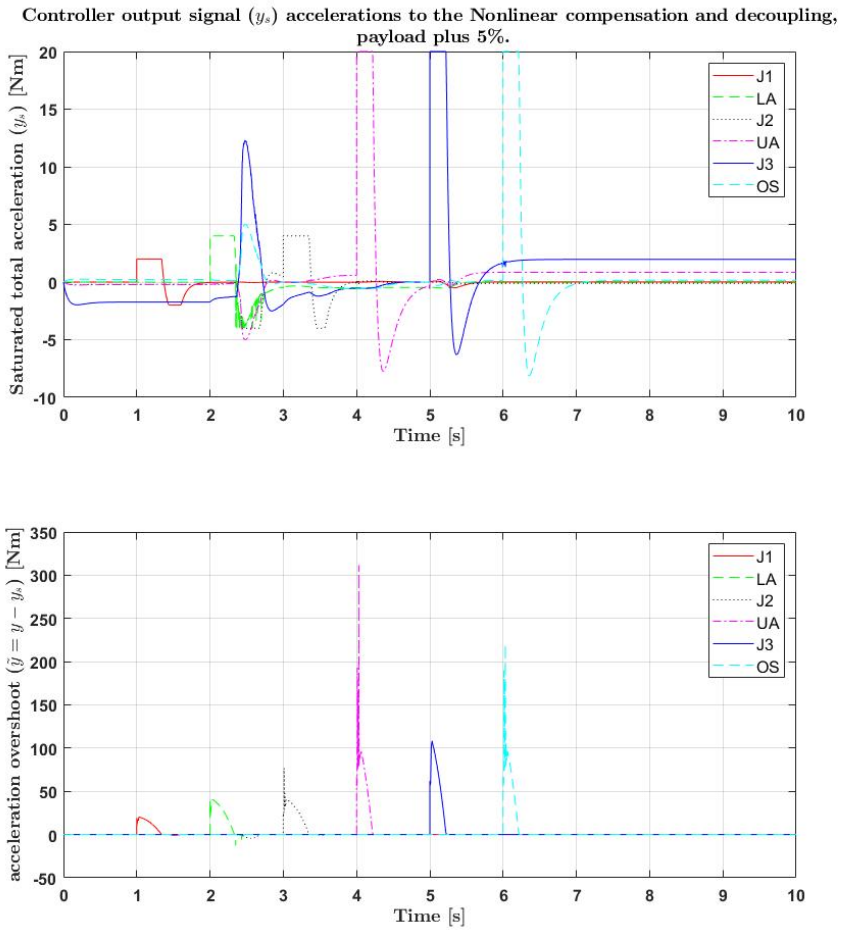


Figure E.5: Payload test: test weight 5% higher. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

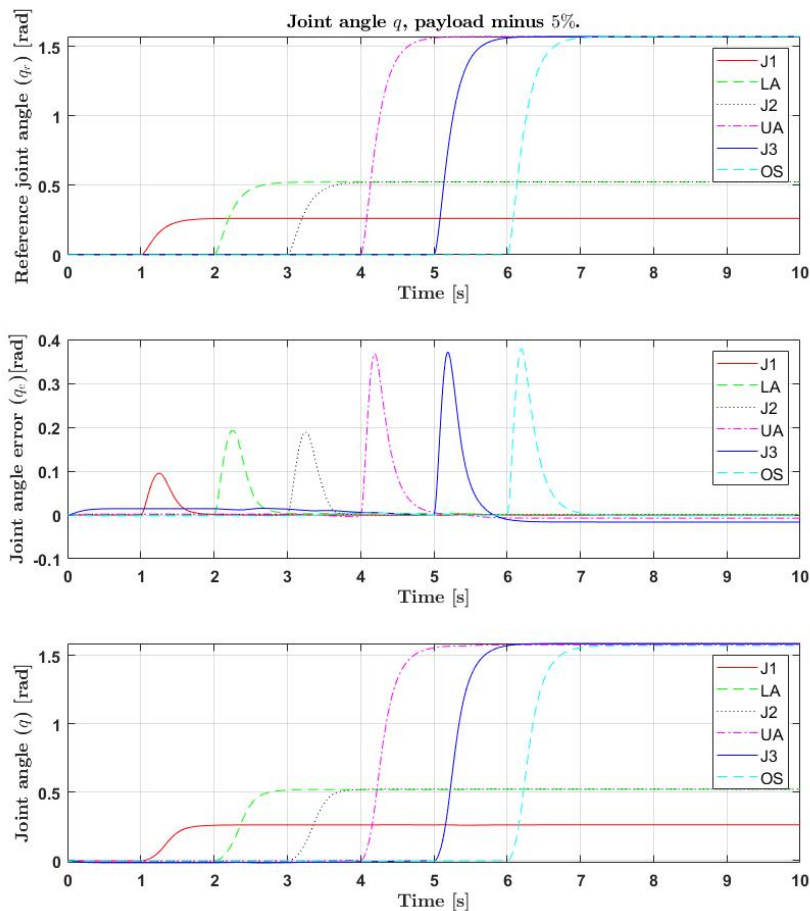


Figure E.6: Payload test: test weight 5% lower. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

