



Norwegian University of  
Science and Technology

# Dynamic Modeling and Simulation of Robot Manipulators

The Newton-Euler Formulation

**Herman Høifødt**

Master of Science in Engineering Cybernetics

Submission date: June 2011

Supervisor: Jan Tommy Gravdahl, ITK





## MSc thesis assignment

Name of the candidate: Herman Høifødt  
Subject: Engineering Cybernetics  
Title: Dynamic Modeling and Simulation of Robot Manipulators: The Newton-Euler Formulation

### **Background**

ABB has produced an industrial robot manipulator called IRB 140. The Department of Engineering Cybernetics at NTNU recently acquired one of these manipulators to do research in robotics, as well as to use it in student lab exercises.

As a continuation of the project work<sup>1</sup>, the student will derive a complete dynamic model of the manipulator by the Newton-Euler formulation.

### **Assignment:**

1. Research dynamic modeling of robot manipulators with particular focus on the Newton-Euler formulation.
2. Search the manipulator datasheets for information that is applicable to the modeling and perform parameter estimation when required.
3. Based on the Newton-Euler formulation, choose an appropriate software tool and design an automated framework for deriving the dynamic model of any serial robot manipulator with revolute joints.
4. Derive the dynamic model for the IRB 140.
5. Simulate the dynamic model in open loop and investigate the energy properties of the model.
6. Perform simulations of position control with a multivariable PD-controller.
7. Compare characteristics of the dynamic model with the dynamic model of the same manipulator derived by the Euler-Lagrange formulation<sup>2</sup>.

To be handed in by: 27/6-2011

Trondheim, 24.01.2011

Jan Tommy Grasdahl  
Professor, supervisor

Uwe Mettin  
Post.doc, advisor

---

<sup>1</sup> H.Høifødt. *Dynamics of robot manipulators by the Newton-Euler formulation*, 5<sup>th</sup> year project work, 2010.

<sup>2</sup> Work done by PhD student S. Pchelkin.



## Abstract

Dynamic modeling means deriving equations that explicitly describes the relationship between force and motion in a system. To be able to control a robot manipulator as required by its operation, it is important to consider the dynamic model in design of the control algorithm and simulation of motion.

In general there are two approaches available; the Euler-Lagrange formulation and the Newton-Euler formulation. This thesis explains briefly the differences of the formulations, and then research the Newton-Euler method in detail. A complete derivation of the method is derived, and an automated framework for applying the method on any serial manipulator with revolute joints is presented.

By using the framework, the Newton-Euler formulation is applied on a modern industrial manipulator with six degrees of freedom. The dynamic parameters of the system are estimated, and the validity of the resulting dynamic model is verified by several simulations in open and closed loop.

The simulations show that the system is unstable in open loop, and that it achieves global asymptotic stability in closed loop with gravity compensation, by including PD controllers with independent joint control. The latter is a confirmation of a mathematical proof based on a Lyapunov stability analysis, which is presented as well. Equivalent simulations of the dynamic model of the same manipulator derived by the standard Euler-Lagrange formulation show that the efficiency of recursive procedures is way higher than treating the manipulator as a whole.

A suggestion for future work is performing thorough dynamic parameter identification. An improved model can ultimately be implemented in the controller of the manipulator, and optimized for a specific job task.



# Preface

This thesis presents the work from the final semester of my Master of Engineering Cybernetics education at the Norwegian University of Science and Technology (NTNU). The topic of the work is dynamic modeling of robot manipulators, where the dynamic model of an industrial manipulator recently acquired by The Department of Engineering Cybernetics has been derived by the Newton-Euler formulation.

I am very grateful for the support from supervisor Jan Tommy Gravdahl, who recommended me to the subject of robotics and gave me the opportunity to research this manipulator. Our regularly meetings have encouraged me through the whole work, and his advices and ideas have been very helpful.

I would also like to thank my advisor Uwe Mettin who has supported me in several modeling challenges during the project. His expertise in robotics and control engineering has been much appreciated.

In the end I would like to thank my fellow co-students for a great time this semester. The numerous hours and days at the office would not have been the same without their help and our social environment.

Trondheim, June 2011  
Herman Høifødt



# Contents

<b>Abstract</b>	<b>i</b>
<b>Preface</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 History and Motivation . . . . .	1
1.2 The IRB 140 . . . . .	2
1.3 Objective . . . . .	2
1.4 Software . . . . .	3
1.5 Outline . . . . .	4
1.6 Contributions . . . . .	5
<b>2 Background Theory and Notation</b>	<b>7</b>
2.1 The Rotation Matrix . . . . .	7
2.1.1 Vectors and Rotational Transformations . . . . .	8
2.1.2 Properties of the Rotation Matrices . . . . .	8
2.1.3 Relation to Skew Symmetric Matrices . . . . .	8
2.2 Kinematic Chains . . . . .	9
2.3 The Inertia Tensor . . . . .	11
2.4 Feedback Controllers . . . . .	11
2.5 Positive and Negative Definiteness . . . . .	12
<b>3 Dynamic Modeling of Robot Manipulators in General</b>	<b>13</b>
3.1 Different Approaches . . . . .	13
3.1.1 Euler-Lagrange versus Newton-Euler . . . . .	14
3.1.2 Dynamic Parameter Identification . . . . .	15
3.2 Derivation of the Newton-Euler Formulation . . . . .	15
3.2.1 The General Case . . . . .	16
3.2.2 Equations of an $n$ -link Manipulator . . . . .	17
3.2.3 Automated Framework . . . . .	20

<b>4</b>	<b>System Description and Dynamic Parameter Estimation</b>	<b>23</b>
4.1	Information from Data Sheets . . . . .	23
4.2	Limitations . . . . .	24
4.3	Modeling Set-up . . . . .	25
4.4	Parameter Estimation . . . . .	26
<b>5</b>	<b>The Dynamic Model</b>	<b>31</b>
5.1	Forward Recursion . . . . .	31
5.2	Backward Recursion . . . . .	33
5.3	Comments . . . . .	35
<b>6</b>	<b>Simulations and Results</b>	<b>37</b>
6.1	Simulation Structure . . . . .	37
6.2	Reduced System Order . . . . .	38
6.3	Open Loop with Desired Torque . . . . .	39
6.3.1	Simulations . . . . .	39
6.3.2	Comments . . . . .	41
6.4	Energy Properties . . . . .	44
6.4.1	Kinetic Energy . . . . .	44
6.4.2	Potential Energy . . . . .	44
6.4.3	Pendulum Comparison . . . . .	45
6.4.4	Simulations . . . . .	45
6.4.5	Comments . . . . .	46
6.5	Closed Loop Position Control . . . . .	49
6.5.1	Proof: PD control with gravity compensation . . . . .	49
6.5.2	Simulations with PD Control . . . . .	51
6.5.3	Comments and Limitations . . . . .	53
<b>7</b>	<b>Comparing the Results to the Euler-Lagrange Model</b>	<b>59</b>
7.1	Remarks on the Direction of State Variables . . . . .	59
7.2	Computation Times . . . . .	62
7.2.1	Open Loop . . . . .	62
7.2.2	Closed loop . . . . .	62
7.2.3	Comments . . . . .	63
<b>8</b>	<b>Concluding Remarks</b>	<b>65</b>
8.1	Conclusion . . . . .	65
8.2	Recommendations for Future Work . . . . .	66
<b>A</b>	<b>IRB 140 Data Sheet</b>	<b>67</b>
<b>B</b>	<b>Automated Framework</b>	<b>71</b>
<b>C</b>	<b>IRB 140 Dynamic Model</b>	<b>77</b>
<b>D</b>	<b>Errata in Chapter 7.6 in [20]</b>	<b>105</b>
<b>E</b>	<b>Contents of Attached ZIP-File</b>	<b>109</b>

# List of Figures

1.1	The IRB 140 with its six degrees of freedom [16] . . . . .	3
2.1	Symbolic representation of robot joints [20] . . . . .	9
2.2	Links, joints and coordinate frames on an elbow manipulator . . .	10
3.1	Forces and torques acting on a random link . . . . .	18
4.1	View of the manipulator from the back and side [16] . . . . .	24
4.2	Symbolic representation of the IRB 140 . . . . .	25
4.3	The centers of mass from the back and side [16] . . . . .	27
4.4	Example of link 2 modeled as a cylinder . . . . .	28
6.1	Open loop simulink model . . . . .	39
6.2	Open loop response with desired torque, simulation 1 . . . . .	42
6.3	Open loop response with desired torque, simulation 2 . . . . .	42
6.4	Open loop response with desired torque, simulation 3 . . . . .	43
6.5	Open loop response with desired torque, simulation 4 . . . . .	43
6.6	Open loop response with zero torque, simulation 1 . . . . .	47
6.7	Energy in the system with zero torque, simulation 1 . . . . .	47
6.8	Open loop response with zero torque, simulation 2 . . . . .	48
6.9	Energy in the system with zero torque, simulation 2 . . . . .	48
6.10	Closed loop simulink model with PD controllers . . . . .	51
6.11	Closed loop position control response, simulation 1 . . . . .	55
6.12	Closed loop position control input, simulation 1 . . . . .	55
6.13	Closed loop position control response, simulation 2 . . . . .	56
6.14	Closed loop position control input, simulation 2 . . . . .	56
6.15	Closed loop position control response, simulation 3 . . . . .	57
6.16	Closed loop position control input, simulation 3 . . . . .	57
7.1	Confirming opposite $q_3$ -direction, simulation 1 . . . . .	60
7.2	Confirming opposite $q_3$ -direction, simulation 2 . . . . .	61
7.3	Euler-Lagrange closed loop position control response, simulation 1	63
7.4	Euler-Lagrange closed loop position control response, simulation 2	64
7.5	Euler-Lagrange closed loop position control response, simulation 3	64



# Chapter 1

## Introduction

Robotics is concerned with the study of machines that can replace human beings. The goal of this introductory chapter is to express the motivation behind the thesis, and to give an overview of the contents. The IRB 140 is introduced, as well as the objective and the software that has been used to solve it. An outline and the contributions of the thesis is presented in the end of the chapter. Historical facts are taken from [20] and [17].

### 1.1 History and Motivation

The English term *robot* was derived from the Czech word *robota* that means executive labor, and was first introduced by the Czech playwright Karel Čapek in his 1921 play *Rossum's Universal Robots*. Since then the term has been applied to virtually anything that operates with some degree of autonomy, usually under computer control. An official definition of the term, dated to 1980, comes from the Robot Institute of America (RIA) and reflects today's status of robotics technology:

*A robot is a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks.*

In the early 1980's, robot manipulators were touted as the ultimate solution to automated manufacturing. Predictions were that entire factories of the future would require few, if any, human operators. It turned out that these predictions were a little exaggerated, as the savings in labor costs often did not outweigh the development costs of creating robot systems. Quite simply, people are good at what they do, and installing a robot involves complex systems integration problems. As a result, robotics fell out of favor in the late 1980's.

A resurgence of interest in robotics can be witnessed in the recent years. Deeper understanding of the subject and new technology have made it possible for robots to explore the surface on Mars, locate sunken ships, searching

out land mines, and finding victims in collapsed buildings. In an industrial environment the advantages of robots are reduction of manufacturing costs, increase of productivity, improvement of quality standards, and the possibility of eliminating harmful tasks for human operators.

## 1.2 The IRB 140

The IRB 140 (full name: IRB 140-6/0.8) is an industrial robot produced by ABB, designed specifically for manufacturing industries. Their website [14] presents various facts about the manipulator, as well as articles, data sheets and movies. The manipulator has a total of six revolute joints that are controlled by AC-motors, hence six degrees of freedom (6 DOF). Figure 1.1 gives a clear view of the manipulator and its degrees of freedom. The compact and robust design is adapted for flexible use, and the robot can be mounted on the floor, the wall or the roof in any angle. It offers outstanding accuracy and speed, and suits a lot of industrial tasks as for example spray painting, packing and palletizing. The website presents some movies of the manipulator in work from companies using it in their industry.

One of the data sheets on the website presents the *Fanta Can Challenge*. This was a showcase from ABB to demonstrate the great accuracy of their motion control systems called *QuickMove* and *TrueMove*. The idea about *QuickMove* is to reach the end position in the shortest possible time, while *TrueMove* ensures that the motion path followed is exactly the same independent of speed. The latest experiment from 2009 involved three IRB 140 robots cooperating to demonstrate the outstanding accuracy of the *TrueMove* concept. Movies from the showcase are published on the website.

To achieve such accurate motion control, a complete dynamic model of the robot is held in the robot controller. This model is considered a valuable trade secret in ABB, and is not available even for buyers of the product. Dynamic parameters in their model are most likely estimated very thoroughly by numerous days and months of measurements and tests with the manipulator. Dynamic parameter identification is probably the greatest challenge in dynamic modeling of robot manipulators, and will be highlighted in later chapters.

Discovering the *Fanta Can Challenge* as well as industrial areas of utilization of the IRB 140 has definitely been a motivation for this thesis.

## 1.3 Objective

The objective of this thesis is to derive the complete dynamic model of the IRB 140 by the Newton-Euler formulation. The work is a continuation of the project work [6], where the dynamic model of the first three degrees of freedom of the manipulator were derived. All degrees of freedom are still modeled in this thesis, and it should be mentioned that some definitions and parameters used in the project work are modified in this thesis to better suit the complete

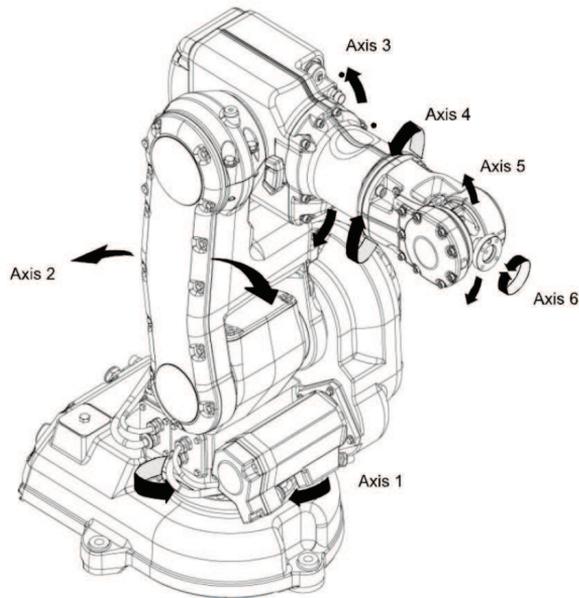


Figure 1.1: The IRB 140 with its six degrees of freedom [16]

model.

Different approaches on dynamic modeling of robot manipulators will be discussed, with particular focus on the Newton-Euler formulation. An automated framework for deriving the dynamic model by the Newton-Euler formulation will be presented, and this framework will then be utilized for the IRB 140. Parameters will be defined by searching for applicable information in the manipulator data sheets, as well as by parameter estimation when required. Limitations in the dynamic model will be described to underline how it differs from a perfect model.

The model will be studied in open loop and closed loop to investigate stability and energy properties. Based on a Lyapunov stability analysis, a mathematical proof of global asymptotic stability for position control will be presented. Then the model for the IRB 140 will be simulated in closed loop to verify this mathematical proof.

Ultimately, some results will be compared to the results of an equivalent dynamic model derived by the Euler-Lagrange formulation [13].

## 1.4 Software

Three computer programs have been used to solve the thesis assignment. Following is a short description of this software and the area of utilization.

## Maple 15

Maple [9] is developed by MapleSoft, and is a technical computing software for doing symbolic, numeric and graphical computations. Because of its great efficiency in symbolic computations, Maple has been used to derive the dynamic model for the IRB 140. The framework for deriving dynamic models of any serial robot manipulator with revolute joints has been designed in Maple as well.

## Matlab R2010a with Simulink 7.5

Matlab [10] is developed by MathWorks, and is a high-level language and numerical computing environment for performing computationally intensive tasks faster than with traditional programming languages. It offers tight integration with other MathWorks products, among them Simulink [11] which is an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems. Matlab and Simulink have been used to simulate the dynamic model for the IRB 140, and to present the results graphically.

## Adobe Illustrator CS4

Adobe Illustrator [21] is a vector graphics editor developed by Adobe Systems. It offers sophisticated drawing tools with high precision that is intuitively easy to use. Most figures in this thesis are designed with this software.

## 1.5 Outline

- **Chapter 2:** Fundamental background theory and notation used throughout the thesis are explained in this chapter. It is put importance on the standard convention of how to interpret robot manipulators, as well as the concept of rotation matrices.
- **Chapter 3:** This chapter presents different approaches on dynamic modeling of robot manipulators, and compares the Newton-Euler formulation to the Euler-Lagrange formulation. The derivation of the Newton-Euler formulation is followed in detail, and an automated framework for dynamic modeling using this formulation is presented.
- **Chapter 4:** It is shown how the IRB 140 is interpreted as a kinematic chain following the standard convention. Parameters of the system is defined on the basis of information from the manipulator data sheets and parameter estimation.
- **Chapter 5:** The Newton-Euler formulation is applied on the IRB 140, and the forward and backward recursions are followed for all links in the kinematic chain.

- **Chapter 6:** This chapter presents simulations of the system in open loop and closed loop. The results include energy investigations and stability analyses, where simulations in closed loop with PD controllers verifies a remarkable mathematical proof.
- **Chapter 7:** It is presented a comparison of some of the results from this thesis, with the results of the dynamic model of the same manipulator by the standard Euler-Lagrange formulation.
- **Chapter 8:** This chapter presents the concluding summary of the thesis, as well as recommendations for future work.

## 1.6 Contributions

- A complete derivation of the Newton-Euler formulation in general. The derivation is based on the one in [20], but is corrected and restated due to the discoveries of misleading notations and errata in the book (see Appendix D).
- An automated framework for deriving the dynamic model of any serial robot manipulator with rigid links and revolute joints. It can easily be adjusted to any number of degrees of freedom, and if desired, it is not a comprehensive task to modify it for prismatic joints as well.
- A convenient representation of the IRB 140 as a kinematic chain of single degree-of-freedom joints. The way the frames are attached leads to significant simplifications in the derivation of the dynamic model by the Newton-Euler formulation.
- The complete dynamic model of the IRB 140, including simulations that demonstrate satisfying behavior of the model in open loop and closed loop. The framework used is designed in such a way that all parameters can be easily modified.
- An investigation of efficiency when using the Newton-Euler formulation and the Euler-Lagrange formulation to derive the dynamic model for the IRB 140. The results show a clear advantage of utilizing a recursive procedure for such a complex system.



## Chapter 2

# Background Theory and Notation

This thesis follows the standard convention of how a robot manipulator is interpreted. Fundamental background theory and important notation that are used throughout the thesis are briefly explained in this chapter to facilitate the understanding of the later chapters.

Understanding the concept of rotation matrices is an essential part of modeling robot manipulators. Section 2.1 describes rotational transformations and presents some important properties of rotation matrices and their relation to skew symmetric matrices. Section 2.2 shows how a robot manipulator is composed by links and joints to form a kinematic chain, and presents guidelines of how to define the links, joints and frames. The material is mainly taken from [20].

The rest of the chapter is dedicated to the concepts of feedback controllers, the inertia tensor, and positive and negative definiteness. A complete description of these concepts are given in most books about control theory, e.g. [3], [8] and [20].

### 2.1 The Rotation Matrix

In order to perform algebraic manipulations with vectors using coordinates, it is essential that all vectors are expressed in the same coordinate frame. Rotation matrices are used to accomplish this. An  $n \times n$  rotation matrix specifies the orientation of one frame relative to another frame in the  $n$ -dimensional Euclidean space. To specify the coordinate vectors of frame 1 with respect to frame 0 in three dimensions, the  $3 \times 3$  rotation matrix is written as

$$R_1^0 = [x_1^0 \quad y_1^0 \quad z_1^0] \tag{2.1}$$

where the columns are the coordinates of the vectors  $x_1$ ,  $y_1$ , and  $z_1$  expressed in frame 0.

### 2.1.1 Vectors and Rotational Transformations

A vector specifies a direction and a magnitude in the Euclidean space, and can be assigned coordinates with respect to a reference coordinate frame. That means a particular vector is always the same geometric entity, but the representation of the vector by coordinates depends directly on the reference coordinate frame. So that the reference frame will always be clear, vectors are denoted with a superscript to show in which frame they are expressed. Introducing two frames (frame 0 and frame 1) and a vector  $v$ , that means  $v^0$  is  $v$  expressed in frame 0, while  $v^1$  is  $v$  expressed in frame 1. Introducing a third frame (frame 2), the relationship between the vector  $v$  and the frames becomes

$$v^0 = R_1^0 v^1 \quad (2.2)$$

$$v^1 = R_2^1 v^2 \quad (2.3)$$

$$v^0 = R_2^0 v^2 \quad (2.4)$$

which gives

$$R_2^0 = R_1^0 R_2^1 \quad (2.5)$$

### 2.1.2 Properties of the Rotation Matrices

In a coordinate frame, the axes are always defined as unit vectors that are mutually orthogonal. A rotation matrix specifies the relationship between two such frames, and is therefore also said to be orthogonal. Restricting all coordinate frames to be right handed frames, such rotation matrices are denoted as the Special Orthogonal group of order  $n$ , shortened as  $SO(n)$ , where  $n$  is the number of dimensions. Any rotation matrix  $R \in SO(n)$  has some important properties, and they are

- $R^T = R^{-1} \in SO(n)$
- The columns (and therefore the rows) of  $R$  are mutually orthogonal
- Each column (and therefore each row) of  $R$  is a unit vector
- $\det R = 1$

### 2.1.3 Relation to Skew Symmetric Matrices

An  $n \times n$  matrix  $S$  is said to be skew symmetric if and only if

$$S^T + S = 0 \quad (2.6)$$

which means that every  $3 \times 3$  skew symmetric matrix has the form

$$S = \begin{bmatrix} 0 & -s_3 & -s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix} \quad (2.7)$$

Skew symmetric matrices have been found useful in relation to rotation matrices. Four important properties are given below.

1. For any vectors  $a$  and  $p$  belonging to  $\mathbb{R}^3$ ,

$$S(a)p = a \times p \quad (2.8)$$

where  $S$  is a  $3 \times 3$  skew symmetric matrix.

2. For  $R \in SO(3)$  and  $a \in \mathbb{R}^3$

$$RS(a)R^T = S(Ra) \quad (2.9)$$

where  $S$  is a  $3 \times 3$  skew symmetric matrix.

3. In the general case of angular velocity about an arbitrary and possibly moving axis we have

$$\dot{R}(t) = S(\omega(t))R(t) \quad (2.10)$$

where  $R = R(t) \in SO(3)$  for every  $t \in \mathbb{R}$ ,  $S$  is a  $3 \times 3$  skew symmetric matrix, and  $\omega(t)$  is the angular velocity of the rotating frame with respect to the fixed frame at time  $t$ .

4. For an  $n \times n$  skew symmetric matrix  $S$  and any vector  $X \in \mathbb{R}^n$

$$X^T S X = 0 \quad (2.11)$$

## 2.2 Kinematic Chains

Robot manipulators are composed of links connected by joints to form a kinematic chain, where the joints are revolute or prismatic. A revolute joint is like a hinge and allows relative rotation between two links, while a prismatic joint allows a linear relative motion between two links. Both types of joints have a single degree of freedom, thus each joint  $i$  can be represented by a single joint variable  $q_i$ . Figure 2.1 shows a symbolic representation of robot joints in 2D and 3D.

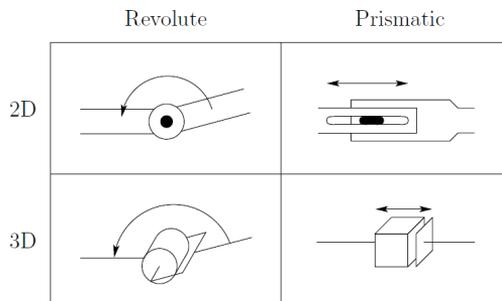


Figure 2.1: Symbolic representation of robot joints [20]

A configuration of a manipulator is a complete specification of every point on the manipulator. Assuming a manipulator with rigid links and a fixed base,

that means the configuration is entirely given by  $q$ , the vector of joint variables. In case of joints with more degrees of freedom, like a ball or a spherical wrist, these joints can always be thought of as a succession of joints with a single degree of freedom.

A coordinate frame is rigidly attached to each link, and an inertial frame is attached to the robots base. Links, joints and frames are defined as summarized below.

- Links are numbered from 0 to  $n$  where link 0 is the base.
- Joints are numbered from 1 to  $n$  where joint  $i$  connects link  $i - 1$  to link  $i$ .
- When joint  $i$  is actuated, link  $i$  moves. The base can not be actuated.
- Frames are numbered from 0 to  $n$  where frame  $i$  is attached to link  $i$ .
- Frames are attached such that axis  $z_i$  of frame  $i$  is the axis of actuation for joint  $i + 1$ .
- The joint variable  $q_i$  is associated with joint  $i$ .

As an example, Figure 2.2 shows these definitions applied in the case of an elbow manipulator with three degrees of freedom.

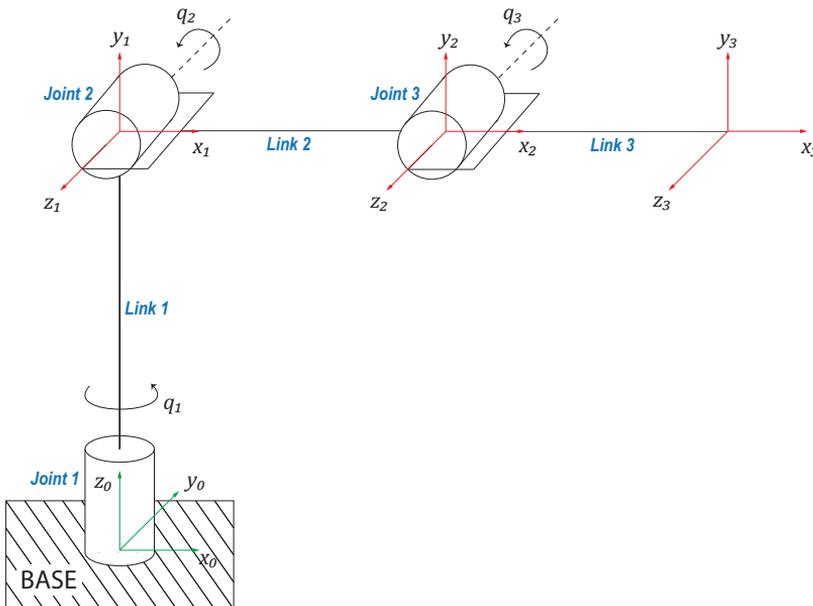


Figure 2.2: Links, joints and coordinate frames on an elbow manipulator

## 2.3 The Inertia Tensor

The moment of inertia is a scalar value expressing the resistance to changes to the rotation of an object. If the axis of rotation is not given, it is possible to generalize the scalar moment of inertia to a  $3 \times 3$  matrix expressing the moment of inertia about arbitrary axes. This matrix is called the inertia tensor.

Let the mass density of an object be represented as a function of position,  $\rho(x, y, z)$ . Then the inertia tensor in a frame attached to the center of mass of the object is computed as

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (2.12)$$

where

$$I_{xx} = \iiint (y^2 + z^2) \rho(x, y, z) dx dy dz \quad (2.13)$$

$$I_{yy} = \iiint (x^2 + z^2) \rho(x, y, z) dx dy dz \quad (2.14)$$

$$I_{zz} = \iiint (x^2 + y^2) \rho(x, y, z) dx dy dz \quad (2.15)$$

and

$$I_{xy} = I_{yx} = - \iiint xy \rho(x, y, z) dx dy dz \quad (2.16)$$

$$I_{xz} = I_{zx} = - \iiint xz \rho(x, y, z) dx dy dz \quad (2.17)$$

$$I_{yz} = I_{zy} = - \iiint yz \rho(x, y, z) dx dy dz \quad (2.18)$$

The diagonal elements  $I_{xx}, I_{yy}, I_{zz}$  are called the principal moments of inertia, and the off-diagonal terms are called the cross products of inertia. If the mass distribution of the object is symmetric with respect to the attached frame, then the inertia tensor will be diagonal (cross products of inertia are identically zero).

## 2.4 Feedback Controllers

A system can be controlled in open loop or closed loop. With an open-loop controller, the input is computed without observing the output that it is controlling. Complex systems will not be possible to control in open loop, because the controller will never know if the output has achieved the desired goal. However, by adding feedback controllers, it might be possible to stabilize the system in closed loop.

A feedback controller observes the output and calculates the error between this output and a reference value. Then the input is computed based on this

error such that the output approaches the reference value. To achieve a desired behavior of the output, controllers can take one or more of three standard control elements. These elements are

- **P - proportional term:** The input is proportional to the error between the reference value and the current output.  $K_p$  is the proportional gain.
- **I - integral term:** Integrates the error over time and multiplies with the integral gain  $K_i$ . The term eliminates steady state error.
- **D - derivative term:** Determines the slope of the error over time and multiplies with the derivative gain  $K_d$ . The term has as a damping effect.

## 2.5 Positive and Negative Definiteness

A function  $V(x)$  satisfying the conditions  $V(0) = 0$  and  $V(x) > 0$  for  $x \neq 0$  is said to be positive definite. If it satisfies  $V(0) = 0$  and the weaker second condition  $V(x) \geq 0$  for  $x \neq 0$  it is said to be positive semidefinite. The function  $V(x)$  is said to be negative definite if  $-V(x)$  is positive definite, and negative semidefinite if  $-V(x)$  is positive semidefinite.

Functions of the quadratic form

$$V(x) = x^T P x \tag{2.19}$$

where  $P$  is a real symmetric matrix, can easily be checked for sign definiteness by inspecting  $P$ . If all eigenvalues of  $P$  are positive (nonnegative), which is true if and only if all the leading principal minors of  $P$  are positive (nonnegative), then  $V(x)$  is a positive definite (positive semidefinite) function. If  $V(x)$  is a positive definite (positive semidefinite) function, the matrix  $P$  is also said to be positive definite (positive semidefinite). Finally,  $P$  is said to be negative definite if  $-P$  is positive definite, and negative semidefinite if  $-P$  is positive semidefinite.

It is common practice to write  $P > 0$  if  $P$  is positive definite, and  $P \geq 0$  if  $P$  is positive semidefinite.

## Chapter 3

# Dynamic Modeling of Robot Manipulators in General

Robot manipulators can be described mathematically in different ways. The problem of kinematics is to describe the motion of the manipulator without consideration of forces and torques causing the motion. These equations determine the position and orientation of the end effector given the values for the joint variables (forward kinematics), and as the opposite the values of the joint variables given the position and orientation of the end effector (inverse kinematics).

Dynamic modeling means deriving equations that explicitly describes the relationship between force and motion. These equations are important to consider in simulation of robot motion, and in design of control algorithms. The most important concepts in fundamental robotics including kinematics, dynamics and control are given in [20] and [17].

Section 3.1 deals with different approaches on dynamic modeling of robot manipulators. The Euler-Lagrange formulation and the Newton-Euler formulation are introduced, and techniques for dynamic parameter estimation are briefly described.

General equations of the Newton-Euler formulation are derived in detail in Section 3.2. The whole derivation from the basis of mechanic laws to the final equations of  $n$ -link manipulators is followed in detail. The chapter ends by presenting an automated framework for deriving the dynamic manipulator of any serial robot manipulator with revolute joints.

### 3.1 Different Approaches

Computing the dynamics of robot manipulators can be challenging. Researchers have discovered different approaches, where in general there are two methods available; the Euler-Lagrange formulation and the Newton-Euler formulation. In the standard Euler-Lagrange formulation the manipulator is treated as a

whole, and the system is analyzed based on its kinetic and potential energy. The Newton-Euler formulation is quite different because each link of the manipulator is treated in turn. First there is a forward recursion describing its linear and angular motion, then a backward recursion to calculate the forces and torques. Both of these formulations are derived from first principles in [20] and [17], including examples of how the methods can be applied. The resulting dynamic model is the same for both methods and can be written in matrix form as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \quad (3.1)$$

where

$q$	=	vector of joint variables
$u$	=	vector of torques
$M$	=	inertia matrix
$C$	=	centrifugal and Coriolis terms
$g$	=	gravity vector

### 3.1.1 Euler-Lagrange versus Newton-Euler

The efficiency of the Euler-Lagrange formulation and the Newton-Euler formulation is an interesting topic. Actually there is no clear answer to the question of which method is better than the other. The main goal is to derive the dynamic model as fast as possible, and how well this goal is satisfied for each method depends on several factors. The number of link and joints in the kinematic chain, the topology of the chain (e.g. serial or parallel), the position and orientation of the coordinate frames, and whether a recursive procedure is used or not, are factors that will influence the computation time.

The Newton-Euler formulation is usually the preferred choice for manipulators with many degrees of freedom. The reason is the recursive structure which the Newton-Euler formulation is based on. If the frames are attached in a convenient way (see Section 5.1), the recursions will be greatly simplified. This advantage of the Newton-Euler formulation is supported in [7] which proves that a recursive approach is in general faster than treating the manipulator as a whole. It should also be mentioned that for the case of parallel manipulators, it is shown in [5] that the Newton-Euler formulation gives an advantage for dynamic computations and control.

However, [7] shows that it is also possible to compute the Euler-Lagrange formulation in a recursive procedure, and [19] shows that the computation time of the two algorithms are equivalent if the frames are attached optimally in both formulations.

In the end, the choice of algorithm is a matter of personal preference, and the main reason for choosing one method over the other is that it might provide different insights. An interesting detail is discovered and explained in Chapter 5, when the Newton-Euler formulation is applied on the IRB 140.

### 3.1.2 Dynamic Parameter Identification

Using the dynamic model (3.1) for solving simulation and control problems demands the knowledge of values of dynamic parameters of the manipulator. In general, a given rigid body is described by ten such parameters; the mass, the six independent entries of the inertia tensor, and the three coordinates of the center of mass. However, due to constraints and coupled kinematics, this number could be lower for a manipulator. An  $n$ -link robot then has a maximum of  $10n$  dynamic parameters. Estimating the parameters from the design data of the manipulator is not simple, but there are a few techniques available.

In order to find accurate estimates of the dynamic parameters, it is possible to use an identification technique which exploits that the dynamic model are linear with respect to a suitable set of dynamic parameters. The system (3.1) can be written as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = Y(q, \dot{q}, \ddot{q})\Theta \quad (3.2)$$

where  $Y(q, \dot{q}, \ddot{q})$  is called the regressor and  $\Theta$  is the parameter vector. Assuming that values of the joint positions  $q$ , velocities  $\dot{q}$ , and accelerations  $\ddot{q}$  can be recorded during execution of trajectories with the manipulator, and that the joint torques can be measured from sensors in the joints or current measurements, it is now possible to calculate the dynamic parameters directly by the parameterized system (3.2). However, finding a minimal set of parameters that can parametrize the dynamic model is difficult in general, and as mentioned, measurements of  $q, \dot{q}, \ddot{q}$  during motion is a requirement for using this method.

Another approach is Computer-Aided-Design (CAD) modeling. The various components of the manipulator are then modeled digitally on the basis of their geometry and type of materials, and features in the CAD system can be used to measure the parameters. Inaccuracies will occur with this technique, because of simplifications in the modeling and the loss of information about complex dynamic effects like joint friction.

Dynamic parameter identification is explained more detailed in [20] and [17].

## 3.2 Derivation of the Newton-Euler Formulation

The rest of this chapter is dedicated to derive equations of the Newton-Euler formulation to be applied on the IRB 140. Section 3.2.1 describes the general case based on important mechanic laws, while Section 3.2.2 modifies the equations to suit any serial  $n$ -link manipulator with revolute joints.

Section 3.2.3 presents a framework for deriving the dynamic model of any serial robot manipulator with only revolute joints. The framework is designed in Maple and can be found in the attached ZIP-file, in addition to the printed version showed in Appendix B.

The following derivation of the Newton-Euler formulation is entirely based on the derivation presented in [20]. The reason for restating all the details is that there was discovered several misleading notations and errata in the book. Appendix D presents a list of these findings.

### 3.2.1 The General Case

The basis of the Newton-Euler formulation is three important mechanic laws:

- Every action has an equal and opposite reaction. Thus, if link 1 applies a force  $f$  and torque  $\tau$  to link 2, then link 2 applies a force  $-f$  and torque  $-\tau$  to link 1.
- The rate of change of the linear momentum equals the total force applied to the link.
- The rate of change of the angular momentum equals the total torque applied to the link.

Applying the second law to the linear motion of a link gives the relationship

$$\frac{d(mv)}{dt} = f \quad (3.3)$$

where  $m$  is the mass of the link,  $v$  is the velocity of the center of mass with respect to an inertial frame, and  $f$  is the sum of external forces applied to the link. Since the mass is constant as a function of time for robot manipulators, Equation (3.3) can be simplified to

$$f = ma \quad (3.4)$$

where  $a$  is the acceleration of the center of mass.

The third law gives the relationship

$$\frac{d(I_0\omega_0)}{dt} = \tau_0 \quad (3.5)$$

where  $I_0$  is the moment of inertia of the link,  $\omega_0$  is the angular velocity of the link, and  $\tau_0$  is the sum of torques applied on the link. All three variables are expressed in an inertial frame whose origin is at the center of mass. Note that  $I_0$  is not necessarily a constant function of time, but this can be taken care of by rewriting Equation (3.5) to be valid for a frame rigidly attached to the link instead of an inertial frame. A similarity transformation of  $I_0$  is given by

$$I = R^{-1}I_0R \quad (3.6)$$

which gives

$$I_0 = RIR^T \quad (3.7)$$

where  $R$  is the rotation matrix that transforms coordinates from the link attached frame to the inertial frame. Equation (3.7) together with the facts

$$\omega_0 = R\omega, \quad \tau_0 = R\tau \quad (3.8)$$

yields

$$\frac{d(I_0\omega_0)}{dt} = \frac{d(RIR^T R\omega)}{dt} \quad (3.9)$$

$$= \frac{d(RI\omega)}{dt} \quad (3.10)$$

$$= \dot{R}I\omega + RI\dot{\omega} \quad (3.11)$$

and the equation for the rate of change of the angular momentum with respect to the link attached frame is

$$\tau = R^T \tau_0 \quad (3.12)$$

$$= R^T (\dot{R}I\omega + RI\dot{\omega}) \quad (3.13)$$

$$= R^T \dot{R}I\omega + I\dot{\omega} \quad (3.14)$$

The rotation matrix in Equation (3.14) can be cancelled out by taking advantage of the properties (2.8), (2.9) and (2.10). The final torque expression becomes

$$\tau = R^T \dot{R}I\omega + I\dot{\omega} \quad (3.15)$$

$$= R^T S(\omega_0)RI\omega + I\dot{\omega} \quad (3.16)$$

$$= S(R^T \omega_0)I\omega + I\dot{\omega} \quad (3.17)$$

$$= S(\omega)I\omega + I\dot{\omega} \quad (3.18)$$

$$= \omega \times (I\omega) + I\dot{\omega} \quad (3.19)$$

This concludes the general case of the derivation with the force balance and moment balance summarized respectively as

$$f = ma \quad (3.20)$$

$$\tau = \omega \times (I\omega) + I\dot{\omega} \quad (3.21)$$

### 3.2.2 Equations of an $n$ -link Manipulator

To begin with, several vectors need to be introduced. Note that all these vectors are expressed in frame  $i$ .

$a_{c,i}$	=	acceleration of the center of mass of link $i$
$a_{e,i}$	=	acceleration of the end of link $i$ (origin of frame $i + 1$ )
$\omega_i$	=	angular velocity of frame $i$ with respect to frame 0
$\alpha_i$	=	angular acceleration of frame $i$ with respect to frame 0
$z_i$	=	axis of actuation of frame $i$ with respect to frame 0
$g_i$	=	acceleration due to gravity
$f_i$	=	force exerted by link $i - 1$ on link $i$
$\tau_i$	=	torque exerted by link $i - 1$ on link $i$
$R_{i+1}^i$	=	rotation matrix from frame $i$ to frame $i + 1$
$m_i$	=	the mass of link $i$
$I_i$	=	inertia tensor of link $i$ about a frame parallel to frame $i$ whose origin is at the center of mass of link $i$
$r_{i-1,ci}$	=	vector from the origin of frame $i - 1$ to the center of mass of link $i$
$r_{i-1,i}$	=	vector from the origin of frame $i - 1$ to the origin of frame $i$
$r_{i,ci}$	=	vector from the origin of frame $i$ to the center of mass of link $i$

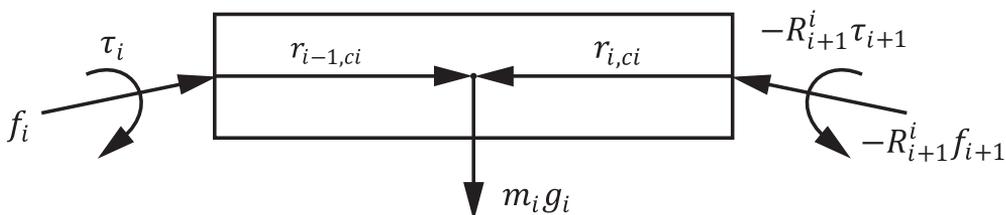


Figure 3.1: Forces and torques acting on a random link

Figure 3.1 shows a random link together with all forces and torques acting on it. By the law of action and reaction,  $f_i$  is the force exerted by link  $i - 1$  on link  $i$ , and  $-f_{i+1}$  is the force exerted by link  $i + 1$  on link  $i$ . According to the definitions above,  $f_i$  is expressed in frame  $i$  while  $-f_{i+1}$  is expressed in frame  $i + 1$ , thus to express both forces in frame  $i$  it is required to multiply the latter with  $R_{i+1}^i$ . The same apply to the torque, again by the law of action and reaction.

When all vectors in Figure 3.1 are expressed in frame  $i$ , the force balance equation based on (3.20) can be stated as

$$\sum_{link} f = ma \quad (3.22)$$

$$f_i - R_{i+1}^i f_{i+1} + m_i g_i = m_i a_{c,i} \quad (3.23)$$

$$f_i = R_{i+1}^i f_{i+1} + m_i a_{c,i} - m_i g_i \quad (3.24)$$

Next, the moment balance equation for the link will be computed, and it is important to note two things. First, the moment exerted by a force  $f$  about a point is given by  $f \times r$ , where  $r$  is the radial vector from the point where the force is applied to the point where the moment is computed. Second, the vector

$m_i g_i$  does not appear in the moment balance since it is applied directly at the center of mass. The moment balance equation based on (3.21) becomes

$$\sum_{link} \tau = \omega \times (I\omega) + I\dot{\omega} \quad (3.25)$$

$$\tau_i - R_{i+1}^i \tau_{i+1} + f_i \times r_{i-1,ci} - (R_{i+1}^i f_{i+1}) \times r_{i,ci} = \omega_i \times (I_i \omega_i) + I_i \alpha_i \quad (3.26)$$

$$\tau_i = R_{i+1}^i \tau_{i+1} - f_i \times r_{i-1,ci} + (R_{i+1}^i f_{i+1}) \times r_{i,ci} + \omega_i \times (I_i \omega_i) + I_i \alpha_i \quad (3.27)$$

The force balance equation is actually a part of the moment balance equation. Solving Equation (3.27) for decreasing  $i$  and substituting (3.24) is the ultimate goal of the formulation, but the solution needs to be expressed only by  $q, \dot{q}, \ddot{q}$  and constant parameters to achieve the general matrix form (3.1). That means it is necessary to find a relation between  $q, \dot{q}, \ddot{q}$  and  $a_{c,i}, \omega_i$  and  $\alpha_i$ . This can be obtained by a recursive procedure of increasing  $i$ .

Since the force and moment equations are expressed with respect to the link attached frame, this also applies to  $a_{c,i}, \omega_i$  and  $\alpha_i$ . However, as a starting point,  $\omega_i$  and  $\alpha_i$  need to be expressed in the inertial frame, and the superscript (0) will be used to denote that. This gives

$$\omega_i^{(0)} = \omega_{i-1}^{(0)} + z_{i-1} \dot{q}_i \quad (3.28)$$

because of the fact that the angular velocity of frame  $i$  equals that of frame  $i-1$  plus the added rotation from joint  $i$ . Using rotation matrices this leads to

$$\omega_i = (R_i^{i-1})^T \omega_{i-1} + b_i \dot{q}_i \quad (3.29)$$

where

$$b_i = (R_i^0)^T R_{i-1}^0 z_0 \quad (3.30)$$

is the rotation of joint  $i$  expressed in frame  $i$ .

For the angular acceleration it is important to note that

$$\alpha_i = (R_i^0)^T \dot{\omega}_i^{(0)} \quad (3.31)$$

which means  $\alpha_i \neq \dot{\omega}_i$ ! By using Newtons Second Law in a rotating frame (see [22], page 342-343), the time derivative of Equation (3.28) becomes

$$\dot{\omega}_i^{(0)} = \dot{\omega}_{i-1}^{(0)} + z_{i-1} \ddot{q}_i + \omega_i^{(0)} \times z_{i-1} \dot{q}_i \quad (3.32)$$

and expressed in frame  $i$  it directly becomes

$$\alpha_i = (R_i^{i-1})^T \alpha_{i-1} + b_i \ddot{q}_i + \omega_i \times b_i \dot{q}_i \quad (3.33)$$

Now it only remains to find an expression for  $a_{c,i}$ . First, the linear velocity of the center of mass of link  $i$  is expressed as

$$v_{c,i}^{(0)} = v_{e,i-1}^{(0)} + \omega_i^{(0)} \times r_{i-1,ci}^{(0)} \quad (3.34)$$

and note that  $r_{i-1,ci}^{(0)}$  is constant in frame  $i$ . Thus

$$a_{c,i}^{(0)} = a_{e,i-1}^{(0)} \times r_{i-1,ci}^{(0)} + \omega_i^{(0)} \times (\omega_i^{(0)} \times r_{i-1,ci}^{(0)}) \quad (3.35)$$

Multiplying with rotation matrices and using the fact that

$$R(a \times b) = (Ra) \times (Rb) \quad (3.36)$$

the final expression for the acceleration of the center of mass of link  $i$ , expressed in frame  $i$ , becomes

$$a_{c,i} = (R_i^{i-1})^T a_{e,i-1} + \dot{\omega}_i \times r_{i-1,ci} + \omega_i \times (\omega_i \times r_{i-1,ci}) \quad (3.37)$$

To find the acceleration of the end of the link,  $r_{i-1,ci}$  is replaced by  $r_{i-1,i}$

$$a_{e,i} = (R_i^{i-1})^T a_{e,i-1} + \dot{\omega}_i \times r_{i-1,i} + \omega_i \times (\omega_i \times r_{i-1,i}) \quad (3.38)$$

This completes the recursive formulation, and the Newton-Euler formulation of an  $n$ -link manipulator can be stated as follows.