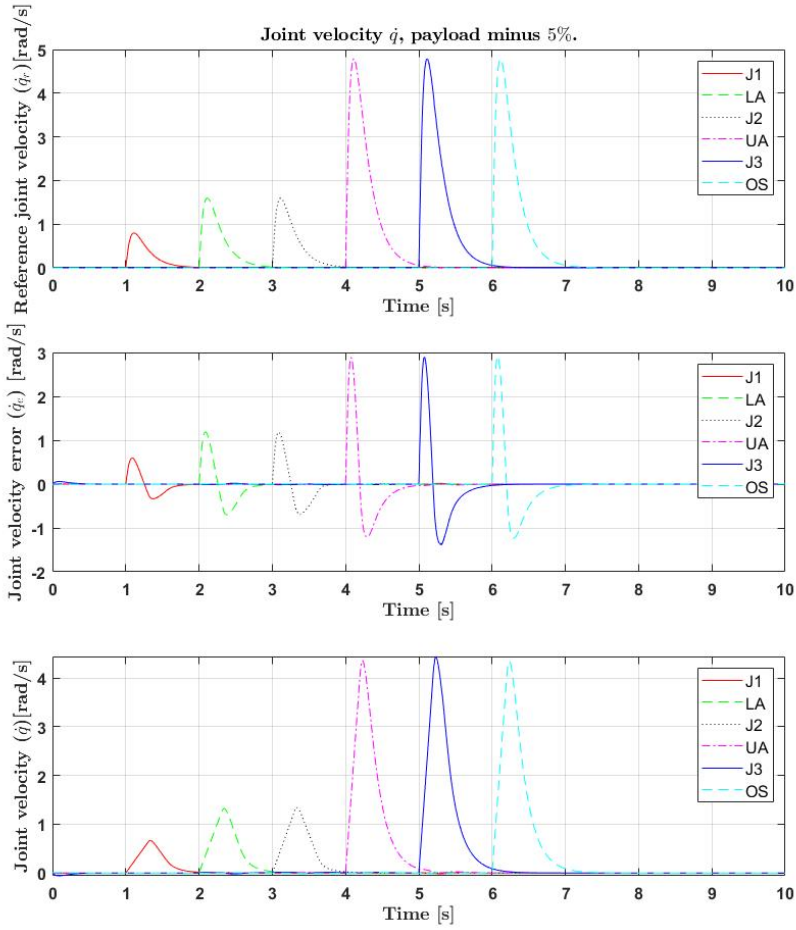


Figure E.7: Payload test: test weight 5% lower. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

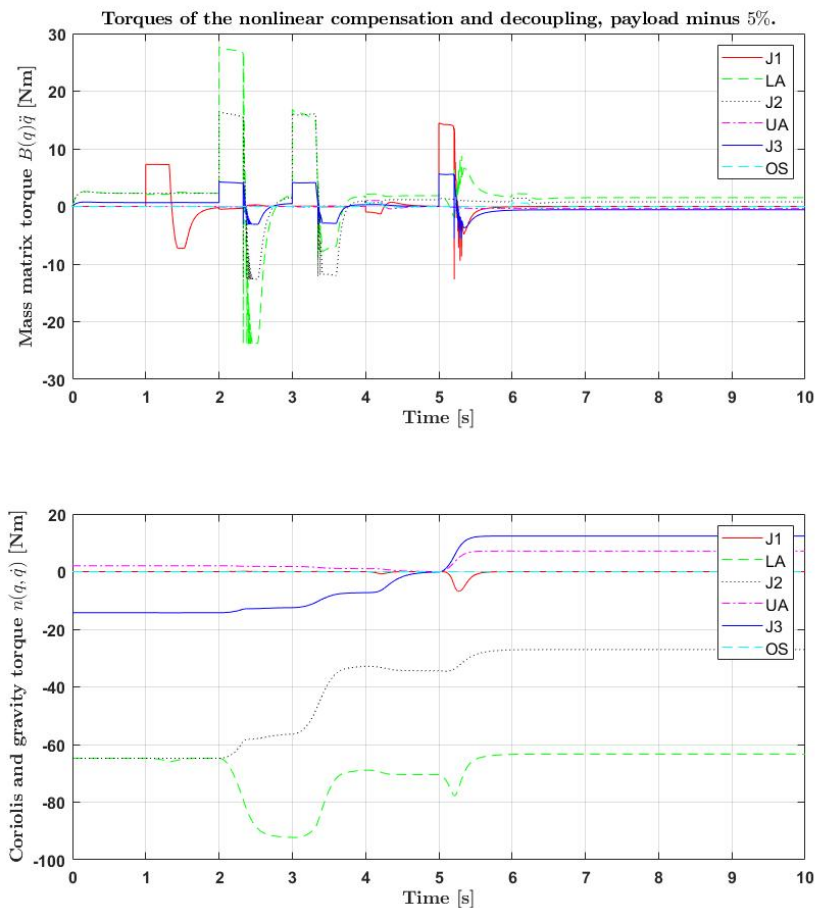


Figure E.8: Payload test: test weight 5% lower. Top graph: torques from the $B(q)\dot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