1. **Forward recursion:** Start with the initial conditions

$$\omega_0 = \alpha_0 = a_{c,0} = a_{e,0} = 0 \quad (3.39)$$

and solve Equations (3.29), (3.33), (3.38) and (3.37) (in that order) to compute  $\omega_i$ ,  $\alpha_i$  and  $a_{c,i}$  for increasing  $i$  from 1 to  $n$ .

2. **Backward recursion:** Start with the terminal conditions

$$f_{n+1} = \tau_{n+1} = 0 \quad (3.40)$$

and solve Equations (3.24) and (3.27) (in that order) for decreasing  $i$  from  $n$  to 1.

### 3.2.3 Automated Framework

This section introduces an automated framework for deriving the dynamic model of robot manipulators by the Newton-Euler formulation. The framework is given in printed form in Appendix B, and the digital file can be found in the attached ZIP-file. The dynamic model of the IRB 140 is derived by this framework.

The framework is restricted to manipulators with rigid links and only revolute joints, but it can be adjusted to suit other kinds of manipulators if desired. The composition presented is for three degrees of freedom, but due to the recursive procedure it can easily be adjusted to any number of degrees of freedom.

Complex manipulators with several degrees of freedom will result in huge dynamic systems, and it might be desirable to evaluate smaller parts of a system individually. Every element of the inertia matrix, as well as the gravity components and the Coriolis and centrifugal components, are evaluated individually towards the end of the framework.

The Matlab conversion code used in the end can be applied to any expression, and is a simple way to convert Maple code to Matlab code. Note that the joint variable derivatives are not known in the Matlab conversion code. These quantities will have to be redefined as desired in Matlab.

It is optional to define symbolic values if the goal is a less specific model, but note that the more symbolic values and the less defined zeros, the larger dynamic model.

All quantities must be defined according to the definitions and notation introduced in Section 3.2.2. In addition, it is important to add or remove lines following the recursive procedure, according to the how many degrees of freedom there are in the manipulator to be analyzed. A look at the Maple code for the IRB 140 with six degrees of freedom will clarify these adjustments (Appendix C).

Explanations for the parts that has to be adjusted are given below. These parts are grayed out in the framework.

- **Joint variables:** Specify the number of joint variables (degrees of freedom) in the manipulator by adjusting the  $q$ -vector.
- **Link length vectors and vectors to the centers of mass:** Define the link length vectors and the vectors to the centers of mass, which all are independent of configuration.
- **Gravity vector in inertial frame:** Define the direction of gravity in the inertial frame (frame 0).
- **Rotation matrices:** Define the rotation matrices describing rotations from every frame to the subsequent frame.
- **Inertia matrices:** Define the inertia tensors for the links. If the mass distribution of a link is symmetric with respect to the attached frame, then the inertia tensor will be diagonal.



## Chapter 4

# System Description and Dynamic Parameter Estimation

ABB has produced the industrial robot manipulator named IRB 140. Their website [14] presents facts about the manipulator, as well as articles and movies from experiments and from companies using the manipulator.

This chapter is presenting all information about the IRB 140 which is needed to derive the dynamic model by the Newton-Euler formulation. The manipulator comes with a product manual, a product specification [16], and a data sheet (Appendix A and [15]). The manual is not of much interest in this thesis, as it focuses solely on safety, installation and maintenance. What is interesting is the data sheet, which is basically a summary of the product specification, presenting some facts about the structure and performance of the manipulator. The relevant information given in the data sheets are summarized in Section 4.1.

Out of consideration for trade secrets in ABB, the data sheets present a very limited amount of information. Section 4.2 states the these limitations and how they lead to simplified dynamic parameter estimation.

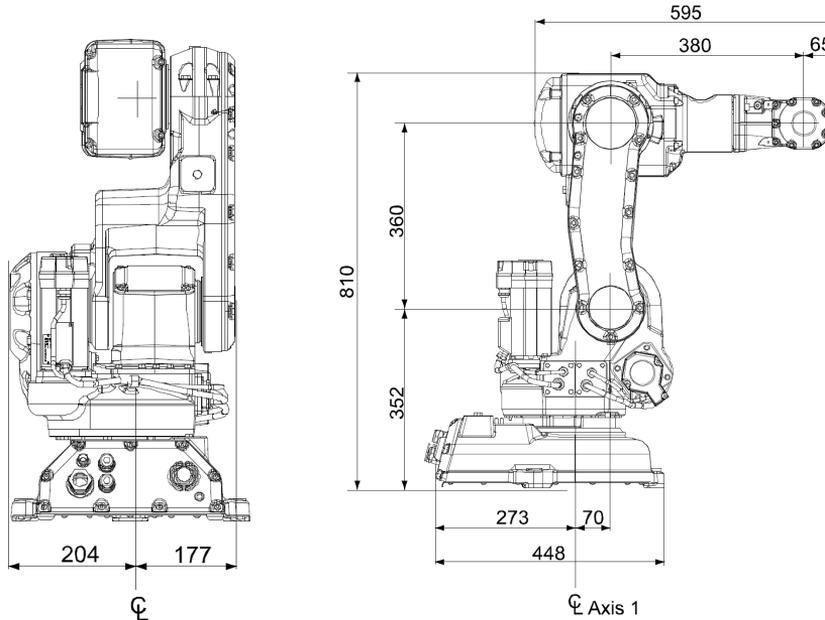
In Section 4.3, a symbolic representation shows how the joints and links can be represented as a serial kinematic chain, and how frames are attached to the links. This representation follows all guidelines described in the previous chapters, and can be said to lay the foundation for the whole dynamic model.

Parameter estimation is carried out in Section 4.4.

### 4.1 Information from Data Sheets

As mentioned in Section 1.2, the manipulator has a total of six revolute joints that are controlled by AC-motors, hence six degrees of freedom (6 DOF). The

total mass including the base and without a payload is 98 kg, and the mass of the payload alone must not exceed 6 kg. Some applicable link dimensions are given in Figure 4.1 (lengths in millimeters).



(a) View of the manipulator from the back

(b) View of the manipulator from the side

Figure 4.1: View of the manipulator from the back and side [16]

## 4.2 Limitations

It is not possible to derive an accurate dynamic model for the IRB 140 with the limited information available in the data sheets. No dynamic parameters for the links are given, and as explained in Section 3.1, these parameters are indeed a demanding task to estimate. The masses of the links could have been identified by dismantling the manipulator and weigh them one by one, but this would have been a comprehensive task by itself. Besides, doing this would not come in particularly useful anyway, unless thorough experiments on estimating the inertia parameters and centers of mass were to be performed as well.

Performing dynamic parameter identification of the IRB 140 is an interesting and challenging task, but would require the possibility to measure  $q, \dot{q}, \ddot{q}$ , or knowledge about alternative identification methods like for example CAD modeling (would again require detailed information about the shape and materials of the manipulator). ABB does not give their customers access to measure

the states of the robot manipulators, not even access to the controller parameters. Consequently, the dynamic parameters in the model have been estimated quite roughly. The estimation is based on intuitive guesses, with the purpose of creating a simple model which still represents the IRB 140 as good as possible.

### 4.3 Modeling Set-up

The IRB 140 can be interpreted in such a way that the first three degrees of freedom make up an elbow manipulator, and the last three degrees of freedom is a spherical wrist attached to the end of the arm. This spherical wrist alone is built up by three single degree-of-freedom revolute joints, where the rotation axes intersect in the wrist center point. Thus the two links in between will have zero length and zero mass.

Examining the manipulator closer, it is discovered that some freedom is given to the choice of how to model joint 4. Actually, modeling the last three joints as a spherical wrist is not the desired choice, because the two links in between (link 4 and 5) do not have zero length and mass. To compensate for this, it is found convenient to interpret the manipulator such that joint 3 and 4 has their center point in common, and joint 5 and 6 has their center point in common. In that case it is link 3 and link 5 which is modeled with zero length and mass. Figure 4.2 shows a symbolic representation of the manipulator by this interpretation, including how the frames have been attached to the links.

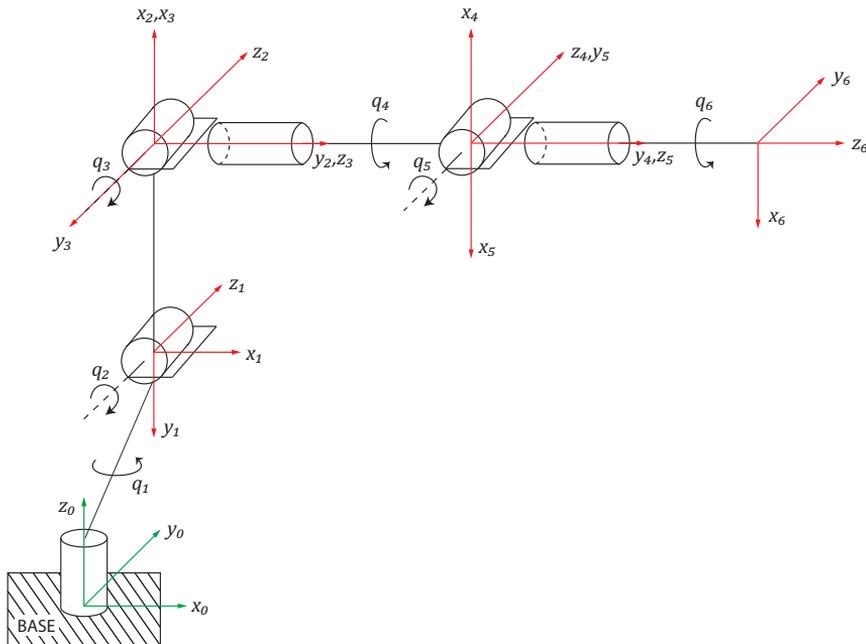


Figure 4.2: Symbolic representation of the IRB 140

All rotation matrices can be calculated as products of basic rotations about the  $z$ -axis and the  $x$ -axis, where these basic rotation matrices are defined in general form as

$$R_{z,\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (4.1)$$

where  $\theta$  is the rotation angle. Observing Figure 4.2, it is quite straight forward to calculate rotation matrices for the manipulator by substituting  $q$  and the constant rotations for  $\theta$ . Note the simplifications due to all constant rotations being multiples of  $\frac{\pi}{2}$ . The result becomes

$$R_1^0 = \begin{bmatrix} \cos(q_1) & 0 & -\sin(q_1) \\ \sin(q_1) & 0 & \cos(q_1) \\ 0 & -1 & 0 \end{bmatrix} \quad (4.2)$$

$$R_2^1 = \begin{bmatrix} \cos(q_2 - \frac{\pi}{2}) & -\sin(q_2 - \frac{\pi}{2}) & 0 \\ \sin(q_2 - \frac{\pi}{2}) & \cos(q_2 - \frac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

$$R_3^2 = \begin{bmatrix} \cos(q_3) & 0 & -\sin(q_3) \\ \sin(q_3) & 0 & \cos(q_3) \\ 0 & -1 & 0 \end{bmatrix} \quad (4.4)$$

$$R_4^3 = \begin{bmatrix} \cos(q_4) & 0 & \sin(q_4) \\ \sin(q_4) & 0 & -\cos(q_4) \\ 0 & 1 & 0 \end{bmatrix} \quad (4.5)$$

$$R_5^4 = \begin{bmatrix} \cos(q_5 + \pi) & 0 & \sin(q_5 + \pi) \\ \sin(q_5 + \pi) & 0 & -\cos(q_5 + \pi) \\ 0 & 1 & 0 \end{bmatrix} \quad (4.6)$$

$$R_6^5 = \begin{bmatrix} \cos(q_6) & -\sin(q_6) & 0 \\ \sin(q_6) & \cos(q_6) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

Further multiplications by a recursive procedure yield

$$R_2^0 = R_1^0 R_2^1 \quad (4.8)$$

$$R_3^0 = R_2^0 R_3^2 \quad (4.9)$$

$$R_4^0 = R_3^0 R_4^3 \quad (4.10)$$

$$R_5^0 = R_4^0 R_5^4 \quad (4.11)$$

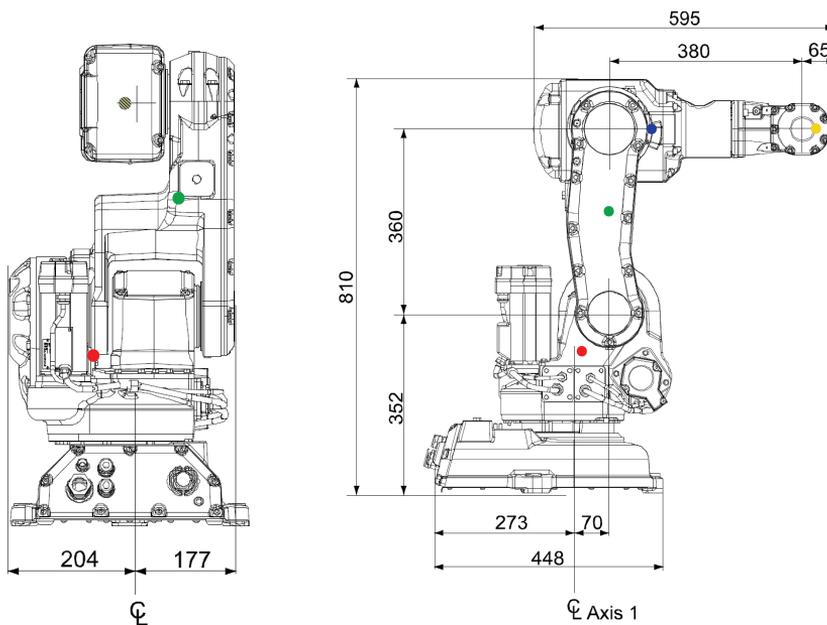
$$R_6^0 = R_5^0 R_6^5 \quad (4.12)$$

## 4.4 Parameter Estimation

This section describes how the dynamic parameters are estimated. It is mentioned in Section 4.2 that the parameters are estimated quite roughly. Still they

should be close enough to the real unknown parameters that simulations show a behavior that is somewhat in accordance to the behavior of a perfect model.

The centers of mass of the four links have been estimated by studying the manipulator thoroughly, assuming the links have uniform mass density. Figure 4.3 shows the estimated centers of mass with colored dots. Link 1 has a red dot, link 2 has a green dot, link 4 has a blue dot, and link 6 has a yellow dot. Note that viewing from the back in Figure 4.3(a), link 4 and 6 have their centers of mass along the same line perpendicular to the paper.



(a) The centers of mass from the back

(b) The centers of mass from the side

Figure 4.3: The centers of mass from the back and side [16]

Vectors between the origins of the frames are defined precisely by the dimensions in Figure 4.3. Vectors from the origins of the frames to the centers of mass are calculated by first computing the scale of the figure, and then multiplying the scale with the lengths measured by a ruler. The clever way of attaching frames to the links in the Newton-Euler formulation make all length vectors independent of the configuration of the manipulator. The results are given below

based on the definitions described in Section 3.2.2 (lengths in meters).

$$r_{0,c1} = [0.014 \quad -0.264 \quad 0.067]^T \quad (4.13)$$

$$r_{1,c2} = [0.201 \quad 0 \quad -0.070]^T \quad (4.14)$$

$$r_{2,c3} = [0 \quad 0 \quad 0]^T \quad (4.15)$$

$$r_{3,c4} = [0 \quad 0.080 \quad 0]^T \quad (4.16)$$

$$r_{4,c5} = [0 \quad 0 \quad 0]^T \quad (4.17)$$

$$r_{5,c6} = [0 \quad 0 \quad 0.029]^T \quad (4.18)$$

$$r_{0,1} = [0.070 \quad -0.352 \quad 0]^T \quad (4.19)$$

$$r_{1,2} = [0.360 \quad 0 \quad 0]^T \quad (4.20)$$

$$r_{2,3} = [0 \quad 0 \quad 0]^T \quad (4.21)$$

$$r_{3,4} = [0 \quad 0.380 \quad 0]^T \quad (4.22)$$

$$r_{4,5} = [0 \quad 0 \quad 0]^T \quad (4.23)$$

$$r_{5,6} = [0 \quad 0 \quad 0.065]^T \quad (4.24)$$

Estimating the inertia parameters are definitely the most difficult task. The irregular shapes of the links makes it highly complicated to come up with realistic parameters without performing some kind of identification. As a fair simplification the links are modeled as cylindrical links with uniform mass density, where the center of mass of each link is the geometric center of the cylinder. Figure 4.4 shows an example of how this simplification can be applied on link 2.

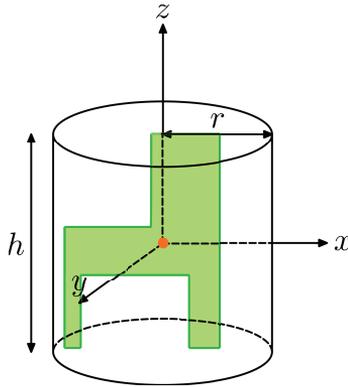


Figure 4.4: Example of link 2 modeled as a cylinder

The green figure illustrates link 2 viewed from the back (as can be recognized in Figure 4.3(a)), and the orange dot is the center of mass. The inertia tensor

of such a cylinder is shown in [4] to be

$$I = \begin{bmatrix} \frac{1}{12}mh^2 + \frac{1}{4}mr^2 & 0 & 0 \\ 0 & \frac{1}{12}mh^2 + \frac{1}{4}mr^2 & 0 \\ 0 & 0 & \frac{1}{2}mr^2 \end{bmatrix} \quad (4.25)$$

where  $m$  is the mass,  $r$  is the radius and  $h$  is the height of the cylinder. The cross products are identically zero such that the inertia tensor becomes a diagonal matrix in its principal axis form.

Determining the the mass, radius and height of the cylinders is kind of a constrained task, where the constraints are that the total mass must be 98 kg (including the base), and that the radius and height of the cylinders match the dimensions of the manipulator given in Figure 4.3. Like the actual links do, the cylinders will also overlap each other since the centers of mass are not geometrically right in between two frames, and it was assumed uniform mass density.

It is fair to believe that the mass density of every link is approximately equal. The links are constructed of a shell of metal with components such as motors, gearboxes, cables and belts on the inside. In addition, large proportions of the total volume is just air in between these components. By a trial-and-error approach, the masses, radii and heights was eventually found to match the physical shape of the manipulator using a mutual mass density of  $1500 \frac{kg}{m^3}$ . The parameter values are given in Table 4.1, where the missing mass of 23 kg is allocated the manipulator base. To make a comparison, the mass density of steel is  $7850 \frac{kg}{m^3}$  according to [12]. That is for massive steel, such that assuming a mass density of the links of about the fifth the mass density for steel seems satisfying.

Link	Mass	Radius	Height
1	27 kg	0.147 m	0.264 m
2	22 kg	0.108 m	0.402 m
3	-	-	-
4	25 kg	0.094 m	0.600 m
5	-	-	-
6	1 kg	0.054 m	0.072 m

Table 4.1: Cylinder parameters

Note that the orientation of the attached frame determines the coordination of the principal moments of inertia. With regard to this, the resulting inertia tensors are calculated by substituting in (4.25). The result becomes

$$I_1 = \begin{bmatrix} \frac{1}{12}m_1h_1^2 + \frac{1}{4}m_1r_1^2 & 0 & 0 \\ 0 & \frac{1}{2}m_1r_1^2 & 0 \\ 0 & 0 & \frac{1}{12}m_1h_1^2 + \frac{1}{4}m_1r_1^2 \end{bmatrix} \quad (4.26)$$

$$I_2 = \begin{bmatrix} \frac{1}{2}m_2r_2^2 & 0 & 0 \\ 0 & \frac{1}{12}m_2h_2^2 + \frac{1}{4}m_2r_2^2 & 0 \\ 0 & 0 & \frac{1}{12}m_2h_2^2 + \frac{1}{4}m_2r_2^2 \end{bmatrix} \quad (4.27)$$

$$I_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.28)$$

$$I_4 = \begin{bmatrix} \frac{1}{12}m_4h_4^2 + \frac{1}{4}m_4r_4^2 & 0 & 0 \\ 0 & \frac{1}{2}m_4r_4^2 & 0 \\ 0 & 0 & \frac{1}{12}m_4h_4^2 + \frac{1}{4}m_4r_4^2 \end{bmatrix} \quad (4.29)$$

$$I_5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.30)$$

$$I_6 = \begin{bmatrix} \frac{1}{12}m_6h_6^2 + \frac{1}{4}m_6r_6^2 & 0 & 0 \\ 0 & \frac{1}{12}m_6h_6^2 + \frac{1}{4}m_6r_6^2 & 0 \\ 0 & 0 & \frac{1}{2}m_6r_6^2 \end{bmatrix} \quad (4.31)$$

# Chapter 5

## The Dynamic Model

The recursive Newton-Euler formulation was derived in general in Chapter 3. Then in Chapter 4, a system description of the IRB 140 was presented. A symbolic representation showed how to interpret the manipulator as a kinematic chain, corresponding rotation matrices were calculated, and parameter estimations were performed. To derive the dynamic model of the manipulator, the rotation matrices and parameters from the system description are substituted in the equations for the Newton-Euler formulation of an  $n$ -link manipulator. The forward and backward recursions are carried out in Section 5.1 and 5.2 respectively, and comments on the results are found in Section 5.3.

With its six degrees of freedom, even the simplified system is quite complex. Actually, the final torque equations in the model are so huge that they do not even suit to be shown in this text. Therefore, to let this chapter be clear and easy to follow, all equations are kept in their recursive form. Appendix C shows how the model has been computed in Maple by adjusting the framework presented in Section 3.2.3 and Appendix B.

### 5.1 Forward Recursion

The forward recursion describes the linear and angular motion of the links, starting with link 1 and ending with link 6. The algorithm is described in Section 3.2.2, and it is just a matter of substituting in the general equations for an  $n$ -link manipulator.

As a part of the forward recursion it is necessary to compute  $b_i$ , the axis of rotation for each joint  $i$  expressed in frame  $i$ . Prior to the recursions, these computations will be carried out right away for all joints to emphasize a great advantage of the Newton-Euler formulation. The rotation axis in frame 0 is given directly as

$$z_0 = [0 \ 0 \ 1]^T \tag{5.1}$$

and then the rotation axes for the joints are computed by Equation (3.30) as

$$b_1 = (R_1^0)^T z_0 = [0 \quad -1 \quad 0]^T \quad (5.2)$$

$$b_2 = (R_2^0)^T R_1^0 z_0 = [0 \quad 0 \quad 1]^T \quad (5.3)$$

$$b_3 = (R_3^0)^T R_2^0 z_0 = [0 \quad -1 \quad 0]^T \quad (5.4)$$

$$b_4 = (R_4^0)^T R_3^0 z_0 = [0 \quad 1 \quad 0]^T \quad (5.5)$$

$$b_5 = (R_5^0)^T R_4^0 z_0 = [0 \quad 1 \quad 0]^T \quad (5.6)$$

$$b_6 = (R_6^0)^T R_5^0 z_0 = [0 \quad 0 \quad 1]^T \quad (5.7)$$

Due to the coupled kinematics, these rotation axes will normally be functions of  $q$  just like the rotation matrices. They will depend on how the coordinate frames are defined, and therefore directly influence the efficiency of the Newton-Euler formulation. By inspecting how the frames are defined in Figure 4.2, it can be seen that when looking from frame  $i$  into frame  $i-1$ , the angular velocity  $\omega_i$  does not depend on  $q_i$  itself, but completely on the axis of rotation. Consequently the rotation axes  $b_i$  are not depending on  $q$ .

## Link 1

The initial conditions are

$$\omega_0 = \alpha_0 = a_{c,0} = a_{e,0} = 0 \quad (5.8)$$

Angular velocity and acceleration are calculated from Equation (3.29) and (3.33) respectively, and becomes

$$\omega_1 = b_1 \dot{q}_1 \quad (5.9)$$

$$\alpha_1 = b_1 \ddot{q}_1 + \omega_1 \times b_1 \dot{q}_1 \quad (5.10)$$

Acceleration of the end of the link and the center of the link are calculated from Equation (3.38) and (3.37) respectively, and becomes

$$a_{e,1} = \dot{\omega}_1 \times r_{0,1} + \omega_1 \times (\omega_1 \times r_{0,1}) \quad (5.11)$$

$$a_{c,1} = \dot{\omega}_1 \times r_{0,c1} + \omega_1 \times (\omega_1 \times r_{0,c1}) \quad (5.12)$$

## Link 2

Substituting in the same equations as for link 1, the angular velocity and acceleration becomes

$$\omega_2 = (R_2^1)^T \omega_1 + b_2 \dot{q}_2 \quad (5.13)$$

$$\alpha_2 = (R_2^1)^T \alpha_1 + b_2 \ddot{q}_2 + \omega_2 \times b_2 \dot{q}_2 \quad (5.14)$$

Acceleration of the center and the end of the link becomes

$$a_{e,2} = (R_2^1)^T a_{e,1} + \dot{\omega}_2 \times r_{1,2} + \omega_2 \times (\omega_2 \times r_{1,2}) \quad (5.15)$$

$$a_{c,2} = (R_2^1)^T a_{c,1} + \dot{\omega}_2 \times r_{1,c2} + \omega_2 \times (\omega_2 \times r_{1,c2}) \quad (5.16)$$

### Link 3

Substituting for link 3 gives

$$\omega_3 = (R_3^2)^T \omega_2 + b_3 \dot{q}_3 \quad (5.17)$$

$$\alpha_3 = (R_3^2)^T \alpha_2 + b_3 \ddot{q}_3 + \omega_3 \times b_3 \dot{q}_3 \quad (5.18)$$

$$a_{e,3} = (R_3^2)^T a_{e,2} \quad (5.19)$$

$$a_{c,3} = (R_3^2)^T a_{c,2} \quad (5.20)$$

### Link 4

Substituting for link 4 gives

$$\omega_4 = (R_4^3)^T \omega_3 + b_4 \dot{q}_4 \quad (5.21)$$

$$\alpha_4 = (R_4^3)^T \alpha_3 + b_4 \ddot{q}_4 + \omega_4 \times b_4 \dot{q}_4 \quad (5.22)$$

$$a_{e,4} = (R_4^3)^T a_{e,3} + \dot{\omega}_4 \times r_{3,4} + \omega_4 \times (\omega_4 \times r_{3,4}) \quad (5.23)$$

$$a_{c,4} = (R_4^3)^T a_{c,3} + \dot{\omega}_4 \times r_{3,c4} + \omega_4 \times (\omega_4 \times r_{3,c4}) \quad (5.24)$$

### Link 5

Substituting for link 5 gives

$$\omega_5 = (R_5^4)^T \omega_4 + b_5 \dot{q}_5 \quad (5.25)$$

$$\alpha_5 = (R_5^4)^T \alpha_4 + b_5 \ddot{q}_5 + \omega_5 \times b_5 \dot{q}_5 \quad (5.26)$$

$$a_{e,5} = (R_5^4)^T a_{e,4} \quad (5.27)$$

$$a_{c,5} = (R_5^4)^T a_{c,4} \quad (5.28)$$

### Link 6

Substituting for link 6 gives

$$\omega_6 = (R_6^5)^T \omega_5 + b_6 \dot{q}_6 \quad (5.29)$$

$$\alpha_6 = (R_6^5)^T \alpha_5 + b_6 \ddot{q}_6 + \omega_6 \times b_6 \dot{q}_6 \quad (5.30)$$

$$a_{c,6} = (R_6^5)^T a_{c,5} + \dot{\omega}_6 \times r_{5,c6} + \omega_6 \times (\omega_6 \times r_{5,c6}) \quad (5.31)$$

Note that there is no need to compute  $a_{e,6}$  because  $a_{e,i}$  is only used to compute  $a_{e,i+1}$  (and there is no link 7).

## 5.2 Backward Recursion

The backward recursion calculates the forces and joint torques acting on the links, starting with link 6 and ending with link 1. Determining the joint torques is the ultimate goal of the Newton-Euler formulation, because the torques are

the externally applied input to the model. As for the forward recursion, the algorithm is described in Section 3.2.2 and it is just a matter of substituting in the general equations for an  $n$ -link manipulator. Note that the force equation includes the gravity vector. This gravity vector differs for each link, but can easily be calculated with the use of rotation matrices as shown in the recursions below.

### Link 6

The terminal conditions are

$$f_7 = \tau_7 = 0 \quad (5.32)$$

The gravity vector becomes

$$g_6 = (R_6^0)^T g_0 \quad (5.33)$$

where  $g_0$  is the gravity vector in the inertial frame defined as

$$g_0 = [0 \quad 0 \quad -g]^T \quad (5.34)$$

The force and joint torque exerted on the link are calculated from Equation (3.24) and (3.27) respectively, and becomes

$$f_6 = m_6 a_{c,6} - m_6 g_6 \quad (5.35)$$

$$\tau_6 = -f_6 \times r_{5,c6} + \omega_6 \times (I_6 \omega_6) + I_6 \alpha_6 \quad (5.36)$$

### Link 5

Following the same procedure as for link 6, the gravity vector for link 5 becomes

$$g_5 = (R_5^0)^T g_0 \quad (5.37)$$

and the force and joint torque equations become

$$f_5 = R_6^5 f_6 \quad (5.38)$$

$$\tau_5 = R_6^5 \tau_6 + \omega_5 \times (I_5 \omega_5) + I_5 \alpha_5 \quad (5.39)$$

### Link 4

Substituting for link 4 gives

$$g_4 = (R_4^0)^T g_0 \quad (5.40)$$

$$f_4 = R_5^4 f_5 + m_4 a_{c,4} - m_4 g_4 \quad (5.41)$$

$$\tau_4 = R_5^4 \tau_5 - f_4 \times r_{3,c4} + R_5^4 f_5 \times r_{4,c4} + \omega_4 \times (I_4 \omega_4) + I_4 \alpha_4 \quad (5.42)$$

### Link 3

Substituting for link 3 gives

$$g_3 = (R_3^0)^T g_0 \quad (5.43)$$

$$f_3 = R_4^3 f_4 \quad (5.44)$$

$$\tau_3 = R_4^3 \tau_4 + \omega_3 \times (I_3 \omega_3) + I_3 \alpha_3 \quad (5.45)$$

### Link 2

Substituting for link 2 gives

$$g_2 = (R_2^0)^T g_0 \quad (5.46)$$

$$f_2 = R_3^2 f_3 + m_2 a_{c,2} - m_2 g_2 \quad (5.47)$$

$$\tau_2 = R_3^2 \tau_3 - f_2 \times r_{1,c2} + R_3^2 f_3 \times r_{2,c2} + \omega_2 \times (I_2 \omega_2) + I_2 \alpha_2 \quad (5.48)$$

### Link 1

Substituting for link 1 gives

$$g_1 = (R_1^0)^T g_0 \quad (5.49)$$

$$f_1 = R_2^1 f_2 + m_1 a_{c,1} - m_1 g_1 \quad (5.50)$$

$$\tau_1 = R_2^1 \tau_2 - f_1 \times r_{0,c1} + R_2^1 f_2 \times r_{1,c1} + \omega_1 \times (I_1 \omega_1) + I_1 \alpha_1 \quad (5.51)$$

## 5.3 Comments

The results in this chapter are interesting and verifies why the Newton-Euler formulation often is the preferred choice for manipulators with many degrees of freedom. The recursive algorithm is easy to implement and consequently there are small chances of doing any mistakes in the derivation. Strange behavior of the model can mostly be connected to the preparations such as the set-up of the kinematic chain and the frames, rotation matrices, vector definitions and inertia tensors.

Note that even though link 3 and 5 have zero length and mass, they still have to be considered in the recursions. The Newton-Euler formulation is based on a kinematic chain with only single degree-of-freedom joints, such that  $n$  degrees of freedom always lead to  $n$  steps in each recursion. However, some terms in the expressions for link 3 and 5 are canceled out.

As mentioned in Section 3.1.1, the Newton-Euler formulation and the Euler-Lagrange formulation can provide different insights. One interesting insight in the Newton-Euler formulation comes from the final joint torque vectors in the backward recursion. All joints in the kinematic chain are single degree-of-freedom joints, such that the torques applied are scalars about the rotation axes computed in Equations (5.2)-(5.7). The other two elements of the torque vectors

can be explained as follows. When applying torque to any of the joints, this will also generate torque components about the other axes of the joints due to the coupled kinematics in the system. These torque components are not included in the dynamic model because they do not induce motion (not affecting  $q$ ), but still it is valuable information about the physics of the manipulator. If the joints in the manipulator are not constructed to physically resist these torque quantities, the joints will break.

Although utilizing the Newton-Euler formulation appears to be quite easy, the complexity of the resulting model should be emphasized. The basic idea behind recursion is that the solution to a problem depends on solutions to smaller instances on the same problem. The backward recursion of link 1 depends on the backward recursion of link 2, which depends of the backwards recursion of link 3, and so on. All in all the backward recursion of link 1 is directly dependent on all 11 steps back to the forward recursion of link 1. Thus it should not be a surprise that calculating  $\tau_1$  from Equation (5.51) results in a huge vector. To demonstrate the complexity of the system, the final expression for  $\tau_1$  (only the component that induces motion) can be found in Appendix C. All details are available to the reader by exploring the model that can be found in the attached ZIP-file.

## Chapter 6

# Simulations and Results

This chapter deals with simulations of the dynamic model in open loop and closed loop. Note that the goal is to perform simulations to prove the validity of the model, not to optimize a control system for a specific job task.

The simulation structure in Simulink and the connection to Matlab is described in Section 6.1. In Section 6.2 the second-order model is reduced to an equivalent first-order model which is needed to perform simulations in Simulink.

The open loop case is presented in Section 6.3 and 6.4. First the model is driven with desired torque to check for open loop stability, and then energy properties are investigated. The closed loop case in Section 6.5 presents a mathematical proof of global asymptotic stability with PD control of a system model in the form (3.1). Then the proof is verified for the model of the IRB 140.

### 6.1 Simulation Structure

In Simulink a so-called *Level-2 Matlab S-Function* is used to implement the dynamics. This is a block with multiple input and output ports where input 1 is the state vector, input 2 is the applied torque vector, and the output is the vector of state derivatives. For each time step in the simulation the updated vector of state derivatives is computed from the new inputs. The contents of the *Level-2 Matlab S-Function* block is the dynamic model in reduced form (see Section 6.2). The Dormand-Prince method *ode45* with relative tolerance  $10^{-6}$  is used for solving the differential equations.

For each time step in the simulation, the state vector is sent from Simulink to the Matlab interface through a *To Workspace* block. That makes it possible to present the results graphically, and use the states to compute kinetic and potential energy.

## 6.2 Reduced System Order

As described in Section 3.1, the dynamic model can be written on matrix form as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \quad (6.1)$$

To simulate the system in Simulink it is necessary to express it in the first-order nonlinear form

$$\dot{x} = f(x, u) \quad (6.2)$$

where  $x$  is the state vector and  $u$  is the torque vector.

The first step is to rearrange the terms in (6.1) to get

$$\ddot{q} = M^{-1}(-C\dot{q} - g + u) \quad (6.3)$$

where it is assumed that the inertia matrix  $M$  is invertible. The inertia matrix is the main factor of the kinetic energy expression  $\frac{1}{2}\dot{q}^T M(q)\dot{q}$  (see Section 6.4.1). Positive definiteness of  $M$  is seen directly by the fact that the kinetic energy is always nonnegative, and is zero if and only if all the joint velocities are zero. Thus,  $M$  is invertible and Equation (6.3) is valid.

The second step is to reduce the system from 6 second-order equations to 12 first-order equations. Defining

$$x_1 = q_1, \quad x_2 = \dot{x}_1 = \dot{q}_1 \quad (6.4)$$

$$x_3 = q_2, \quad x_4 = \dot{x}_3 = \dot{q}_2 \quad (6.5)$$

$$\vdots \quad \quad \quad \vdots$$

$$x_{11} = q_6, \quad x_{12} = \dot{x}_{11} = \dot{q}_6 \quad (6.6)$$

the dynamic system can be expressed in the form (6.2) as

$$\dot{x}_1 = x_2 \quad (6.7)$$

$$\dot{x}_2 = f_2(x, u) \quad (6.8)$$

$$\dot{x}_3 = x_4 \quad (6.9)$$

$$\dot{x}_4 = f_4(x, u) \quad (6.10)$$

$$\dot{x}_5 = x_6 \quad (6.11)$$

$$\dot{x}_6 = f_6(x, u) \quad (6.12)$$

$$\dot{x}_7 = x_8 \quad (6.13)$$

$$\dot{x}_8 = f_8(x, u) \quad (6.14)$$

$$\dot{x}_9 = x_{10} \quad (6.15)$$

$$\dot{x}_{10} = f_{10}(x, u) \quad (6.16)$$

$$\dot{x}_{11} = x_{12} \quad (6.17)$$

$$\dot{x}_{12} = f_{12}(x, u) \quad (6.18)$$

where  $f_{2i}(x, u)$  is the expression for  $\ddot{q}_i$  in Equation (6.3) for  $i = 1, 2, \dots, 6$  and substituting with  $x$  for  $q$  and  $\dot{q}$ .

Note that this first-order model is only how the dynamics are implemented in Simulink and Matlab. All figures and text for the rest of this chapter will refer to the original second-order system with  $q$  as the state vector.

## 6.3 Open Loop with Desired Torque

In open loop there is no feedback from the system output. In other words, no information about the joint variables and its derivatives is available when computing the input torque. Figure 6.1 shows the open loop model in Simulink, where the block called *irb140* contains all the dynamics.

Due to the excitation of gravity on the links being dependent on the joint variables, it is quite intuitive that controlling the system in open loop is impossible. The behavior of the system can be studied by driving the system with the desired torque, that is the constant torque derived when substituting in the dynamic equations for the desired joint variables and derivatives. If  $\dot{q}_{des} = 0$ , this control torque can be explained as the constant torque which is needed to keep the manipulator steady in the desired position.

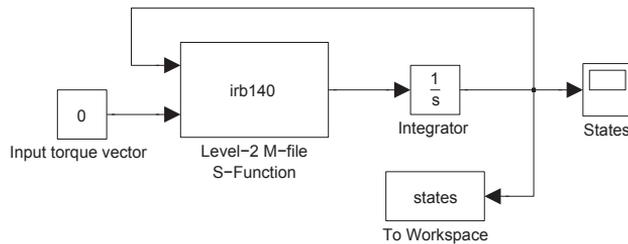


Figure 6.1: Open loop simulink model

### 6.3.1 Simulations

The desired position and velocity are set to

$$q_{des} = [0 \quad \frac{\pi}{2} \quad -\frac{\pi}{2} \quad 0 \quad 0 \quad 0]^T \quad (6.19)$$

$$\dot{q}_{des} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (6.20)$$

which is the position when the manipulator arm is stretched out to the maximum in the  $x_0$ -direction (see Figure 4.2). By substituting the desired position and velocity in the dynamic equations ( $\dot{q}_{des}$  gives  $\ddot{q}_{des} = 0$ ), the control torque

becomes

$$u_{des} = \begin{bmatrix} 0 \\ -2.409 \cdot g - 13.782 \cdot g \\ -2.409 \cdot g \\ 0 \\ -0.029 \cdot g \\ 0 \end{bmatrix} \quad (6.21)$$

From an intuitive perspective this control torque is as expected. To keep the manipulator steady in the chosen desired position, joint 2, 3 and 5 will have to be actuated to compensate for the gravity, based on the law of action and reaction. Joint 1, 4 and 6 will not be influenced by gravity as long as  $\dot{q} = 0$ , and is therefore given zero control torque.

Four simulations, each with different initial conditions, shows the behavior of the system when applied this control torque. The gravity acceleration is set to  $g = 9.81 \frac{m}{s^2}$ , the gravity of earth.

### Simulation 1

Initial conditions are set to

$$q_{init} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.22)$$

$$\dot{q}_{init} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.23)$$

The response shown in Figure 6.2.

### Simulation 2

Initial conditions are set to

$$q_{init} = [0 \ \pi \ -\frac{\pi}{2} \ 0 \ 0 \ 0]^T \quad (6.24)$$

$$\dot{q}_{init} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.25)$$

The response is shown in Figure 6.3.

### Simulation 3

Initial conditions are set to

$$q_{init} = [0 \ \frac{\pi}{2} \ -\frac{\pi}{2} \ 0 \ 0 \ 0]^T \quad (6.26)$$

$$\dot{q}_{init} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.27)$$

The response is shown in Figure 6.4.

### Simulation 4

Initial conditions are set to

$$q_{init} = [0 \quad \frac{\pi}{2} + 0.05 \quad -\frac{\pi}{2} \quad 0 \quad 0 \quad 0]^T \quad (6.28)$$

$$\dot{q}_{init} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (6.29)$$

The response is shown in Figure 6.5.

#### 6.3.2 Comments

Simulation 1 and 2 show a clearly unstable behavior when attempting to control the system to a desired position that is not in immediate proximity to the initial conditions. As mentioned, this is just as expected because the excitation of gravity on the links is dependent on the joint variables.

In simulation 3, the initial conditions are equal to the desired position and velocity, and the graph is showing the expected response. The joints are actuated exactly as required to compensate for the gravity and to keep the manipulator steady in the desired state. An interesting result is observed in simulation 4 where the initial position is only 0.05 radians (in joint 2) away from the desired position. The first four seconds looks exactly as in simulation 3, but suddenly the system collapses into completely unstable behavior like in simulation 1 and 2.

The conclusion corresponds to what was assumed in advance of the simulations. The behavior of the system is unstable, and just the slightest disturbance in the system leads to a completely uncontrollable motion because the gravity on the links is dependent on the joint variables, and the input is computed without observing the output. The system requires feedback controllers to be stabilized.