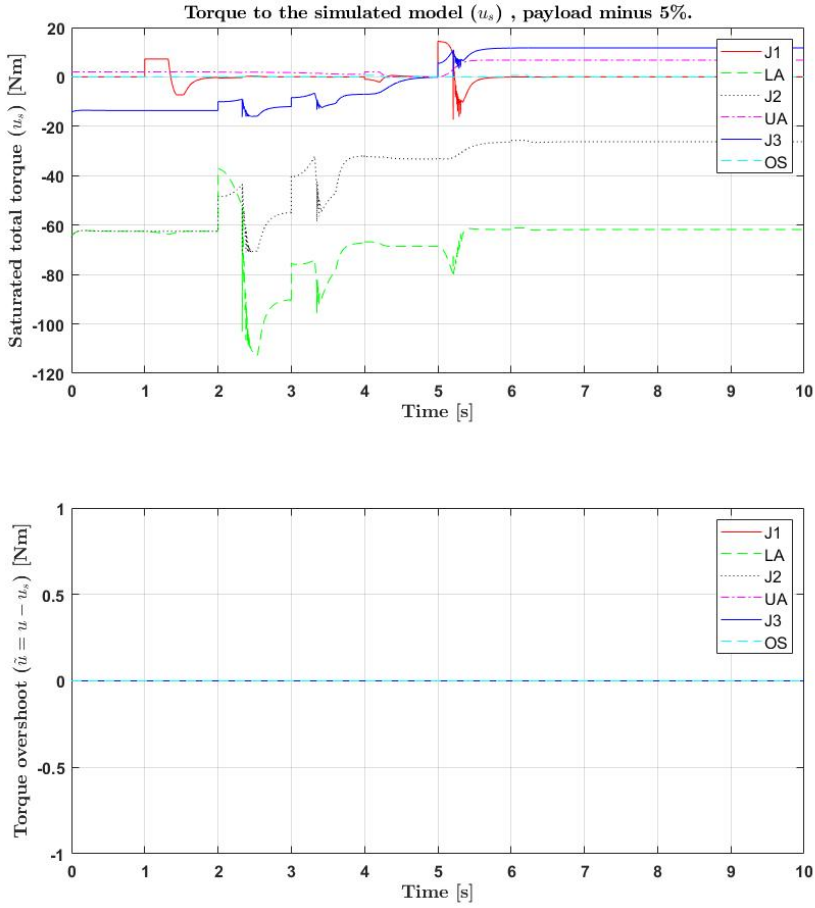


Figure E.9: Payload test: test weight 5% lower. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

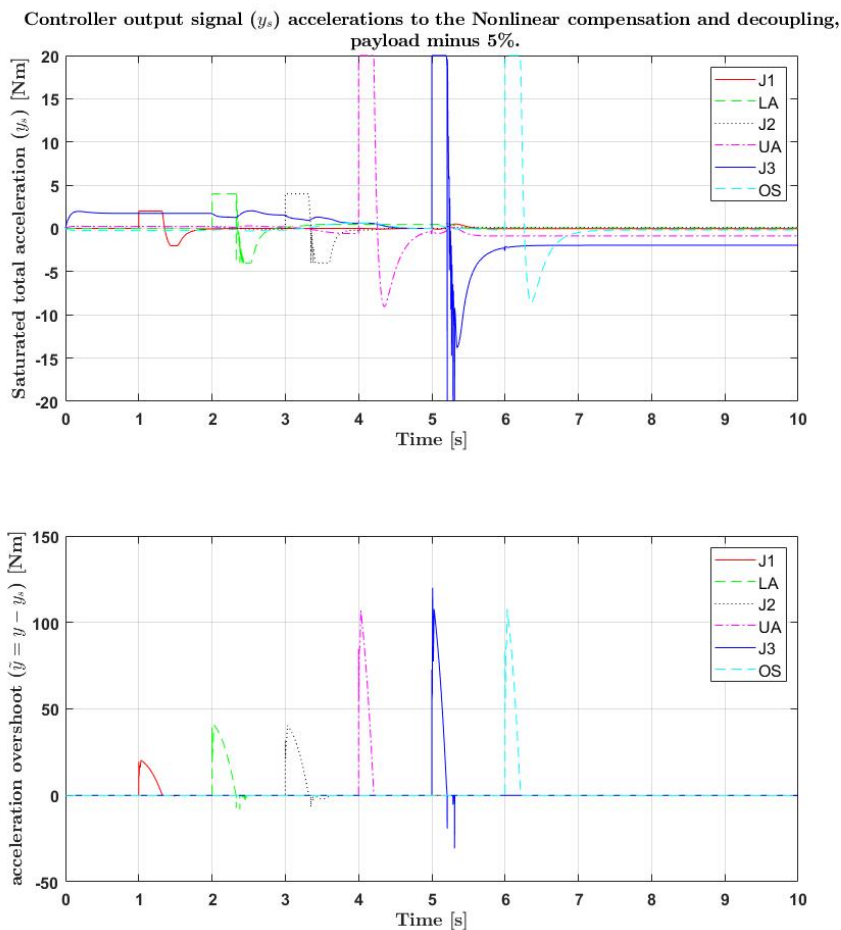


Figure E.10: Payload test: test weight 5% lower. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