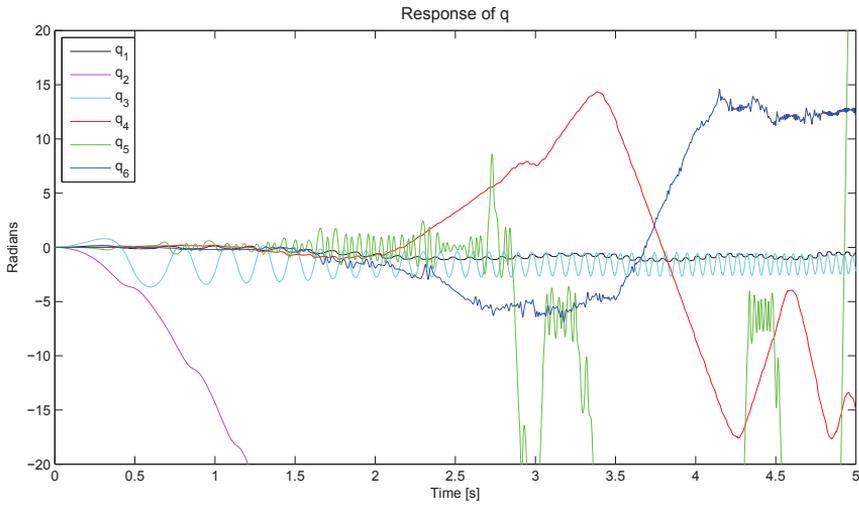


Figure 6.2: Open loop response with desired torque, simulation 1

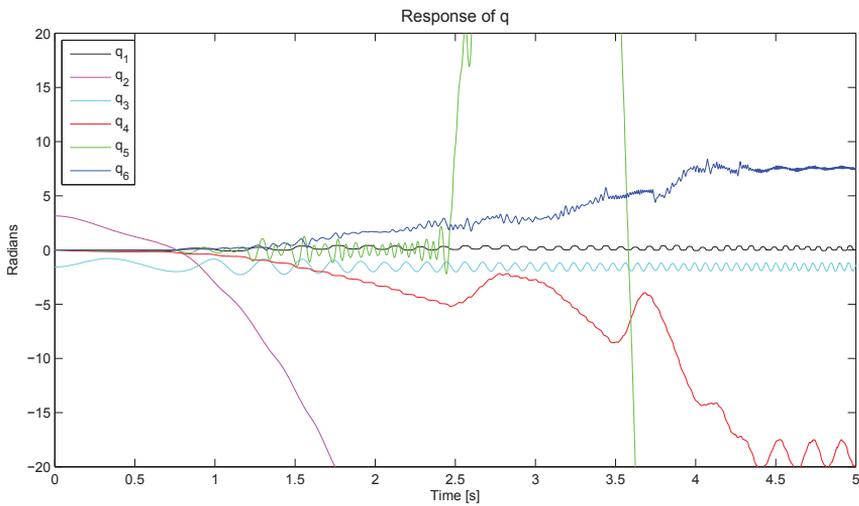


Figure 6.3: Open loop response with desired torque, simulation 2

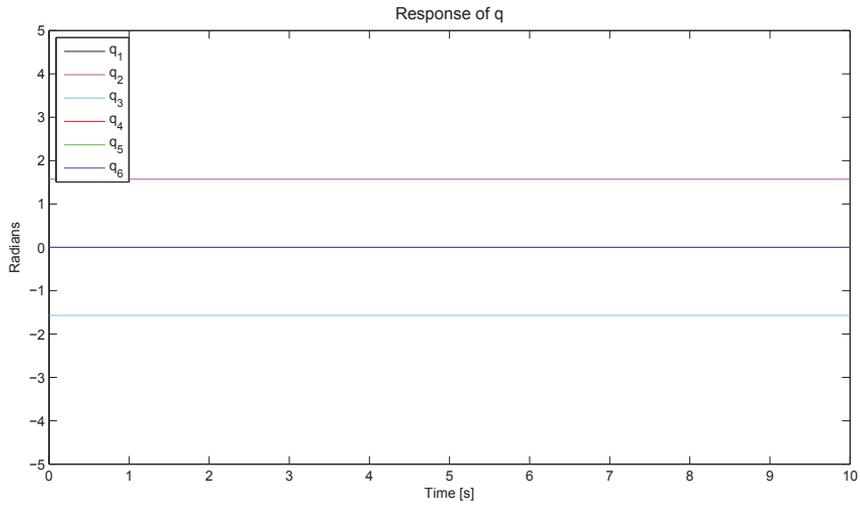


Figure 6.4: Open loop response with desired torque, simulation 3

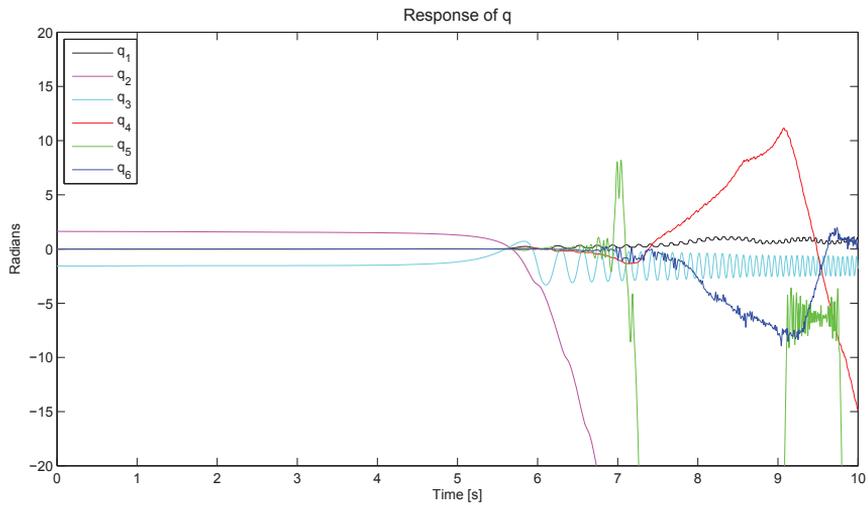


Figure 6.5: Open loop response with desired torque, simulation 4

## 6.4 Energy Properties

The internal energy of the system is the sum of kinetic and potential energy. It increases as energy is supplied and decreases as energy dissipates. The only supplied energy in the model is the joint torque from the AC-motors, and no dissipations are taken into account. As a result, the internal energy of the system should be conserved during any motion as long as no torque is applied. Note however that in a perfect model there would be dissipations from for instance joint friction.

### 6.4.1 Kinetic Energy

The kinetic energy of a rigid object is the sum of two terms, the translational kinetic energy obtained by concentrating the entire mass of the object at the center of mass, and rotational kinetic energy of the body about the center of mass. Mathematically, it is given as

$$K = \frac{1}{2}mv^T v + \frac{1}{2}\omega^T I\omega \quad (6.30)$$

where  $m$  is the mass of the object,  $v$  and  $\omega$  are the linear and angular velocity vectors respectively, and  $I$  is the inertia tensor (see [20]).

For an  $n$ -link manipulator, the overall kinetic energy is the sum of the kinetic energy for each link. It can be shown that the overall kinetic energy equals

$$K = \frac{1}{2}\dot{q}^T M(q)\dot{q} \quad (6.31)$$

where  $M$  is the configuration dependent inertia matrix recognized from the general model in Equation (6.1). This is described in more detail in [20] on page 250-254. Since the dynamic model is already derived in Chapter 5, it is straightforward to compute the kinetic energy expression by substitution of  $M$  in Equation (6.31). The resulting expression is huge and can be found in Appendix C.

### 6.4.2 Potential Energy

The overall potential energy of an  $n$ -link manipulator is the sum of the potential energy for each link. Given that the links are rigid, the only source of potential energy is gravity. The overall potential energy can then be calculated as

$$P = \sum_{i=1}^n P_i = \sum_{i=1}^n m_i g h_{ci} \quad (6.32)$$

where  $n$  is the number of links,  $m_i$  is the mass of link  $i$ ,  $g$  is the gravitational acceleration, and  $h_{ci}$  is the height of the center of mass of link  $i$  with respect to a chosen zero level.

With the origin of frame 0 as the zero level, the potential energy is computed as

$$P_1 = m_1 \cdot g \cdot -r_{0,c1,y} \quad (6.33)$$

$$P_2 = m_2 \cdot g \cdot [-r_{0,1,y} + r_{1,c2,x} \cdot \cos(q_2)] \quad (6.34)$$

$$P_3 = 0 \quad (6.35)$$

$$P_4 = m_4 \cdot g \cdot [-r_{0,1,y} + r_{1,2,x} \cdot \cos(q_2) + r_{3,c4,y} \cdot \sin(-q_3 - q_2)] \quad (6.36)$$

$$P_5 = 0 \quad (6.37)$$

$$P_6 = m_6 \cdot g \cdot [-r_{0,1,y} + r_{1,2,x} \cdot \cos(q_2) + r_{3,4,y} \cdot \sin(-q_3 - q_2) + r_{5,c6,z} \cdot \sin(-q_5 - q_3 - q_2)] \quad (6.38)$$

$$P = P_1 + P_2 + P_4 + P_6 \quad (6.39)$$

where the third index added to the length vectors indicates the component of the vector. Since link 3 and 5 have zero mass, their potential energy is zero as well.

### 6.4.3 Pendulum Comparison

Simulating the system without any input makes an interesting comparison. The system can now be interpreted as a complex 3D-pendulum with four joints and four links, where two of the joints are ball joints (2 DOF) and the other two joints are revolute joints (1 DOF). Multiple linked pendulums have received a lot of attention from scientists because these systems are so easily constructed yet the behavior is extremely complicated. Even the simplest multiple linked pendulum, the double pendulum (2 DOF), is proved to be a chaotic system [18] and [2]. A chaotic system is sensitive to initial conditions, meaning that infinitesimal changes to the initial conditions in an otherwise perfect experiment will produce widely different results. It is not possible to predict the behavior, and it is a complex question to determine when any of the links will flip (if possible).

### 6.4.4 Simulations

It is fair to assume that the 3D-pendulum (6 DOF) will also be a chaotic system, as it can be interpreted as a double pendulum with additional degrees of freedom. Two simulations with very small changes in initial conditions show the responses in the joint variables, as well as the energy in the system during motion.

### Simulation 1

Initial conditions and joint torques are set to

$$q = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.40)$$

$$\dot{q} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.41)$$

$$\tau = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.42)$$

The response in joint variables are shown in Figure 6.6 and the energy is shown in Figure 6.7.

### Simulation 2

Initial conditions and joint torques are set to

$$q = [0 \ 0.000001 \ 0 \ 0 \ 0 \ 0]^T \quad (6.43)$$

$$\dot{q} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.44)$$

$$\tau = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.45)$$

The response in joint variables are shown in Figure 6.8 and the energy is shown in Figure 6.9.

## 6.4.5 Comments

These simulations confirm the expected chaotic behavior when the manipulator is interpreted as a 3D-pendulum. The difference in initial conditions is only 0.000001 radians (in joint 2), and the two simulations show entirely different responses of the joint variables.

In addition, the energy plots provide some really valuable information. The variations in kinetic and potential are different for the simulations, and is in accordance to the motion of the manipulator. Still, the total energy is constant in both simulations, and this confirms that the internal energy is conserved during the motion as it should be with zero torque applied. However, by studying the energy plots closely it can be observed that there are some small variations in the total energy. These variations are caused by numerical error when Matlab solves the differential equations, and would not be the case in a perfect experiment.

This section has presented some outstanding observations, which close to confirm that the derivation of the dynamic model is correct.

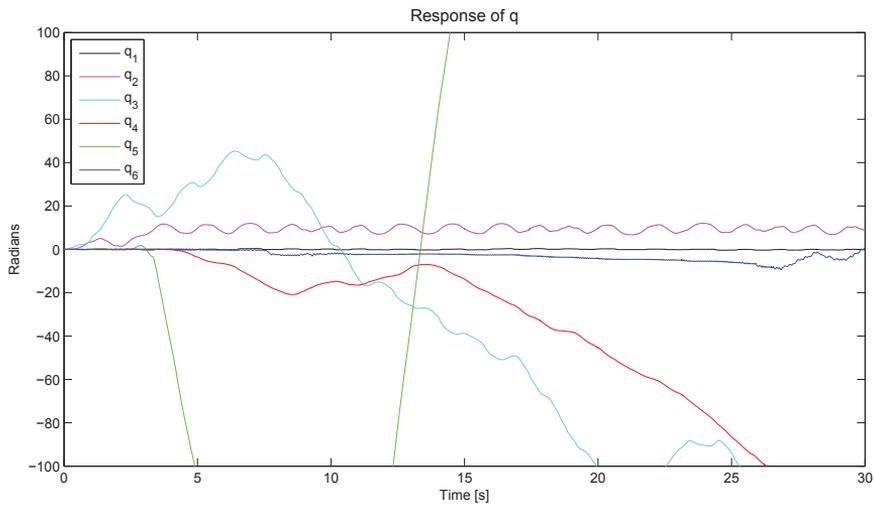


Figure 6.6: Open loop response with zero torque, simulation 1

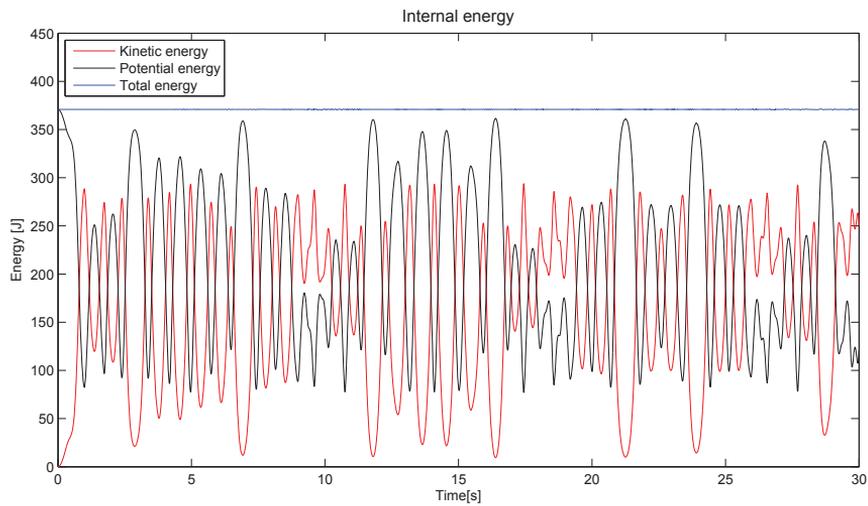


Figure 6.7: Energy in the system with zero torque, simulation 1

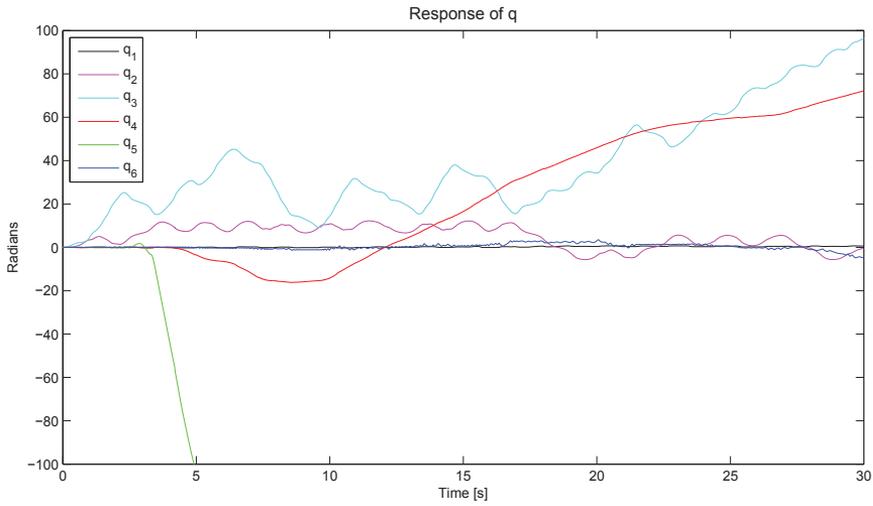


Figure 6.8: Open loop response with zero torque, simulation 2

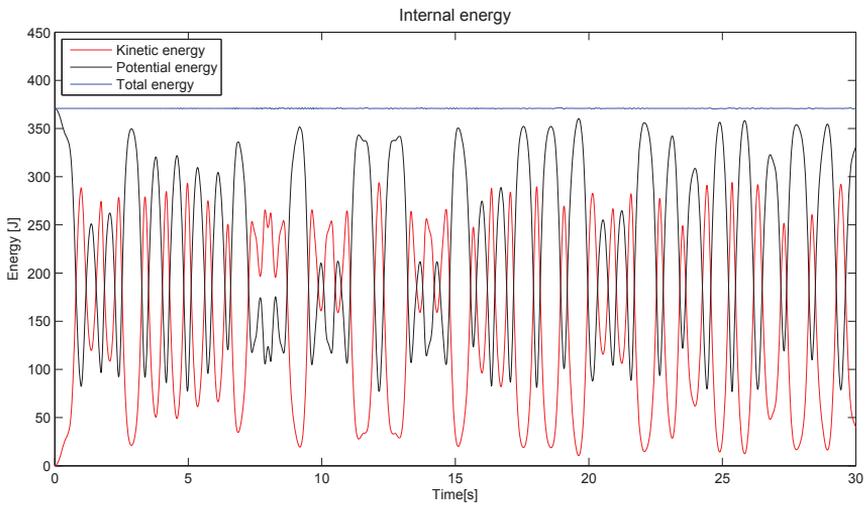


Figure 6.9: Energy in the system with zero torque, simulation 2

## 6.5 Closed Loop Position Control

The open loop analysis with desired torque in Section 6.3 showed that controlling the system in open loop is impossible. This section deals with the attempt of controlling the system in closed loop. In closed loop, feedback controllers observe the output and calculate the error between this output and a reference. To achieve desired output, controllers can take one or more of three standard control elements that were described in Section 2.4.

Closed loop position control is also called the set-point tracking problem. The goal is to demonstrate that the manipulator can move from the position given as initial conditions, position A, to the position given as the reference value, position B. The joint torque input is continuously calculated by the feedback controllers. The path taken from A to B, as well as how long the motion lasts, is not controlled in the set-point tracking problem.

Section 6.5.1 presents a mathematical proof showing that a simple PD control structure works great for position control of systems in the general form (6.1). Then in Section 6.5.2, PD controllers are added to the model in Simulink, and simulations verify that the system is stable and that the position control is satisfying.

### 6.5.1 Proof: PD control with gravity compensation

It is a remarkable fact that the simple PD scheme for set-point control can be shown to work in the general case of a system model in the form of Equation (6.1). This can be proved in a Lyapunov stability analysis, as shown in [20] and [17]. This proof is of such importance and relevance to this thesis that it will be restated in this section.

The proof is based on independent joint control, which means that each joint is controlled as a single-input/single-output (SISO) system. Adding PD controllers in the model, the input torque  $u$  can be written in vector form as

$$u = -K_p(q_{ref} - q) - K_d\dot{q} = -K_p\tilde{q} - K_d\dot{q} \quad (6.46)$$

where  $\tilde{q}$  is the error between the joint references and the actual joint variables, and  $K_p$  and  $K_d$  are positive definite diagonal matrices of proportional and derivative gains.

It can be assumed that the gravitational acceleration is constant and known, such that  $g(q)$  can be computed explicitly for all instants. By adding  $g(q)$  to the input, gravity compensation is achieved such that the complete system model is now given by

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \quad (6.47)$$

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = -K_p\tilde{q} - K_d\dot{q} + g(q) \quad (6.48)$$

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = -K_p\tilde{q} - K_d\dot{q} \quad (6.49)$$

To show that the input torque given in Equation (6.48) achieves asymptotic tracking, consider the Lyapunov function candidate

$$V = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} \tilde{q}^T K_p \tilde{q} \quad (6.50)$$

For the manipulator,  $V$  represents the total energy that would result if the actuators were replaced by springs with stiffness constants represented by  $K_p$ , and with equilibrium position in  $q_{ref}$ . Thus,  $V$  is a positive function except in the equilibrium position  $q = q_{ref}$  with  $\dot{q} = 0$ , at which point  $V$  is zero. If it can be shown that  $V$  is decreasing along any motion, this implies that the robot is moving toward that equilibrium position.

Noting that  $q_{ref}$  is constant, the derivative of  $V$  is given by

$$\dot{V} = \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \dot{q}^T K_p \tilde{q} \quad (6.51)$$

Solving for  $M(q) \ddot{q}$  in Equation (6.47) and substituting into the (6.51) yields

$$\dot{V} = \dot{q}^T (u - C(q, \dot{q}) \dot{q} - g(q)) + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \dot{q}^T K_p \tilde{q} \quad (6.52)$$

$$= \dot{q}^T (u - g(q) + K_p \tilde{q}) + \frac{1}{2} \dot{q}^T [\dot{M}(q) - 2C(q, \dot{q})] \dot{q} \quad (6.53)$$

$$= \dot{q}^T (u - g(q) + K_p \tilde{q}) \quad (6.54)$$

where  $\dot{M}(q) - 2C(q, \dot{q})$  is skew symmetric<sup>1</sup>, and Equation (2.11) then gives  $\dot{q}^T [\dot{M}(q) - 2C(q, \dot{q})] \dot{q} = 0$ . Substituting the input torque in Equation (6.48) for  $u$  in (6.54) above yields

$$\dot{V} = -\dot{q}^T K_d \dot{q} \leq 0 \quad (6.55)$$

The above analysis shows that  $V$  is decreasing as long as  $\dot{q}$  is not zero.

Moreover it is necessary to prove that the manipulator can not reach a position where  $\dot{q} = 0$  but  $q \neq q_{ref}$ . Suppose  $\dot{V} \equiv 0$ , meaning that  $\dot{V}$  is zero for all instants. Since  $K_d$  is positive definite, this implies that  $\dot{q} \equiv 0$  and hence  $\ddot{q} \equiv 0$ . Substituting this in the system model (6.49), the result becomes

$$0 = -K_p \tilde{q} \quad (6.56)$$

which implies that  $\tilde{q} = 0$ . Finally, La Salle's theorem<sup>2</sup> then proves that the equilibrium position  $q = q_{ref}$  is globally asymptotic stable.

It should be noted that if the gravitational terms  $g(q)$  are unknown, they can not be added to the input because then the input cannot be computed. Controlling the system would then require controllers with robust and adaptive properties (see Section 6.5.3).

<sup>1</sup>The skew-symmetry of  $\dot{M}(q) - 2C(q, \dot{q})$  is explained in [17], page 142-143

<sup>2</sup>La Salle's theorem is given in [20], page 456-457

## 6.5.2 Simulations with PD Control

The goal of this section is to perform simulations of the system with PD controllers, checking for asymptotic stability. If this can be accomplished, the mathematical proof in Section 6.5.1 is verified for the model. Figure 6.10 shows the Simulink model of the system in closed loop. With gravity compensation the model becomes

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = -K_p\tilde{q} - K_d\dot{q} \quad (6.57)$$

where the input is

$$u = -K_p\tilde{q} - K_d\dot{q} \quad (6.58)$$

Note that to increase the efficiency of the simulations, it is chosen to remove

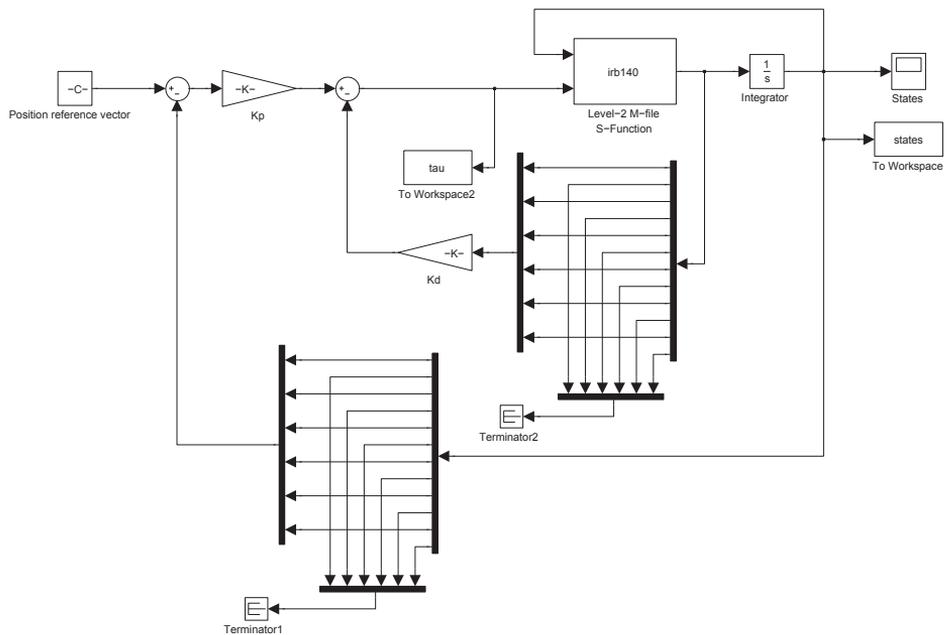


Figure 6.10: Closed loop simulink model with PD controllers

the gravitational terms directly in the model (in the *irb140* block) instead of adding it to the input.

The mathematical proof gives no other bounds on  $K_p$  and  $K_d$  except for being positive definite. Adjusting these controller gains optimally have not been a priority, because it will not be decisive for global asymptotic stability.

A set of satisfying gain matrices was found as simple as

$$K_p = \begin{bmatrix} 50 & 0 & 0 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 60 \end{bmatrix} \quad (6.59)$$

$$K_d = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 22 \end{bmatrix} \quad (6.60)$$

Three simulations with different initial conditions and reference values are presented to show the behavior of the system with these gains.

### Simulation 1

Initial conditions and reference values are set to

$$q_{init} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.61)$$

$$\dot{q}_{init} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.62)$$

$$q_{ref} = \left[ \frac{\pi}{2} \ 0 \ -\frac{\pi}{2} \ \pi \ \frac{\pi}{2} \ -\pi \right]^T \quad (6.63)$$

The response in joint variables are shown in Figure 6.11 and the control input is shown in Figure 6.12.