F

Inertia Test

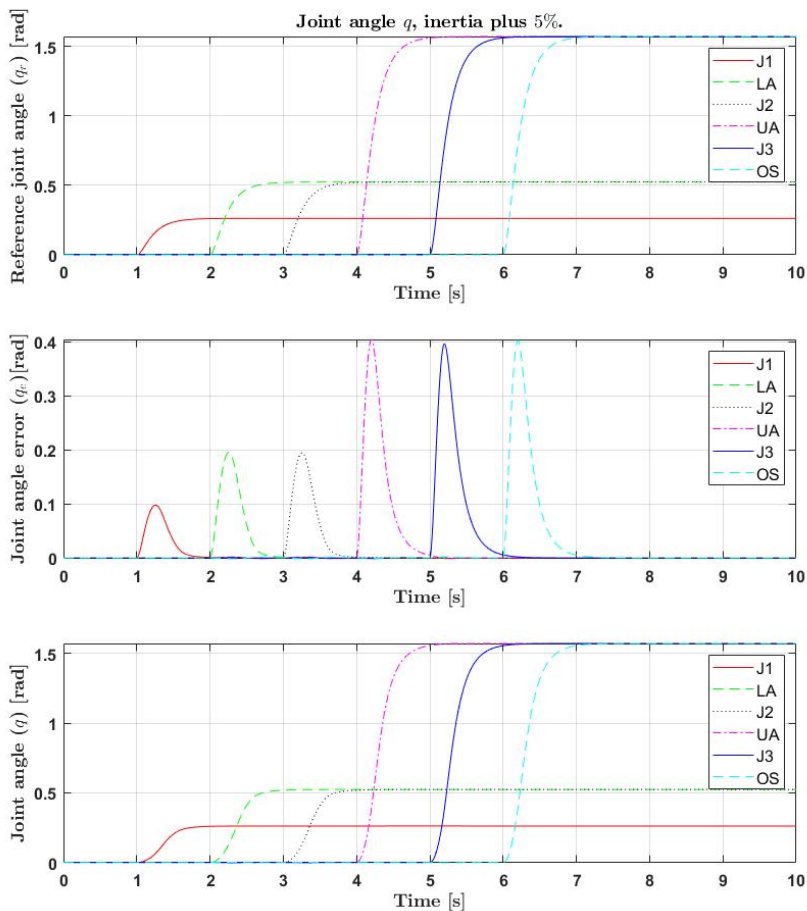


Figure F.1: Inertia test: inertia tensor 5% higher. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

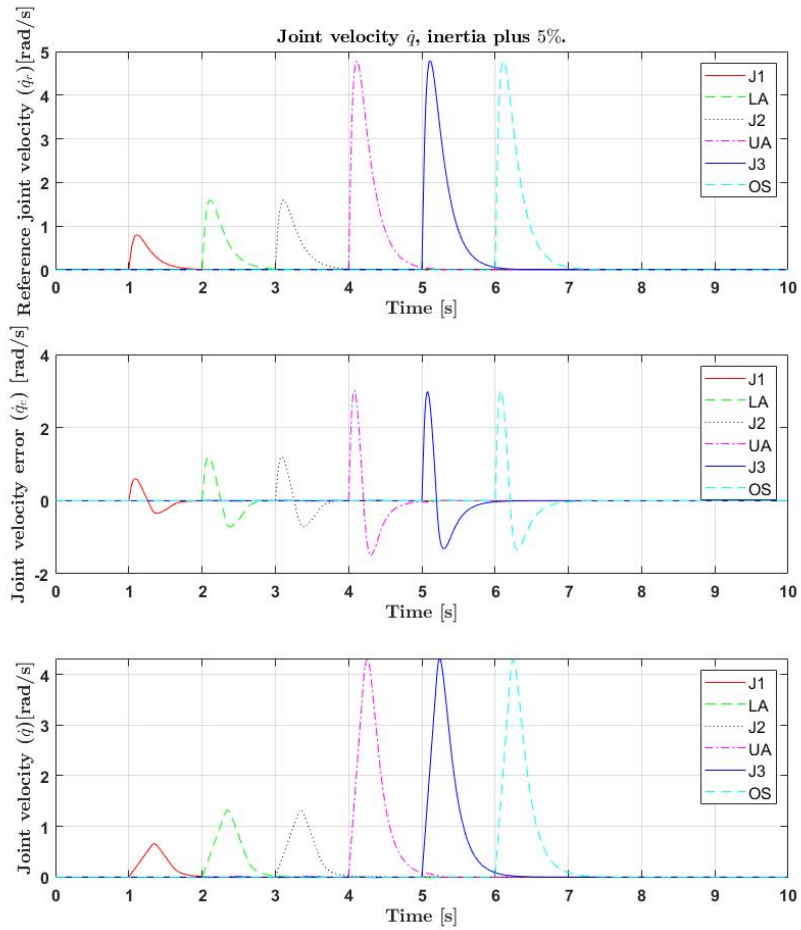


Figure E.2: Inertia test: inertia tensor 5% higher. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

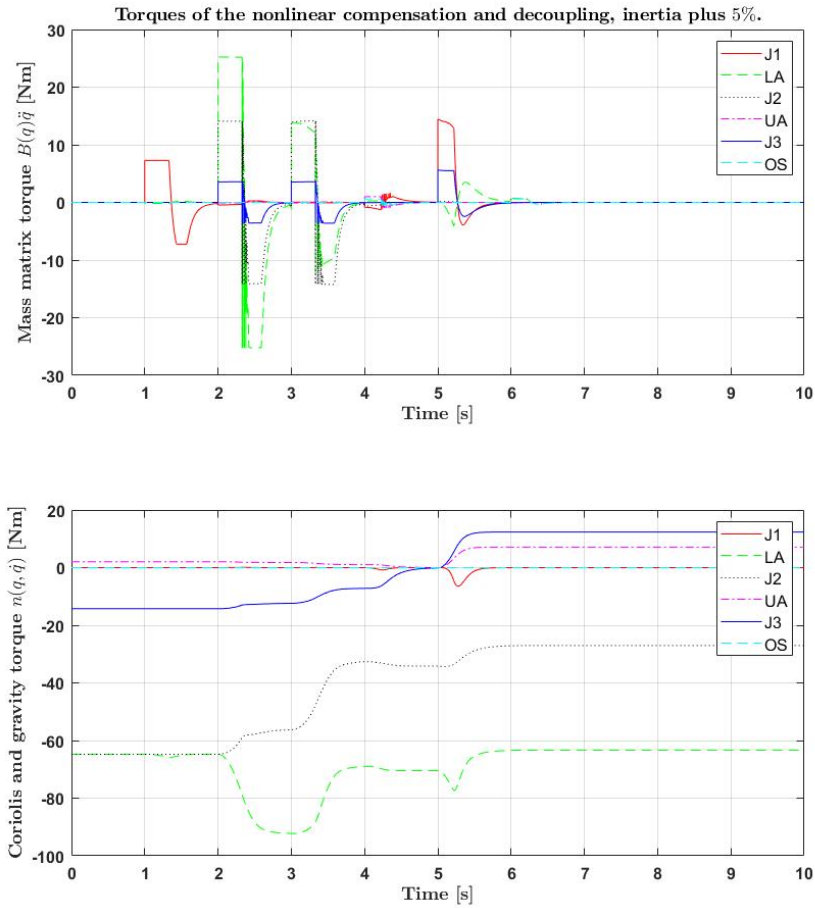


Figure F.3: Inertia test: inertia tensor 5% higher. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

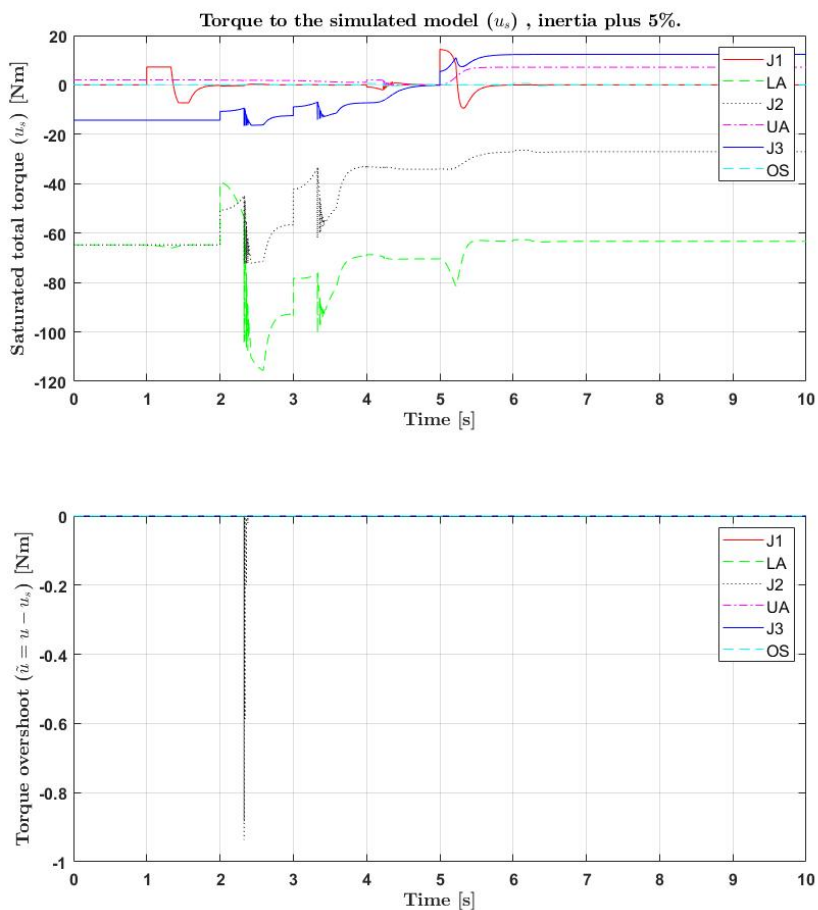


Figure F4: Inertia test: inertia tensor 5% higher. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

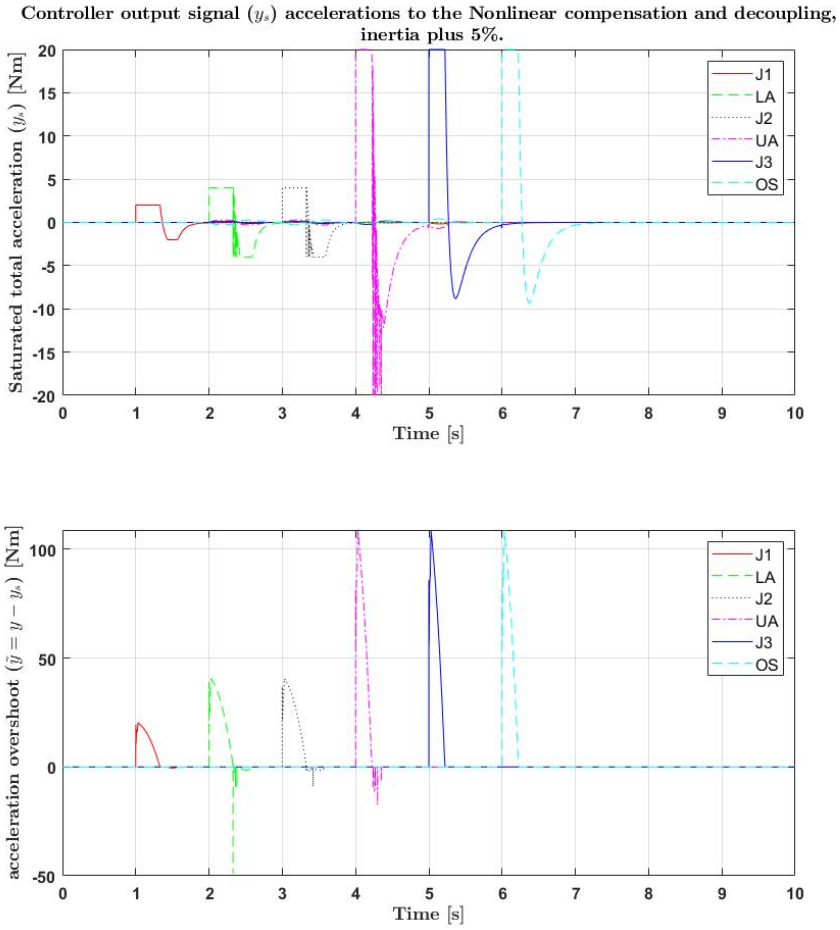


Figure F.5: Inertia test: inertia tensor 5% higher. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