### Simulation 2

Initial conditions and reference values are set to

$$q_{init} = [0 \ \pi \ -\frac{\pi}{2} \ 0 \ 0 \ 0]^T \quad (6.64)$$

$$\dot{q}_{init} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (6.65)$$

$$q_{ref} = \left[ \pi \ 0 \ 0 \ \pi \ -\frac{\pi}{2} \ 0 \right]^T \quad (6.66)$$

The response in joint variables are shown in Figure 6.13 and the control input is shown in Figure 6.14.

### Simulation 3

Initial conditions and reference values are set to

$$q_{init} = [0 \quad \frac{\pi}{2} \quad -\frac{\pi}{2} \quad 0 \quad 0 \quad 0]^T \quad (6.67)$$

$$\dot{q}_{init} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (6.68)$$

$$q_{ref} = [-\pi \quad \pi \quad -\pi \quad -\pi \quad -\frac{\pi}{2} \quad \pi]^T \quad (6.69)$$

The response in joint variables are shown in Figure 6.15 and the control input is shown in Figure 6.16.

#### 6.5.3 Comments and Limitations

All simulations show that the states converge to the reference in about 3 seconds. Asymptotic stability is verified, and the response is very satisfying. Nevertheless, several factors deserve to be emphasized. First of all, actuators cannot supply infinite torque. The nominal torque of the actuators and their gear ratio limits the maximum input torque. This constraint can be included in Simulink simply by saturating the input, but doing this is not necessary of reasons explained as follows. In the proof in Section 6.5.1 the gravitational terms  $g(q)$  were added to the input because the gravitational acceleration was assumed to be constant and known. With this simplification it is taken for granted that the maximum input torque in the actuators is larger than  $g(q)$ . The data sheets for the IRB 140 do not state any torque values or other motor characteristics, but obviously this assumption is valid since the manipulator is observed to "beat the gravity" in a real environment. The mathematical proof gives no other bounds on the input torque, thus global asymptotic stability is proved also for saturated inputs. The only difference in the simulations will be the increased time to reach steady state.

Secondly, actuators cannot change the input torque value from  $\tau_a$  to  $\tau_b$  in zero time. In other words, the input can never be a perfect step function. The IRB 140 are controlled by electric AC-motors which supplies torque by passing electricity to an electromagnet creating a magnetic field. How fast this magnetic field is created will determine the maximum rate of change in input torque. Rate limiters can be included in Simulink, but it is assumed that electric motors create their electric fields very quickly. Consequently, rate limiters will not make any significant difference in the simulations.

Some limitations have been chosen deliberately. First, joint friction is not taken into account because of two reasons. First, it will be like a shot in the dark to estimate the friction parameters without any given information. Secondly, it does not really make a difference to the simulations anyway when the input is not saturated. However, if joint friction was to be taken into account, the simplest way to include it would be to only model viscous friction, being proportional to the joint velocity. The system model would then be

$$M(q)\ddot{q} + C(q, \dot{q}) + F_v\dot{q} + \dot{q} + g(q) = u \quad (6.70)$$

where  $F_v$  is a diagonal matrix of the joint friction coefficients. This joint friction material is taken from [1].

Note also that the simulations do not take into account the workspace of the manipulator at all. Since the main goal of this chapter is to prove the validity of the model, and not to optimize a control system for a specific job task, it was found convenient to not include the workspace restrictions. The joints are allowed to revolve freely, and no obstacles, floor, roof or walls are considered. The data sheet (Appendix A) specifies the actual working range for the joints.

It should be mentioned that there exists several other control techniques and methodologies that can be applied to the control of manipulators. The choice of control structure should therefore match the requirements for the robot operation. If there are obstacles within the workspace of the manipulator, continuous path tracking could be necessary to avoid collisions. Many operations may also require that the manipulator moves from point A to point B in a precise fixed time interval. If the robot operation requires objects to be moved around, robust and adaptive controllers are superior. Note that this can often be the case for the IRB 140, as it is designed to handle payloads of up to 6 kg. The mechanical design, motor characteristics, and problems due to backlash, friction and gear reduction, may also affect the choice of control structure.

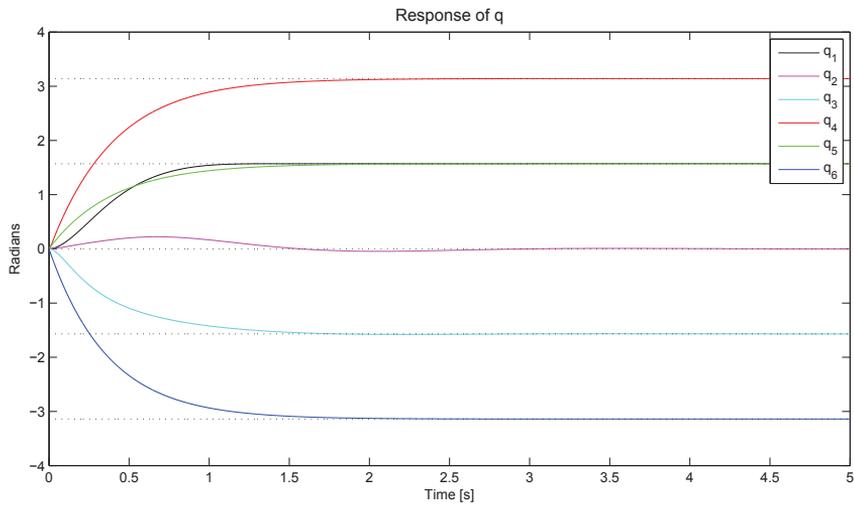


Figure 6.11: Closed loop position control response, simulation 1

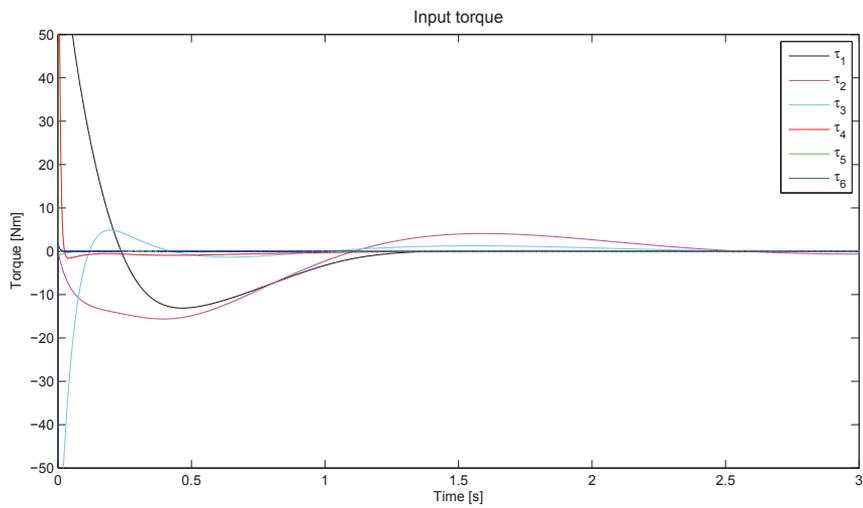


Figure 6.12: Closed loop position control input, simulation 1

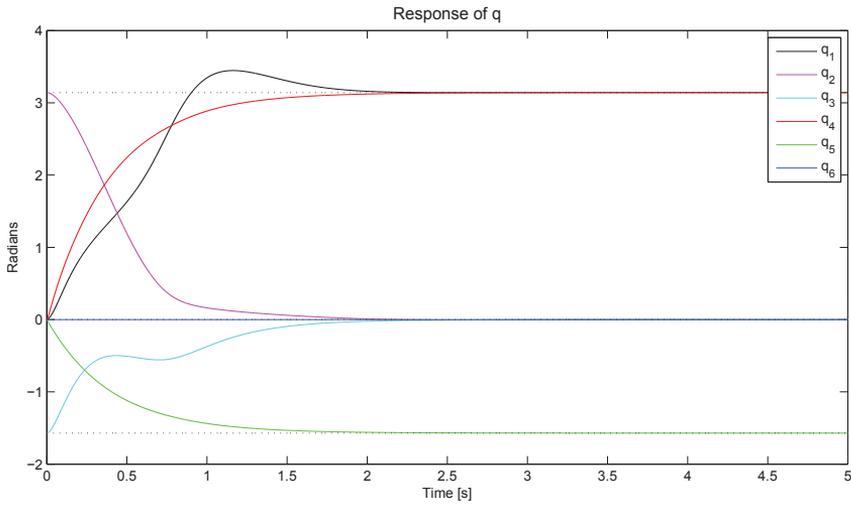


Figure 6.13: Closed loop position control response, simulation 2

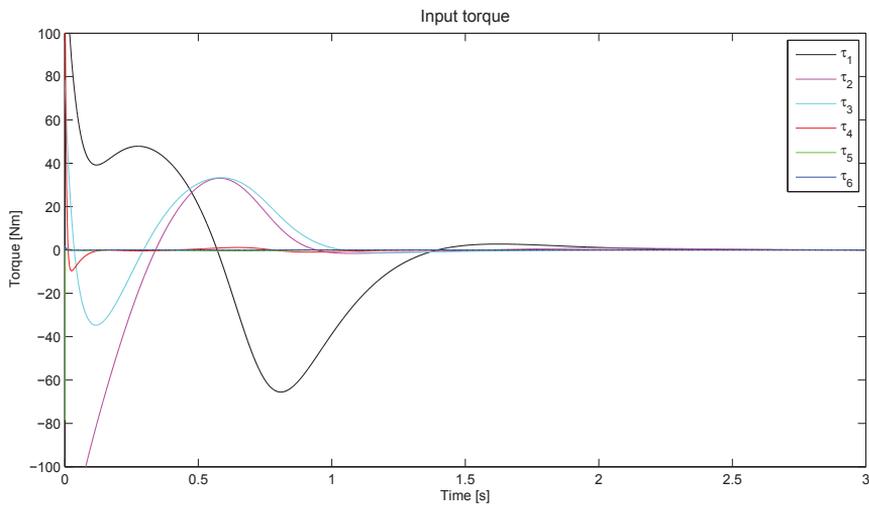


Figure 6.14: Closed loop position control input, simulation 2

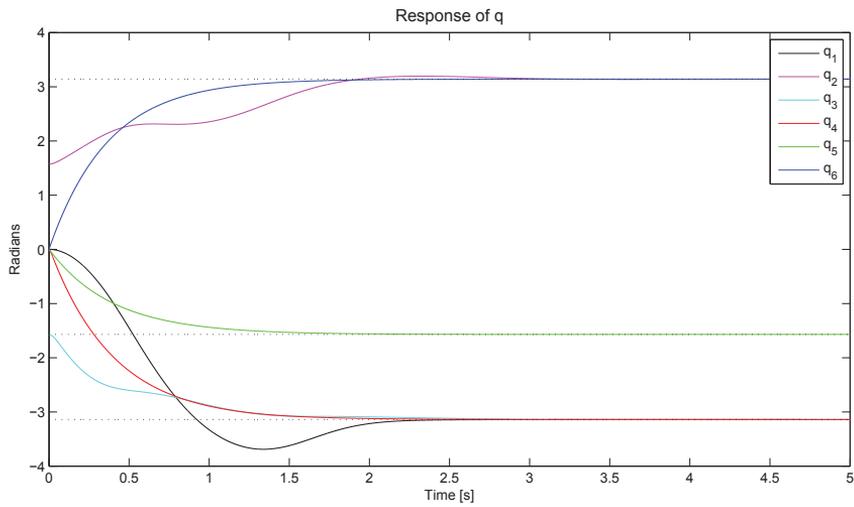


Figure 6.15: Closed loop position control response, simulation 3

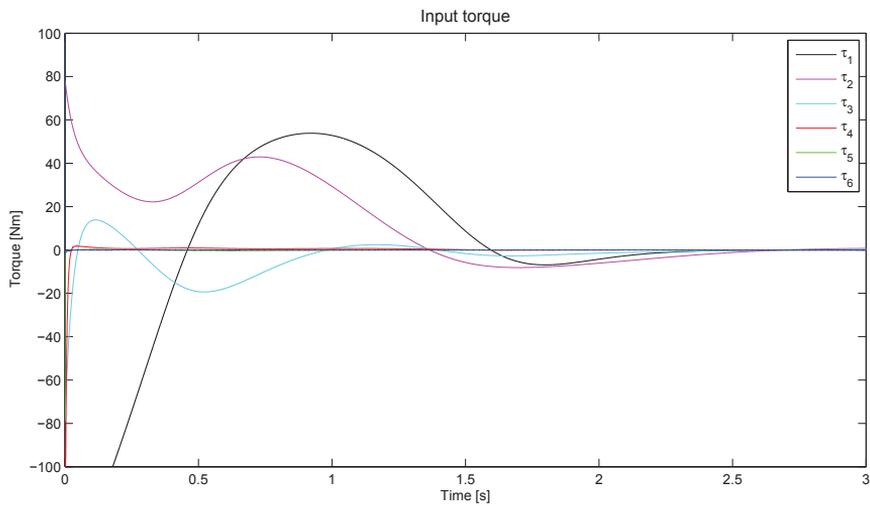


Figure 6.16: Closed loop position control input, simulation 3



## Chapter 7

# Comparing the Results to the Euler-Lagrange Model

The two main approaches for dynamic modeling of robot manipulators are the Newton-Euler formulation and the Euler-Lagrange formulation. In Chapter 3.1.1 it was stated that there is no clear answer to the question of which of the methods is better than the other, because of all the factors that influence the computation time. However, [7] proves at least that a recursive procedure is more efficient than treating the manipulator as a whole.

The purpose of this chapter is to compare the behavior and computation times of the model derived by the Newton-Euler formulation in this thesis, to a model of the IRB 140 derived by the Euler-Lagrange formulation [13]. The latter was derived by the standard Euler-Lagrange procedure, meaning the manipulator was treated as whole and analyzed based on its kinetic and potential energy.

### 7.1 Remarks on the Direction of State Variables

The ideal situation for the comparison would be that all definitions of the kinematic chain as well as the parameters of system are equal in the two models. This was originally the intention, and the frames and parameters in the Newton-Euler model were defined deliberately to accomplish this. Unfortunately, it was discovered by experimental simulations later on that the direction of change in at least one state variable is defined opposite in the Euler-Lagrange model. Referring to the representation of the Newton-Euler system in Figure 4.2, this finding applies to  $q_3$ , meaning the direction of positive  $q_3$  in the Euler-Lagrange model is counter-clockwise when looking at that figure. Defining the subscript *ne* for Newton-Euler and *el* for Euler-Lagrange, the following simulations confirm this discovery.

## Simulation 1

The initial conditions and applied torque for the simulations in Figure 7.1 are

$$q_{ne,init} = [0 \quad \pi \quad -\frac{\pi}{2} \quad 0 \quad 0 \quad 0]^T \quad (7.1)$$

$$\dot{q}_{ne,init} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (7.2)$$

$$\tau_{ne} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (7.3)$$

$$q_{el,init} = [0 \quad \pi \quad \frac{\pi}{2} \quad 0 \quad 0 \quad 0]^T \quad (7.4)$$

$$\dot{q}_{el,init} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (7.5)$$

$$\tau_{el} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (7.6)$$

As observed in the figures, the behavior of the two models are identical when  $q_3$  has opposite signs. The initial conditions corresponds to the configuration where the manipulator is hanging like a pendulum in a stable equilibrium point.

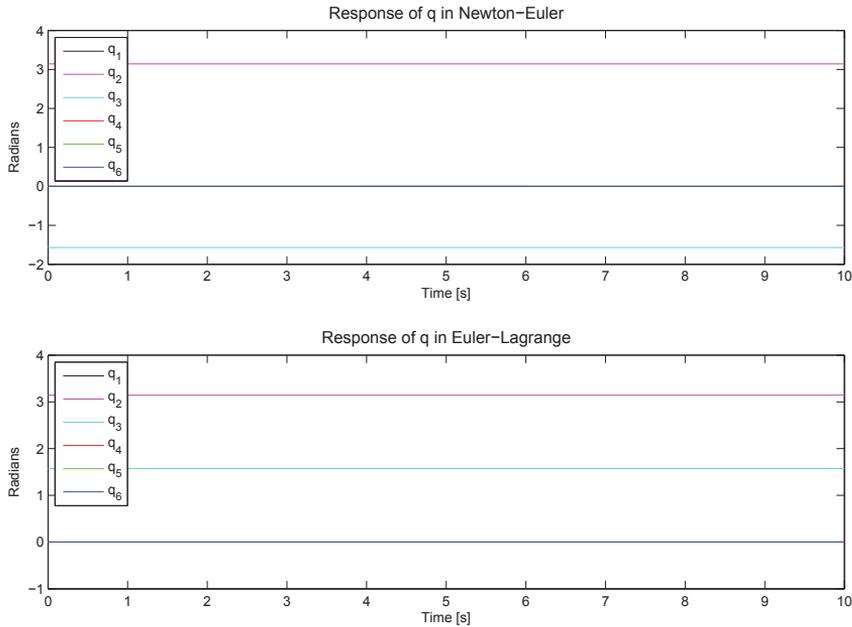


Figure 7.1: Confirming opposite  $q_3$ -direction, simulation 1

## Simulation 2

The initial conditions and applied torque for the simulations in Figure 7.2 are

$$q_{ne,init} = [0 \quad \pi \quad \frac{\pi}{2} \quad 0 \quad 0 \quad 0]^T \quad (7.7)$$

$$\dot{q}_{ne,init} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (7.8)$$

$$\tau_{ne} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (7.9)$$

$$q_{el,init} = [0 \quad \pi \quad -\frac{\pi}{2} \quad 0 \quad 0 \quad 0]^T \quad (7.10)$$

$$\dot{q}_{el,init} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (7.11)$$

$$\tau_{el} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (7.12)$$

The behavior of the models are now similar but not identical when  $q_3$  has opposite signs. The initial conditions corresponds to the configuration where link 2 is hanging straight down, while link 4 and 6 represents a double inverted pendulum on top of link 2. As observed, the combination of gravity and numerical error in the solver will eventually move the manipulator out of the unstable equilibrium point. The difference in state responses is simply caused by chaotic behavior (see Section 6.4.3).

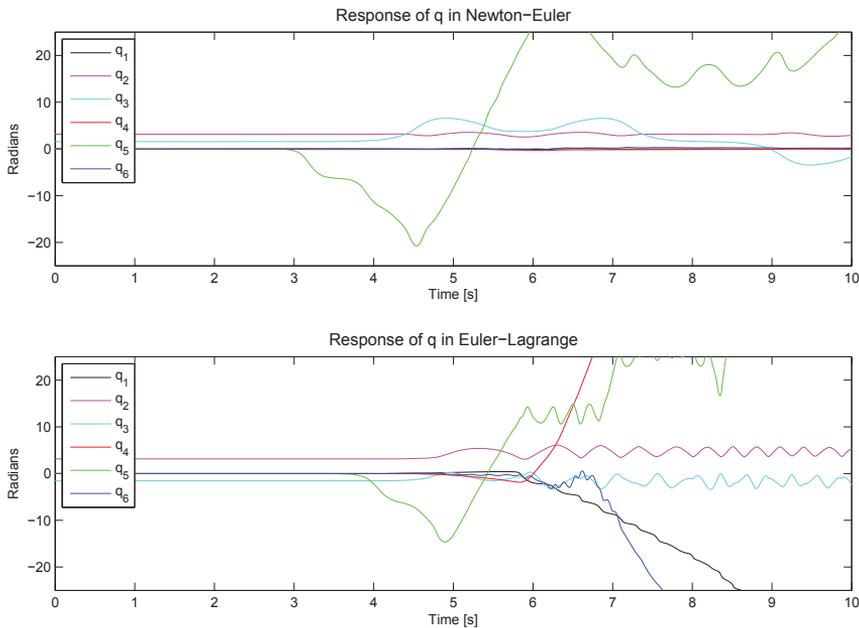


Figure 7.2: Confirming opposite  $q_3$ -direction, simulation 2

## 7.2 Computation Times

The results in Section 7.1 confirm the difference in direction of  $q_3$ , but the coupled kinematics make it hard to draw any conclusions about the direction of other states in the Euler-Lagrange model. However, the variations in computation time caused by different directions of state variables will not be of any significance. This section will investigate the efficiency of the formulations described in terms of computation times in open loop and closed loop. To distinguish clearly, the simulation time chosen in Simulink will be referred to as *simulation time*, and the actual time recorded during the simulation will be referred to as *real time*.

### 7.2.1 Open Loop

The simulations showed in Figure 7.2 are used to compare computation times in open loop. The simulation time for both models were set to 10 seconds. The real times were recorded as 7 minutes for the Euler-Lagrange model and only 6 seconds for the Newton-Euler model.

### 7.2.2 Closed loop

Three different simulations for the Newton-Euler model were performed in closed loop with PD controllers (see Section 6.5.2). All of them showed that the states converged to the reference in about 3 seconds of simulation time. The total simulation times were 5 seconds for all simulations, and the real times were recorded to be 28 minutes, 32 minutes and 27 minutes respectively.

Equivalent simulations in closed loop with PD controllers have been performed with the Euler-Lagrange model, and the results are quite remarkable. The simulations were awfully time-consuming and they required so much computer capacity that it was chosen to stop the simulations after 2.3 seconds of simulation time. The real time was then at about 18 hours for all three simulations.

The three simulations for the Euler-Lagrange model are presented below. Note that the transient responses differs from the the ones in the Newton-Euler model because of the difference in direction of state variables. Unfortunately, 2.3 seconds was just a little short simulation time to reach steady state, but still the plots show enough to prove the same asymptotic stable behavior as for the Newton-Euler model.

### Simulation 1

Initial conditions and reference values are set equal to (6.61)-(6.63), and the response is shown in Figure 7.3.

## Simulation 2

Initial conditions and reference values are set equal to (6.64)-(6.66), and the response is shown in Figure 7.4.

## Simulation 3

Initial conditions and reference values are set equal to (6.67)-(6.69), and the response is shown in Figure 7.5.

### 7.2.3 Comments

All simulation times and recorded real times in this comparison are summarized in Table 7.1. Several times throughout this thesis it has been pointed out that a recursive procedure is faster than treating the manipulator as a whole. This experiment underlines that statement to the full extent.

Simulation plot	Newton-Euler		Euler-Lagrange	
	Sim. time	Real time	Sim. time	Real time
Figure 7.2	10 s	6 s	10 s	7 m
Figure 6.11 and 7.3	5 s	28 m	2.3 s	18 h
Figure 6.13 and 7.4	5 s	32 m	2.3 s	18 h
Figure 6.15 and 7.5	5 s	27 m	2.3 s	18 h

Table 7.1: Computation times

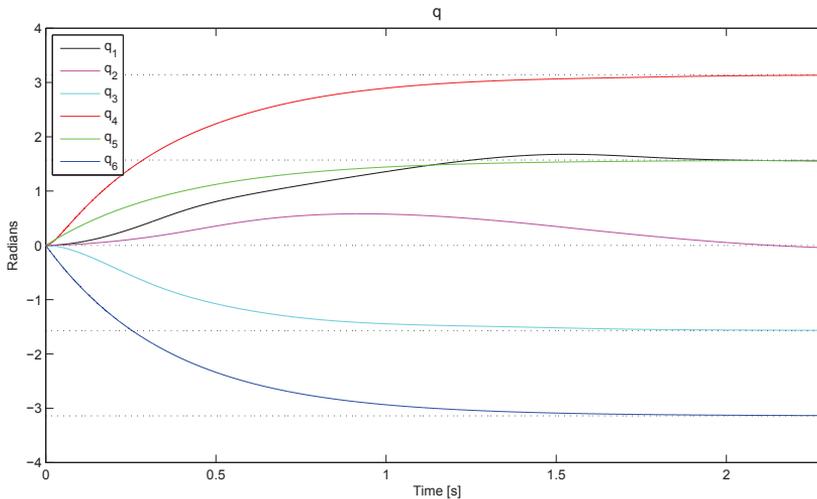


Figure 7.3: Euler-Lagrange closed loop position control response, simulation 1

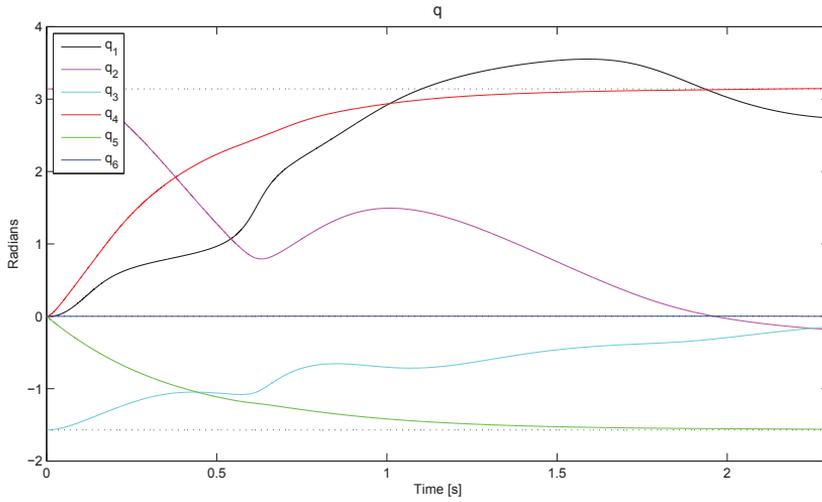


Figure 7.4: Euler-Lagrange closed loop position control response, simulation 2

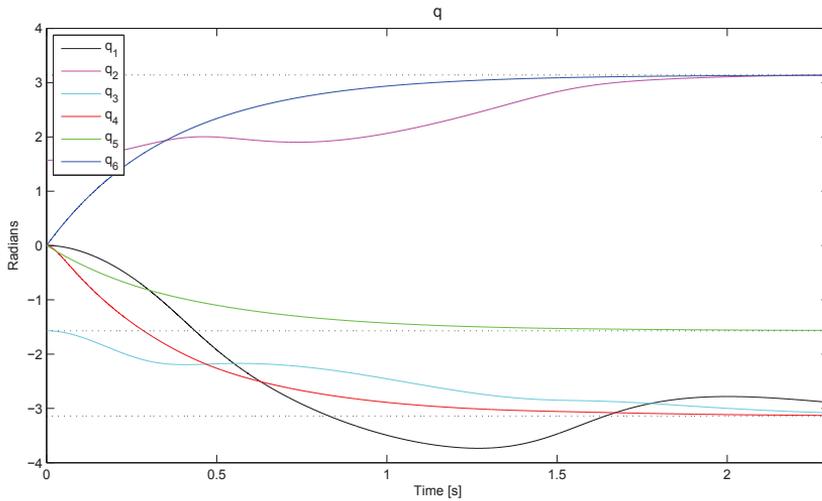


Figure 7.5: Euler-Lagrange closed loop position control response, simulation 3

# Chapter 8

## Concluding Remarks

This chapter presents a concluding summary of the thesis, as well as recommendations for improvement and usage of the results.

### 8.1 Conclusion

The topic of this thesis has been dynamic modeling and simulation of robot manipulators. Different methods for dynamic modeling have been introduced, with particular focus on the Newton-Euler formulation. The efficiency and advantages of the formulation have been compared to the characteristics of the Euler-Lagrange formulation, although there could not be drawn a conclusion to the question of which method is better than the other in general. The computation time depends on several aspects in the system to be analyzed, and the approaches provide different insights such that personal preference becomes a factor as well.

It has been shown that estimating the dynamic parameters accurately is a hard and time-consuming challenge. It requires either the possibility to measure the state variables and its derivatives during motion of the manipulator, or specific knowledge about other identification techniques as for example CAD modeling. Even if such an attempt is to be performed, the dynamic parameters will not be perfectly accurate. In the model for the IRB 140, the dynamic parameters have been estimated based on inspecting the manipulator carefully, making intuitive guesses when required.

Simulations of the dynamic model had as main purpose to prove the validity of the model. Open loop simulations with desired torque showed that the behavior of the system was unstable just as assumed; the slightest disturbance in the system led to a completely uncontrollable motion. Energy investigations of the system with zero torque applied showed that the internal energy of the system is constant during any motion as long as no dissipations of energy are taken into account. That is an outstanding observation that really underlines that the model is behaving correctly.

Global asymptotic stability of the system with PD control and gravity compensation was proved mathematically in a Lyapunov stability analysis. Afterwards, this was confirmed to be the case for the model by simulations with PD control.

It was mentioned that the computation times of the Newton-Euler formulation and Euler-Lagrange formulation depends on several factors. However, in any case, it is a fact that a recursive procedure is more efficient than treating the manipulator as a whole. By comparing simulations of the Newton-Euler model and the Euler-Lagrange model of the IRB 140, this statement was proved to hold in all cases.

## 8.2 Recommendations for Future Work

Derivation of dynamic models will in almost any case involve challenging parameter estimation. The dynamic parameters in this thesis are estimated quite roughly, and accordingly there is much potential for improvement. Being able to measure the state variables and its derivatives in experiments with the IRB 140 would be a huge step towards estimating the centers of mass and inertia tensors quite accurately. If the torques can also be measured, for example by measuring the current, it is actually possible to compute the parameters explicitly by the identification method explained in Section 3.1.2.

The model in this thesis is also limited in other aspects. A perfect model would contain dynamic effects like joint friction and joint flexibility, and there would be bounds on the maximum input torque for the motors. Actually, the data sheet (see Appendix A) states the maximum velocity for each joint, such that the maximum input torques can be calculated explicitly if motor specifications are known.

Controller design should also be recommended for future work. Inverse dynamics control can be utilized for trajectory tracking of the manipulator. As mentioned in Chapter 6, trajectory tracking could be required to avoid collisions with obstacles within the workspace of the manipulator. Obviously the workspace of the manipulator is bounded by the working range of the joints (given in Appendix A) as well as the floor, the roof, and the walls, in addition to possible obstacles. The main drawback with inverse dynamics is that the dynamic parameters must be known, if not, the ideal performance is no longer guaranteed. Robust and adaptive motion control is one way of overcoming this problem. The goal of this kind of control is to maintain desired performance despite parametric uncertainty. Thus, robust and adaptive control is superior for robot operation that requires objects to be moved around by the robot arm.

If actions and improvements like mentioned above are performed, the model attained might be good enough to achieve pretty accurate motion control of the IRB 140. Being able to design a control system that can compete with ABB's *QuickMove* and *TrueMove* systems would be the ultimate achievement. Unfortunately, implementing the model cannot be accomplished as long as access to the state variables and the controller in the manipulator is not granted.

# Appendix A

## IRB 140 Data Sheet



# IRB 140 Industrial Robot

## Main Applications

Arc welding  
Assembly  
Cleaning/Spraying  
Machine tending  
Material handling  
Packing  
Deburring



### Small, Powerful and Fast

Compact, powerful IRB 140 industrial robot.

Six axis multipurpose robot that handles payload of 6 kg, with long reach (810 mm). The IRB 140 can be floor mounted, inverted or on the wall in any angle. Available as Standard, FoundryPlus, Clean Room and Wash versions, all mechanical arms completely IP67 protected, making IRB 140 easy to integrate in and suitable for a variety of applications. Uniquely extended radius of working area due to bend-back mechanism of upper arm, axis 1 rotation of 360 degrees even as wall mounted.

The compact, robust design with integrated cabling adds to overall flexibility. The Collision Detection option with full path retraction makes robot reliable and safe.

Using IRB 140T, cycle-times are considerably reduced where axis 1 and 2 predominantly are used.

Reductions between 15-20 % are possible using pure axis 1 and 2 movements. This faster versions is well suited for packing applications and guided operations together with PickMaster.

IRB Foundry Plus and Wash versions are suitable for operating in extreme foundry environments and other harsh environments with high requirements on corrosion resistance and tightness. In addition to the IP67 protection, excellent surface treatment makes the robot high pressure steam washable. The white-finish Clean Room version meets Clean Room class 10 regulations, making it especially suited for environments with stringent cleanliness standards.

# IRB 140

## Specification

Robot versions	Handling capacity	Reach of 5th axis	Remarks
IRB 140/IRB 140T	6 kg	810 mm	
IRB 140F/IRB 140TF	6 kg	810 mm	Foundry Plus Protection
IRB 140CR/ IRB 140TCR	6 kg	810 mm	Clean Room
IRB 140W/ IRB 140TW	6 kg	810 mm	Wash Protection
Supplementary load (on upper arm alt. wrist)			
on upper arm	1 kg		
on wrist	0.5 kg		
Number of axes			
Robot manipulator		6	
External devices		6	
Integrated signal supply		12 signals on upper arm	
Integrated air supply		Max. 8 bar on upper arm	

## Performance

Position repeatability 0.03 mm (average result from ISO test)

Axis movement	Axis	Working range
	1, C Rotation	360°
	2, B Arm	200°
	3, A Arm	280°
	4, D Wrist	Unlimited (400° default)
	5, E Bend	240°
	6, P Turn	Unlimited (800° default)

Max. TCP velocity 2.5 m/s

Max. TCP acceleration 20 m/s<sup>2</sup>

Acceleration time 0-1 m/s 0.15 sec

## Velocity

Axis no.	IRB 140	IRB 140T
1	200°/s	250°/s
2	200°/s	250°/s
3	260°/s	260°/s
4	360°/s	360°/s
5	360°/s	360°/s
6	450°/s	450°/s

## Cycle time

5 kg Picking side	IRB 140	IRB 140T
cycle 25 x 300 x 25 mm	0,85s	0,77s

## Electrical Connections

Supply voltage	200–600 V, 50/60 Hz
Rated power	
Transformer rating	4.5 kVA
Power consumption typicly	0,4 kW

## Physical

Robot mounting	Any angle
Dimensions	
Robot base	400 x 450 mm
Robot controller H x W x D	950 x 800 x 620 mm
Weight	
Robot manipulator	98 kg

## Environment

Ambient temperature for	
Robot manipulator	5 – 45°C
Relative humidity	Max. 95%
Degree of protection,	

Manipulator	IP67
Options	
	Foundry
	Wash (High pressure steam washable)
	Clean Room, class 6 (certified by IPA)

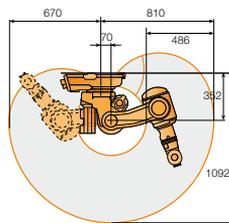
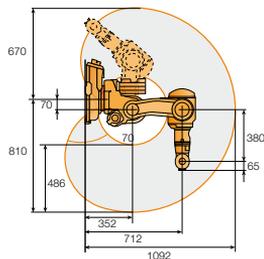
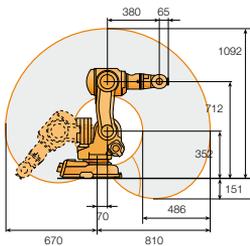
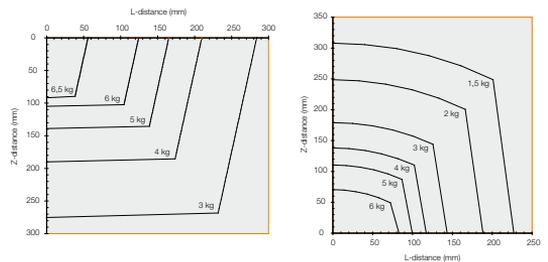


Noise level	Max. 70 dB (A)
Safety	
	Double circuits with supervision, emergency stops and safety functions,
	3-position enable device

Emission EMC/EMI-shielded

Data and dimensions may be changed without notice

## Working range



## Appendix B

# Automated Framework



## Automated framework for deriving the dynamic model of robot manipulators by the Newton-Euler formulation

This is an automated framework for deriving the dynamic model of robot manipulators by the Newton-Euler formulation. The framework is restricted to manipulators with rigid links and only revolute joints, but it can be adjusted to suit other kinds of manipulators if desired. The composition presented is for three degrees of freedom, but due to the recursive procedure it can easily be adjusted to any number of degrees of freedom.

Complex manipulators with several degrees of freedom will result in huge dynamic systems, and it might be desirable to evaluate smaller parts of a system individually. Every element of the inertia matrix, as well as the gravity components and the Coriolis and centrifugal components, are evaluated individually towards the end of the framework. It is optional to define symbolic values if the goal is a less specific model, but note that the more symbolic values and the less defined zeros, the larger dynamic model.

The Matlab conversion code used in the end can be applied to any expression, and is a simple way to convert Maple code to Matlab code. Note that the joint variable derivatives are not known in the Matlab conversion code. These quantities will have to be redefined as desired in Matlab.

All quantities must be defined according to the standard convention of how a manipulator is interpreted. In addition, it is important to add or remove lines following the recursive procedure, according to the how many degrees of freedom there are in the manipulator to be analyzed. Explanations for the parts that has to be adjusted (grayed out) are given below as well as in the code (green text).

**Joint variables:** Specify the number of joint variables (degrees of freedom) in the manipulator by adjusting the q-vector.

**Link length vectors and vectors to the centers of mass:** Define the link length vectors and the vectors to the centers of mass, which all are independent of configuration.

**Gravity vector in inertial frame:** Define the direction of gravity in the inertial frame (frame 0).

**Rotation matrices:** Define the rotation matrices describing rotations from every frame to the subsequent frame.

**Inertia matrices:** Define the inertia tensors for the links. If the mass distribution of a link is symmetric with respect to the attached frame, then the inertia tensor will be diagonal.

```
[> restart :
[> with(LinearAlgebra) :
[>
[> #Joint variables (degrees of freedom)
[> #TO DO: Specify the number of joint variables (degrees of freedom) in the
manipulator by adjusting the q-vector.
[> q := Vector( [ [q1(t)], [q2(t)], [q3(t)] ] ) :
[> Dq := map(diff, q, t) :
[> DDq := map(diff, Dq, t) :
[>
[> #Link length vectors and vectors to the centers of mass
[> #TO DO: Define the link length vectors and the vectors to the centers of
mass, which all are independent of configuration.
[> r0c1 := Vector( [ [rc1x], [rc1y], [rc1z] ] ) :
[> r1c2 := Vector( [ [rc2x], [rc2y], [rc2z] ] ) :
[> r2c3 := Vector( [ [rc3x], [rc3y], [rc3z] ] ) :
```

```

> r01 := Vector( [ [r1x], [r1y], [r1z]] ) :
> r12 := Vector( [ [r2x], [r2y], [r2z]] ) :
> r23 := Vector( [ [r3x], [r3y], [r3z]] ) :
> r1c1 := r0c1 - r01 :
> r2c2 := r1c2 - r12 :
> r3c3 := r2c3 - r23 :
>
> #Gravity vector in inertial frame
> #TO DO: Define the direction of gravity in the inertial frame (frame 0).
> g0 := Vector( [ [0], [0], [-g]] ) :
>
> #Rotation matrices
> #TO DO: Define the rotation matrices describing rotations from every frame to
the subsequent frame.
> R01 :=  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  :
> R12 :=  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  :
> R23 :=  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  :
> R02 := MatrixMatrixMultiply( R01, R12 ) :
> R03 := MatrixMatrixMultiply( R02, R23 ) :
>
> #Inertia matrices
> #TO DO: Define the inertia tensors for the links. If the mass distribution of
a link is symmetric with respect to the attached frame, then the inertia
tensor will be diagonal.
> I1 :=  $\begin{bmatrix} I1xx & I1xy & I1xz \\ I1yx & I1yy & I1yz \\ I1zx & I1zy & I1zz \end{bmatrix}$  :
> I2 :=  $\begin{bmatrix} I2xx & I2xy & I2xz \\ I2yx & I2yy & I2yz \\ I2zx & I2zy & I2zz \end{bmatrix}$  :
> I3 :=  $\begin{bmatrix} I3xx & I3xy & I3xz \\ I3yx & I3yy & I3yz \\ I3zx & I3zy & I3zz \end{bmatrix}$  :
>
> #Axis of rotation of joint i expressed in frame i
> z0 := Vector( [ [0], [0], [1]] ) :
> b1 := MatrixMatrixMultiply( Transpose( R01 ), z0 ) :
> b2 := combine( MatrixMatrixMultiply( Transpose( R02 ), MatrixVectorMultiply( R01, z0 ) ), trig ) :

```

```

> b3 := combine(MatrixMatrixMultiply(Transpose(R03), MatrixVectorMultiply(R02, z0)), trig) :
>
> #Forward recursion: Link 1
> ω1 := b1·Dq[1] :
> α1 := b1·DDq[1] + CrossProduct(ω1, b1·Dq[1]) :
> Dω1 := map(diff, ω1, t) :
> ae1 := CrossProduct(Dω1, r01) + CrossProduct(ω1, CrossProduct(ω1, r01)) :
> ac1 := CrossProduct(Dω1, r0c1) + CrossProduct(ω1, CrossProduct(ω1, r0c1)) :
> g1 := MatrixVectorMultiply(Transpose(R01), g0) :
>
> #Forward recursion: Link 2
> ω2 := combine(MatrixVectorMultiply(Transpose(R12), ω1) + b2·Dq[2], trig) :
> α2 := combine(MatrixVectorMultiply(Transpose(R12), α1) + b2·DDq[2] + CrossProduct(ω2, b2·Dq[2]),
  trig) :
> Dω2 := map(diff, ω2, t) :
> ae2 := combine(MatrixVectorMultiply(Transpose(R12), ae1) + CrossProduct(Dω2, r12) + CrossProduct(ω2,
  CrossProduct(ω2, r12)), trig) :
> ac2 := combine(MatrixVectorMultiply(Transpose(R12), ae1) + CrossProduct(Dω2, r1c2)
  + CrossProduct(ω2, CrossProduct(ω2, r1c2)), trig) :
> g2 := combine(MatrixVectorMultiply(Transpose(R02), g0), trig) :
>
> #Forward recursion: Link 3
> ω3 := combine(MatrixVectorMultiply(Transpose(R23), ω2) + b3·Dq[3], trig) :
> α3 := combine(MatrixVectorMultiply(Transpose(R23), α2) + b3·DDq[3] + CrossProduct(ω3, b3·Dq[3]),
  trig) :
> Dω3 := map(diff, ω3, t) :
> ae3 := combine(MatrixVectorMultiply(Transpose(R23), ae2) + CrossProduct(Dω3, r23) + CrossProduct(ω3,
  CrossProduct(ω3, r23)), trig) :
> ac3 := combine(MatrixVectorMultiply(Transpose(R23), ae2) + CrossProduct(Dω3, r2c3)
  + CrossProduct(ω3, CrossProduct(ω3, r2c3)), trig) :
> g3 := combine(MatrixVectorMultiply(Transpose(R03), g0), trig) :
>
> #Backward recursion: Link 3
> f3 := m3·ac3 - m3·g3 :
> τ3 := -CrossProduct(f3, r2c3) + MatrixVectorMultiply(I3, α3) + CrossProduct(ω3, MatrixVectorMultiply(I3,
  ω3)) :
> τ3Dyn := collect(combine(MatrixVectorMultiply(Transpose(b3), τ3), trig), {DDq[1], DDq[2], DDq[3],
  Dq[1], Dq[2], Dq[3]}) :
>
> #Backward recursion: Link 2
> f2 := MatrixVectorMultiply(R23, f3) + m2·ac2 - m2·g2 :
> τ2 := MatrixVectorMultiply(R23, τ3) - CrossProduct(f2, r1c2) + CrossProduct(MatrixVectorMultiply(R23, f3),
  r2c2) + MatrixVectorMultiply(I2, α2) + CrossProduct(ω2, MatrixVectorMultiply(I2, ω2)) :
> τ2Dyn := collect(combine(MatrixVectorMultiply(Transpose(b2), τ2), trig), {DDq[1], DDq[2], DDq[3],
  Dq[1], Dq[2], Dq[3]}) :
>
> #Backward recursion: Link 1
> f1 := MatrixVectorMultiply(R12, f2) + m1·ac1 - m1·g1 :

```

```

>  $\tau 1 := \text{MatrixVectorMultiply}(R12, \tau 2) - \text{CrossProduct}(f1, r0c1) + \text{CrossProduct}(\text{MatrixVectorMultiply}(R12, f2),$ 
   $r1c1) + \text{MatrixVectorMultiply}(I1, \alpha 1) + \text{CrossProduct}(\omega 1, \text{MatrixVectorMultiply}(I1, \omega 1)) :$ 
>  $\tau 1 \text{Dyn} := \text{collect}(\text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(b1), \tau 1), \text{trig}), \{DDq[1], DDq[2], DDq[3],$ 
   $Dq[1], Dq[2], Dq[3]\}) :$ 
>
> #The final dynamic system
>  $\tau 0 := \text{Vector}([\tau 1 \text{Dyn}], [\tau 2 \text{Dyn}], [\tau 3 \text{Dyn}]) :$ 
>
> #Evaluating smaller parts of the system
>  $m11 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1] = 1, DDq[2] = 0, DDq[3] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, g = 0\}) :$ 
>  $m12 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1] = 0, DDq[2] = 1, DDq[3] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, g = 0\}) :$ 
>  $m13 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1] = 0, DDq[2] = 0, DDq[3] = 1, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, g = 0\}) :$ 
>
>  $m21 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1] = 1, DDq[2] = 0, DDq[3] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, g = 0\}) :$ 
>  $m22 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1] = 0, DDq[2] = 1, DDq[3] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, g = 0\}) :$ 
>  $m23 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1] = 0, DDq[2] = 0, DDq[3] = 1, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, g = 0\}) :$ 
>
>  $m31 := \text{eval}(\tau 3 \text{Dyn}, \{DDq[1] = 1, DDq[2] = 0, DDq[3] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, g = 0\}) :$ 
>  $m32 := \text{eval}(\tau 3 \text{Dyn}, \{DDq[1] = 0, DDq[2] = 1, DDq[3] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, g = 0\}) :$ 
>  $m33 := \text{eval}(\tau 3 \text{Dyn}, \{DDq[1] = 0, DDq[2] = 0, DDq[3] = 1, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, g = 0\}) :$ 
>
>  $g3 := \text{eval}(\tau 3 \text{Dyn}, \{DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0\}) :$ 
>  $g2 := \text{eval}(\tau 2 \text{Dyn}, \{DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0\}) :$ 
>  $g1 := \text{eval}(\tau 1 \text{Dyn}, \{DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0\}) :$ 
>
>  $M0 := \text{Matrix}([\ [m11, m12, m13], [m21, m22, m23], [m31, m32, m33] ])$  :
>  $G0 := \text{Vector}([\ [g1], [g2], [g3] ])$  :
>  $C0Dq := \tau 0 - \text{MatrixVectorMultiply}(M0, DDq) - G0 :$ 
>
>  $cDq1 := C0Dq(1) :$ 
>  $cDq2 := C0Dq(2) :$ 
>  $cDq3 := C0Dq(3) :$ 
>
> #Matlab conversion
>  $\text{with}(\text{CodeGeneration}) :$ 
>  $\text{Matlab}(\tau 0, \text{resultname} = \text{"DynamicSystem"}) :$ 