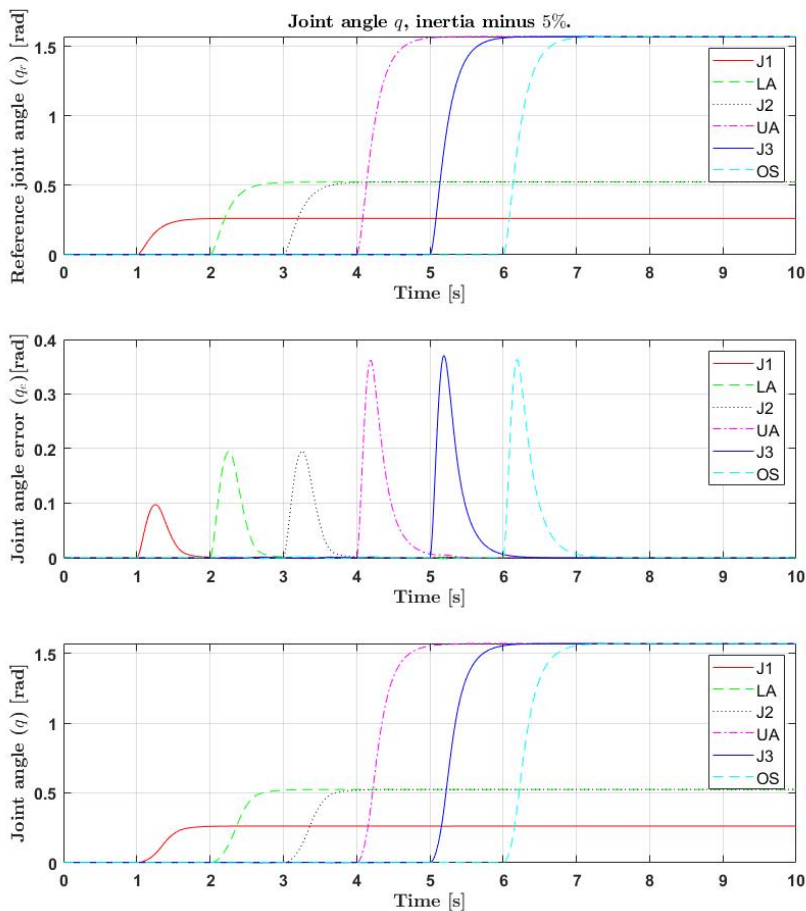


Figure F.6: Inertia test: inertia tensor 5% lower. From top to bottom: joint angle reference signal q_r , joint angle error q_e and actual joint angle q . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

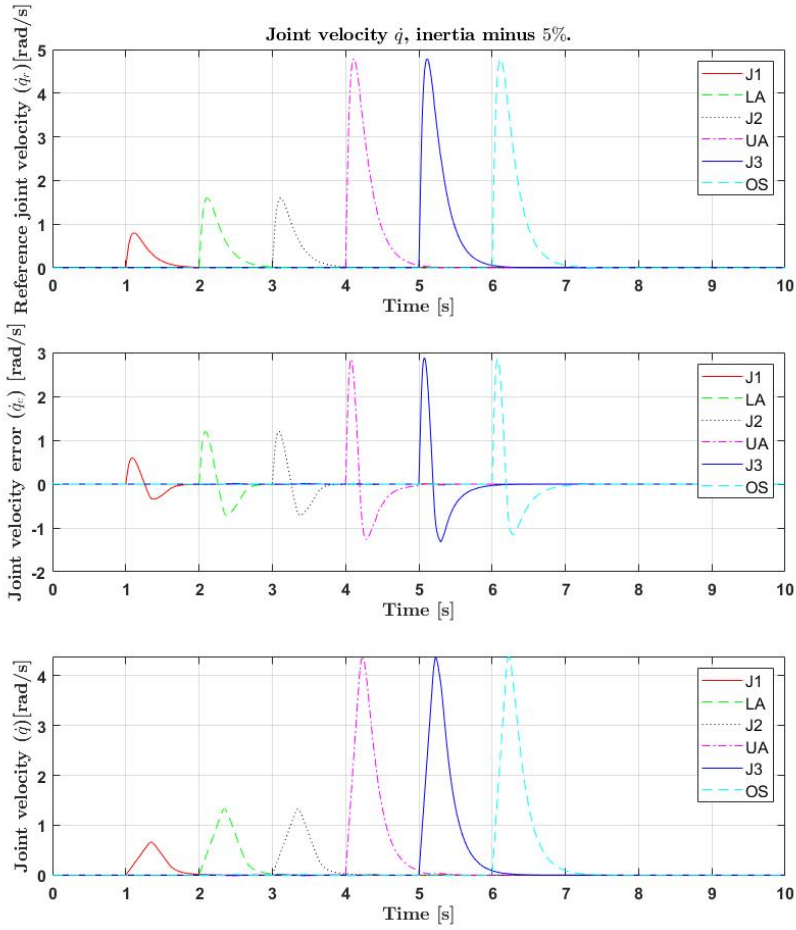


Figure F.7: Inertia test: inertia tensor 5% lower. From top to bottom: joint velocity reference signal \dot{q}_r , joint velocity error \dot{q}_e and actual joint velocity \dot{q} . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

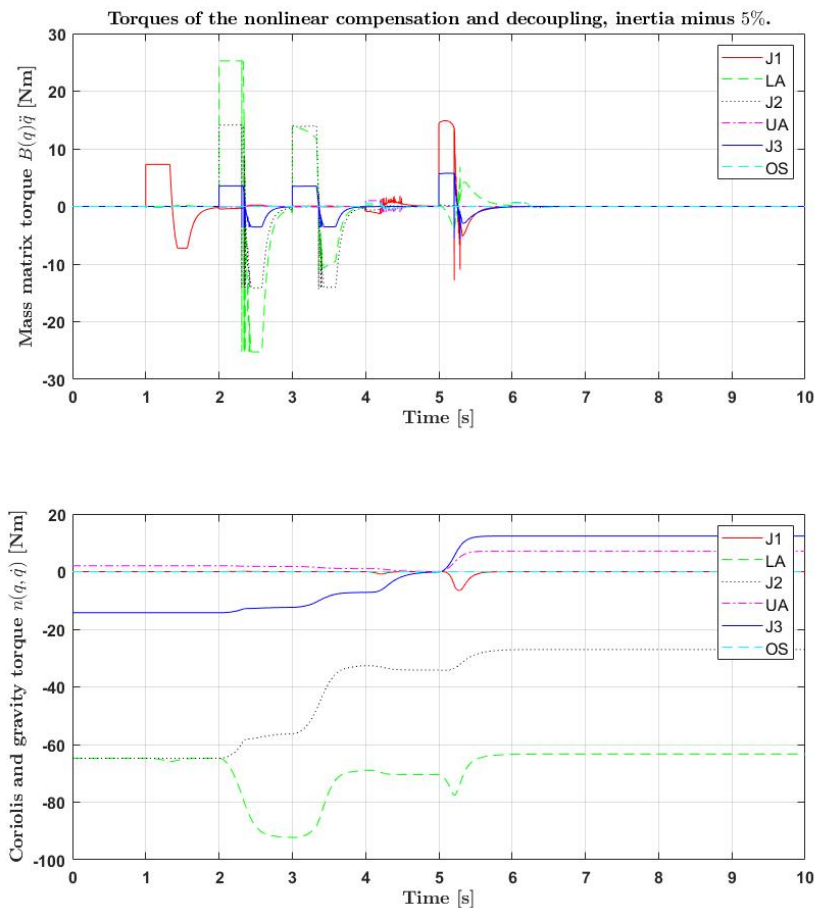


Figure E8: Inertia test: inertia tensor 5% lower. Top graph: torques from the $B(q)\ddot{q}$. Second graph: torques from $n(q, \dot{q})$. The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

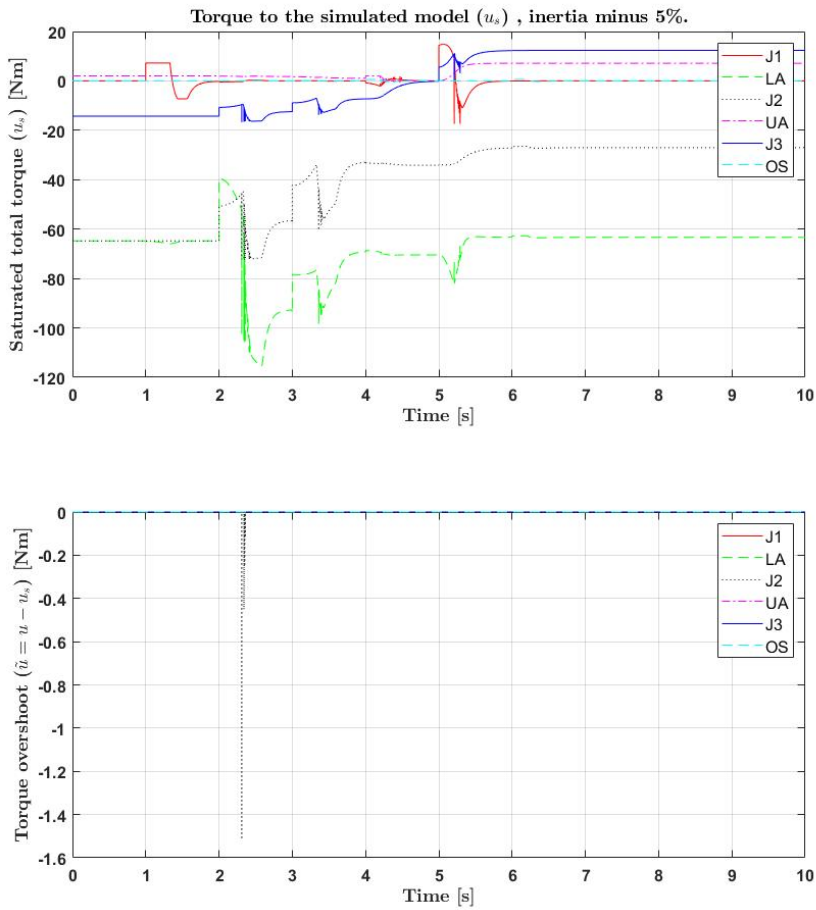


Figure F.9: Inertia test: inertia tensor 5% lower. Top graph: total saturated torques u_s . Second graph: saturated values of u . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

Controller output signal (y_s) accelerations to the Nonlinear compensation and decoupling, inertia minus 5%.

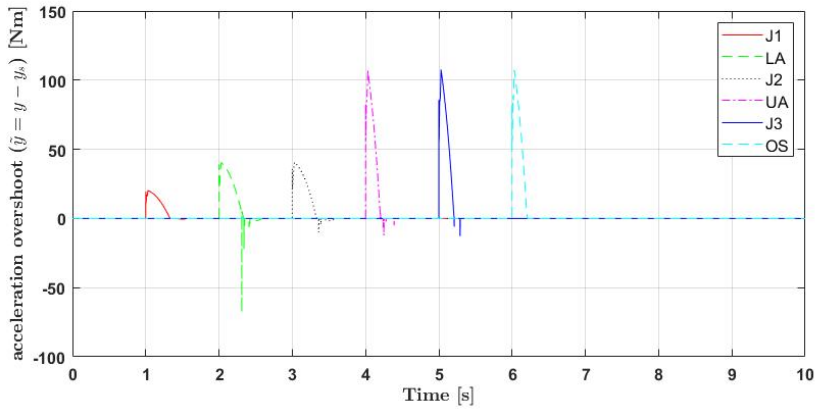
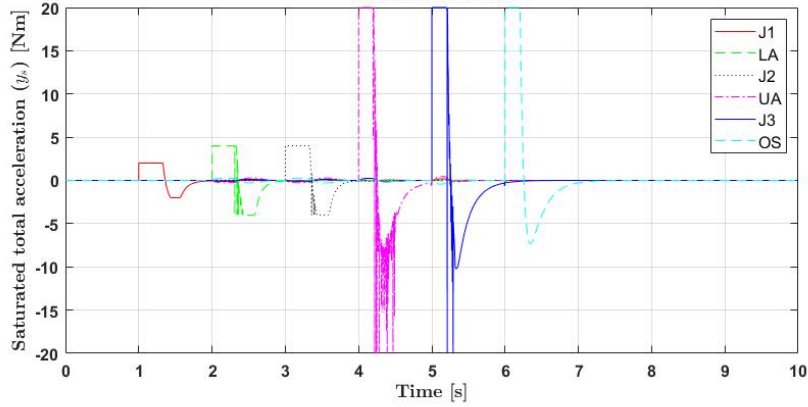


Figure E.10: Inertia test: inertia tensor 5% lower. Top graph: total saturated control signal y_s . Second graph: saturated values of y . The legends (J1, LA, J2, UA, J3, OS) are associated with the joint of each link, see Figure 4.1.

Bibliography

- [1] Mr. Marco Bertagnoli. *Model-Based Stability Analysis for Mobile Manipulators*. PhD thesis, Ostbayerische Technische Hochschule Regensburg, Regensburg, Bavaria, Germany, 06 2017.
- [2] Raphael Deimel and Oliver Brock. A compliant hand based on a novel pneumatic actuator. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2039–2045, 2013.
- [3] Olaf Diegel, Aparna Badve, Glen Bright, Johan Potgieter, and Sylvester Tlale. *Improved Mecanum Wheel Design for Omni-Directional Robots*. PhD thesis, Institute of Technology and Engineering, Massey University., 11 2002.
- [4] Roy Featherstone. Robot dynamics, 2007-10-08. [Online; accessed December 16, 2020], http://www.scholarpedia.org/article/Robot_dynamics.
- [5] Parham M. Kebria, Saba Al-wais, Hamid Abdi, and Saeid Nahavandi. Kinematic and dynamic modelling of UR5 manipulator. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, oct 2016. doi: 10.1109/smc.2016.7844896.
- [6] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.
- [7] MathWorks®. Robotics system toolbox, -. [Online; accessed December 16, 2020], <https://se.mathworks.com/products/robotics.html>.
- [8] MathWorks®. Symbolic math toolbox, . [Online; accessed December 16, 2020], <https://se.mathworks.com/products/symbolic.html>.
- [9] MathWorks®. Getting started with matlab, 1994-2020. [Online; accessed December 16, 2020], <https://se.mathworks.com/products/matlab/getting-started.html>.
- [10] MathWorks®. Matlab, 2018-08-31. [Online; accessed December 16, 2020], <https://www.mathworks.com/videos/series/teaching-rigid-body-dynamics.html>.

- [11] Omron. Omron showcases mobile manipulator at ppma total show, October 01, 2019. URL <https://industrial.omron.eu/en/news-events/news/mobile-manipulator-at-ppma-total-show>. [Online; accessed December 16, 2020].
- [12] Peter Corke. Robotics toolbox, . [Online; accessed December 16, 2020], <https://petercorke.com/toolboxes/robotics-toolbox/>.
- [13] Robotnik. Rising mobile manipulator, -. URL <https://www.robotnik.eu/customization/rising-2/>. [Online; accessed December 16, 2020].
- [14] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer-Verlag, 2009.
- [15] Universal Robots. Inventor of the robot arm and its continued development, 2019-10-01. [Online; accessed December 16, 2020], <https://blog.universal-robots.com/history-of-robot-arm-invention>.
- [16] ROS components. Rb-kairos, 2016. URL <https://www.roscomponents.com/en/13-mobile-manipulators>. [Online; accessed December 16, 2020].