```

## Appendix C

# IRB 140 Dynamic Model



## [Dynamic Model of the IRB 140

```

> restart :
> with(LinearAlgebra) :
>
> #Defining joint variables
> q := Vector( [ [q1(t)], [q2(t)], [q3(t)], [q4(t)], [q5(t)], [q6(t)] ] ) :
> Dq := map(diff, q, t) :
> DDq := map(diff, Dq, t) :
>
> #Defining link length vectors
> r0c1 := Vector( [ [0.014], [-0.264], [0.067] ] ) :
> r1c2 := Vector( [ [0.201], [0], [-0.070] ] ) :
> r2c3 := Vector( [ [0], [0], [0] ] ) :
> r3c4 := Vector( [ [0], [0.080], [0] ] ) :
> r4c5 := Vector( [ [0], [0], [0] ] ) :
> r5c6 := Vector( [ [0], [0], [0.029] ] ) :
> r0l := Vector( [ [0.070], [-0.352], [0] ] ) :
> r1l := Vector( [ [0.360], [0], [0] ] ) :
> r2l := Vector( [ [0], [0], [0] ] ) :
> r3l := Vector( [ [0], [0.380], [0] ] ) :
> r4l := Vector( [ [0], [0], [0] ] ) :
> r5l := Vector( [ [0], [0], [0.065] ] ) :
> r1c1 := r0c1 - r0l :
> r2c2 := r1c2 - r1l :
> r3c3 := r2c3 - r2l :
> r4c4 := r3c4 - r3l :
> r5c5 := r4c5 - r4l :
> r6c6 := r5c6 - r5l :
>
> #Gravity vector in inertial frame
> g0 := Vector( [ [0], [0], [-g] ] ) :
>
> #Rotation matrices
> R01 := MatrixMatrixMultiply  $\left( \begin{bmatrix} \cos(q[1]) & -\sin(q[1]) & 0 \\ \sin(q[1]) & \cos(q[1]) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \right) :$ 
> R12 := MatrixMatrixMultiply  $\left( \begin{bmatrix} \cos\left(q[2] - \frac{\pi}{2}\right) & -\sin\left(q[2] - \frac{\pi}{2}\right) & 0 \\ \sin\left(q[2] - \frac{\pi}{2}\right) & \cos\left(q[2] - \frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) :$ 
> R23 := MatrixMatrixMultiply  $\left( \begin{bmatrix} \cos(q[3]) & -\sin(q[3]) & 0 \\ \sin(q[3]) & \cos(q[3]) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \right) :$ 
> R34 := MatrixMatrixMultiply  $\left( \begin{bmatrix} \cos(q[4]) & -\sin(q[4]) & 0 \\ \sin(q[4]) & \cos(q[4]) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \right) :$ 

```

```

> R45 := MatrixMatrixMultiply( [ [ cos(q[5]+pi)  -sin(q[5]+pi)  0 ], [ 1  0  0 ],
                                [ sin(q[5]+pi)   cos(q[5]+pi)  0 ], [ 0  0 -1 ],
                                [ 0             0             1 ], [ 0  1  0 ] ] ) :
> R56 := MatrixMatrixMultiply( [ [ cos(q[6])  -sin(q[6])  0 ], [ 1  0  0 ],
                                [ sin(q[6])   cos(q[6])  0 ], [ 0  1  0 ],
                                [ 0             0             1 ], [ 0  0  1 ] ] ) :
> R02 := MatrixMatrixMultiply( R01, R12 ) :
> R03 := MatrixMatrixMultiply( R02, R23 ) :
> R04 := MatrixMatrixMultiply( R03, R34 ) :
> R05 := MatrixMatrixMultiply( R04, R45 ) :
> R06 := MatrixMatrixMultiply( R05, R56 ) :
>
> #Axis of rotation of joint i expressed in frame i
> z0 := Vector( [ [0], [0], [1]] ) :
> b1 := MatrixMatrixMultiply( Transpose( R01 ), z0 ) :
> b2 := combine( MatrixMatrixMultiply( Transpose( R02 ), MatrixVectorMultiply( R01, z0 ) ), trig ) :
> b3 := combine( MatrixMatrixMultiply( Transpose( R03 ), MatrixVectorMultiply( R02, z0 ) ), trig ) :
> b4 := combine( MatrixMatrixMultiply( Transpose( R04 ), MatrixVectorMultiply( R03, z0 ) ), trig ) :
> b5 := combine( MatrixMatrixMultiply( Transpose( R05 ), MatrixVectorMultiply( R04, z0 ) ), trig ) :
> b6 := combine( MatrixMatrixMultiply( Transpose( R06 ), MatrixVectorMultiply( R05, z0 ) ), trig ) :
>
> #Link Masses
> m1 := 27 :
> m2 := 22 :
> m3 := 0 :
> m4 := 25 :
> m5 := 0 :
> m6 := 1 :
>
> #Cylinder link dimensions
> r1 := 0.147 :
> r2 := 0.108 :
> r3 := 0 :
> r4 := 0.094 :
> r5 := 0 :
> r6 := 0.054 :
>
> h1 := 0.264 :
> h2 := 0.402 :
> h3 := 0 :
> h4 := 0.600 :
> h5 := 0 :
> h6 := 0.072 :
>
> #Inertia matrices

```

$$I1 := \begin{bmatrix} \frac{1}{12} \cdot m1 \cdot (3 \cdot r1^2 + h1^2) & 0 & 0 \\ 0 & \frac{1}{2} \cdot m1 \cdot r1^2 & 0 \\ 0 & 0 & \frac{1}{12} \cdot m1 \cdot (3 \cdot r1^2 + h1^2) \end{bmatrix} :$$

$$I2 := \begin{bmatrix} \frac{1}{2} \cdot m2 \cdot r2^2 & 0 & 0 \\ 0 & \frac{1}{12} \cdot m2 \cdot (3 \cdot r2^2 + h2^2) & 0 \\ 0 & 0 & \frac{1}{12} \cdot m2 \cdot (3 \cdot r2^2 + h2^2) \end{bmatrix} :$$

$$I3 := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} :$$

$$I4 := \begin{bmatrix} \frac{1}{12} \cdot m4 \cdot (3 \cdot r4^2 + h4^2) & 0 & 0 \\ 0 & \frac{1}{2} \cdot m4 \cdot r4^2 & 0 \\ 0 & 0 & \frac{1}{12} \cdot m4 \cdot (3 \cdot r4^2 + h4^2) \end{bmatrix} :$$

$$I5 := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} :$$

$$I6 := \begin{bmatrix} \frac{1}{12} \cdot m6 \cdot (3 \cdot r6^2 + h6^2) & 0 & 0 \\ 0 & \frac{1}{12} \cdot m6 \cdot (3 \cdot r6^2 + h6^2) & 0 \\ 0 & 0 & \frac{1}{2} \cdot m6 \cdot r6^2 \end{bmatrix} :$$

>

> **#Forward recursion: Link 1**

>  $\omega1 := b1 \cdot Dq[1] :$

>  $\alpha1 := b1 \cdot DDq[1] + \text{CrossProduct}(\omega1, b1 \cdot Dq[1]) :$

>  $D\omega1 := \text{map}(\text{diff}, \omega1, t) :$

>  $ae1 := \text{CrossProduct}(D\omega1, r01) + \text{CrossProduct}(\omega1, \text{CrossProduct}(\omega1, r01)) :$

>  $ac1 := \text{CrossProduct}(D\omega1, r0c1) + \text{CrossProduct}(\omega1, \text{CrossProduct}(\omega1, r0c1)) :$

>  $g1 := \text{MatrixVectorMultiply}(\text{Transpose}(R01), g0) :$

>

> **#Forward recursion: Link 2**

>  $\omega2 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R12), \omega1) + b2 \cdot Dq[2], \text{trig}) :$

>  $\alpha2 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R12), \alpha1) + b2 \cdot DDq[2] + \text{CrossProduct}(\omega2, b2 \cdot Dq[2]), \text{trig}) :$

>  $D\omega2 := \text{map}(\text{diff}, \omega2, t) :$

>  $ae2 := \text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(R12), ae1) + \text{CrossProduct}(D\omega2, r12) + \text{CrossProduct}(\omega2,$

```

CrossProduct( $\omega_2$ ,  $r_{12}$ ), trig) :
> ac2 := combine(MatrixVectorMultiply(Transpose(R12), ae1) + CrossProduct(D $\omega_2$ ,  $r_{1c2}$ )
+ CrossProduct( $\omega_2$ , CrossProduct( $\omega_2$ ,  $r_{1c2}$ )), trig) :
> g2 := combine(MatrixVectorMultiply(Transpose(R02), g0), trig) :
>
> #Forward recursion: Link 3
>  $\omega_3$  := combine(MatrixVectorMultiply(Transpose(R23),  $\omega_2$ ) +  $b_3 \cdot Dq[3]$ , trig) :
>  $\alpha_3$  := combine(MatrixVectorMultiply(Transpose(R23),  $\alpha_2$ ) +  $b_3 \cdot DDq[3]$  + CrossProduct( $\omega_3$ ,  $b_3 \cdot Dq[3]$ ),
trig) :
> D $\omega_3$  := map(diff,  $\omega_3$ , t) :
> ae3 := combine(MatrixVectorMultiply(Transpose(R23), ae2), trig) :
> ac3 := combine(MatrixVectorMultiply(Transpose(R23), ae2), trig) :
> g3 := combine(MatrixVectorMultiply(Transpose(R03), g0), trig) :
>
> #Forward recursion: Link 4
>  $\omega_4$  := combine(MatrixVectorMultiply(Transpose(R34),  $\omega_3$ ) +  $b_4 \cdot Dq[4]$ , trig) :
>  $\alpha_4$  := combine(MatrixVectorMultiply(Transpose(R34),  $\alpha_3$ ) +  $b_4 \cdot DDq[4]$  + CrossProduct( $\omega_4$ ,  $b_4 \cdot Dq[4]$ ),
trig) :
> D $\omega_4$  := map(diff,  $\omega_4$ , t) :
> ae4 := combine(MatrixVectorMultiply(Transpose(R34), ae3) + CrossProduct(D $\omega_4$ ,  $r_{34}$ ) + CrossProduct( $\omega_4$ ,
CrossProduct( $\omega_4$ ,  $r_{34}$ )), trig) :
> ac4 := combine(MatrixVectorMultiply(Transpose(R34), ae3) + CrossProduct(D $\omega_4$ ,  $r_{3c4}$ )
+ CrossProduct( $\omega_4$ , CrossProduct( $\omega_4$ ,  $r_{3c4}$ )), trig) :
> g4 := combine(MatrixVectorMultiply(Transpose(R04), g0), trig) :
>
> #Forward recursion: Link 5
>  $\omega_5$  := combine(MatrixVectorMultiply(Transpose(R45),  $\omega_4$ ) +  $b_5 \cdot Dq[5]$ , trig) :
>  $\alpha_5$  := combine(MatrixVectorMultiply(Transpose(R45),  $\alpha_4$ ) +  $b_5 \cdot DDq[5]$  + CrossProduct( $\omega_5$ ,  $b_5 \cdot Dq[5]$ ),
trig) :
> D $\omega_5$  := map(diff,  $\omega_5$ , t) :
> ae5 := combine(MatrixVectorMultiply(Transpose(R45), ae4), trig) :
> ac5 := combine(MatrixVectorMultiply(Transpose(R45), ae4), trig) :
> g5 := combine(MatrixVectorMultiply(Transpose(R05), g0), trig) :
>
> #Forward recursion: Link 6
>  $\omega_6$  := combine(MatrixVectorMultiply(Transpose(R56),  $\omega_5$ ) +  $b_6 \cdot Dq[6]$ , trig) :
>  $\alpha_6$  := combine(MatrixVectorMultiply(Transpose(R56),  $\alpha_5$ ) +  $b_6 \cdot DDq[6]$  + CrossProduct( $\omega_6$ ,  $b_6 \cdot Dq[6]$ ),
trig) :
> D $\omega_6$  := map(diff,  $\omega_6$ , t) :
> ae6 := combine(MatrixVectorMultiply(Transpose(R56), ae5) + CrossProduct(D $\omega_6$ ,  $r_{56}$ ) + CrossProduct( $\omega_6$ ,
CrossProduct( $\omega_6$ ,  $r_{56}$ )), trig) :
> ac6 := combine(MatrixVectorMultiply(Transpose(R56), ae5) + CrossProduct(D $\omega_6$ ,  $r_{5c6}$ )
+ CrossProduct( $\omega_6$ , CrossProduct( $\omega_6$ ,  $r_{5c6}$ )), trig) :
> g6 := combine(MatrixVectorMultiply(Transpose(R06), g0), trig) :
>
> #Backward recursion: Link 6
> f6 := Add( $m_6 \cdot ac_6$ ,  $-m_6 \cdot g_6$ ) :
>  $\tau_6$  := -CrossProduct(f6,  $r_{5c6}$ ) + MatrixVectorMultiply(I6,  $\alpha_6$ ) + CrossProduct( $\omega_6$ , MatrixVectorMultiply(I6,
 $\omega_6$ )) :

```

>  $\tau 6z := \text{collect}(\text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(b6), \tau 6), \text{trig}), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\}) :$

> **#Backward recursion: Link 5**

>  $f5 := \text{MatrixVectorMultiply}(R56, f6) :$

>  $\tau 5 := \text{MatrixVectorMultiply}(R56, \tau 6) + \text{MatrixVectorMultiply}(I5, \alpha 5) + \text{CrossProduct}(\omega 5, \text{MatrixVectorMultiply}(I5, \omega 5)) :$

>  $\tau 5y := \text{collect}(\text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(b5), \tau 5), \text{trig}), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\}) :$

> **#Backward recursion: Link 4**

>  $f4 := \text{MatrixVectorMultiply}(R45, f5) + \text{Add}(m4 \cdot ac4, -m4 \cdot g4) :$

>  $\tau 4 := \text{MatrixVectorMultiply}(R45, \tau 5) - \text{CrossProduct}(f4, r3c4) + \text{CrossProduct}(\text{MatrixVectorMultiply}(R45, f5), r4c4) + \text{MatrixVectorMultiply}(I4, \alpha 4) + \text{CrossProduct}(\omega 4, \text{MatrixVectorMultiply}(I4, \omega 4)) :$

>  $\tau 4x := \text{collect}(\text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(b4), \tau 4), \text{trig}), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\}) :$

> **#Backward recursion: Link 3**

>  $f3 := \text{MatrixVectorMultiply}(R34, f4) :$

>  $\tau 3 := \text{MatrixVectorMultiply}(R34, \tau 4) + \text{MatrixVectorMultiply}(I3, \alpha 3) + \text{CrossProduct}(\omega 3, \text{MatrixVectorMultiply}(I3, \omega 3)) :$

>  $\tau 3y := \text{collect}(\text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(b3), \tau 3), \text{trig}), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\}) :$

> **#Backward recursion: Link 2**

>  $f2 := \text{MatrixVectorMultiply}(R23, f3) + \text{Add}(m2 \cdot ac2, -m2 \cdot g2) :$

>  $\tau 2 := \text{MatrixVectorMultiply}(R23, \tau 3) - \text{CrossProduct}(f2, r1c2) + \text{CrossProduct}(\text{MatrixVectorMultiply}(R23, f3), r2c2) + \text{MatrixVectorMultiply}(I2, \alpha 2) + \text{CrossProduct}(\omega 2, \text{MatrixVectorMultiply}(I2, \omega 2)) :$

>  $\tau 2z := \text{collect}(\text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(b2), \tau 2), \text{trig}), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\}) :$

> **#Backward recursion: Link 1**

>  $f1 := \text{MatrixVectorMultiply}(R12, f2) + \text{Add}(m1 \cdot ac1, -m1 \cdot g1) :$

>  $\tau 1 := \text{MatrixVectorMultiply}(R12, \tau 2) - \text{CrossProduct}(f1, r0c1) + \text{CrossProduct}(\text{MatrixVectorMultiply}(R12, f2), r1c1) + \text{MatrixVectorMultiply}(I1, \alpha 1) + \text{CrossProduct}(\omega 1, \text{MatrixVectorMultiply}(I1, \omega 1)) :$

>  $\tau 1x := \text{collect}(\text{combine}(\text{MatrixVectorMultiply}(\text{Transpose}(b1), \tau 1), \text{trig}), \{DDq[1], DDq[2], DDq[3], DDq[4], DDq[5], DDq[6], Dq[1], Dq[2], Dq[3], Dq[4], Dq[5], Dq[6]\})$

$$\begin{aligned} \tau 1x := & \left( -1.28749007982654 \cdot 10^{-19} \sin\left(q2(t) + \frac{3\pi}{2} + 3q3(t) + 2q5(t) + q4(t)\right) + 1.28749007982654 \cdot 10^{-19} \sin\left(3q2(t) \right. \right. \\ & \left. \left. + \frac{\pi}{2} + 2q5(t) + q4(t) + q3(t)\right) + 1.72794721239877 \cdot 10^{-19} \sin\left(-3q2(t) + \frac{7\pi}{2} - 3q3(t) + 2q5(t) + 2q4(t)\right) \right. \\ & \left. + 1.08420217248550 \cdot 10^{-19} \sin\left(-3q2(t) + \frac{5\pi}{2} - 3q3(t) + q5(t) + q4(t)\right) + 1.32137139771671 \cdot 10^{-19} \sin\left(-q4(t) \right. \right. \\ & \left. \left. + \frac{3\pi}{2} + 2q5(t) + q3(t) + q2(t)\right) + 2.59192081859816 \cdot 10^{-19} \sin\left(2q5(t) + \frac{3\pi}{2} - 2q4(t) + q3(t) + q2(t)\right) \right. \\ & \left. + 2.60886147754325 \cdot 10^{-19} \sin\left(-q2(t) + \frac{5\pi}{2} - 3q3(t) + q4(t) + 2q5(t)\right) + 1.08420217248550 \cdot 10^{-19} \sin\left(-q3(t) \right. \right. \\ & \left. \left. + q5(t) + \frac{\pi}{2} + q4(t) + q2(t)\right) + 1.08420217248550 \cdot 10^{-19} \sin\left(q3(t) + q5(t) + \frac{3\pi}{2} - q4(t) - q2(t)\right) \right. \\ & \left. - 1.72794721239877 \cdot 10^{-19} \sin\left(2q5(t) + \frac{5\pi}{2} + 2q4(t) - q3(t) - q2(t)\right) + 2.60886147754325 \cdot 10^{-19} \sin\left(2q5(t) \right. \right. \end{aligned} \quad (1)$$

$$\begin{aligned}
& + \frac{5\pi}{2} - q^3(t) - q^2(t) \Big) - 1.08420217248550 \cdot 10^{-19} \sin\left(q^5(t) + \frac{3\pi}{2} - 2q^4(t) - q^3(t) - q^2(t)\right) \\
& - 1.08420217248550 \cdot 10^{-19} \sin\left(-3q^2(t) + \frac{\pi}{2} - 3q^3(t) - q^5(t) + 2q^4(t)\right) + 1.08420217248550 \cdot 10^{-19} \sin\left(-3q^2(t) \right. \\
& + \left. \frac{\pi}{2} - 3q^3(t) - q^5(t) - q^4(t)\right) + 4.40457132572236 \cdot 10^{-20} \sin\left(2q^4(t) + 2q^5(t) + \frac{3\pi}{2} + q^3(t) + q^2(t)\right) \\
& + 4.40457132572236 \cdot 10^{-20} \sin\left(q^2(t) + \frac{3\pi}{2} + 3q^3(t) + 2q^5(t) + 2q^4(t)\right) - 4.40457132572236 \cdot 10^{-20} \sin\left(3q^2(t) \right. \\
& + \left. \frac{\pi}{2} + 3q^3(t) + 2q^5(t) + 2q^4(t)\right) - 4.23516473627150 \cdot 10^{-20} \sin\left(3q^2(t) + \frac{\pi}{2} + 2q^5(t) + q^3(t)\right) \\
& + 4.23516473627150 \cdot 10^{-20} \sin\left(q^2(t) + \frac{3\pi}{2} + 3q^3(t) + 2q^5(t)\right) + 3.03237795117040 \cdot 10^{-19} \sin\left(3q^2(t) + \frac{\pi}{2} \right. \\
& + \left. 3q^3(t) + 2q^5(t)\right) - 8.80914265144472 \cdot 10^{-20} \sin\left(-2q^4(t) + 2q^5(t) + \frac{5\pi}{2} - q^3(t) - q^2(t)\right) \\
& - 3.81503639443337 \cdot 10^{-18} \sin\left(-2q^4(t) - q^3(t) - q^2(t) + \frac{\pi}{2}\right) + 8.80914265144472 \cdot 10^{-20} \sin\left(-3q^2(t) + \frac{7\pi}{2} \right. \\
& - \left. 3q^3(t) + 2q^5(t) - 2q^4(t)\right) - 3.03237795117040 \cdot 10^{-19} \sin\left(2q^5(t) + \frac{3\pi}{2} + q^3(t) + q^2(t)\right) \\
& - 1.08420217248550 \cdot 10^{-19} \sin\left(q^5(t) + \frac{\pi}{2} + q^3(t) + q^2(t) - q^4(t)\right) + 1.08420217248550 \cdot 10^{-19} \sin\left(q^5(t) + \frac{\pi}{2} \right. \\
& + \left. q^3(t) + q^2(t) + q^4(t)\right) - 2.60886147754325 \cdot 10^{-19} \sin\left(-3q^2(t) + \frac{7\pi}{2} - 3q^3(t) + 2q^5(t)\right) \\
& - 1.08420217248550 \cdot 10^{-19} \sin\left(-q^3(t) + q^5(t) + \frac{\pi}{2} - q^4(t) + q^2(t)\right) - 1.08420217248550 \cdot 10^{-19} \sin\left(q^3(t) + q^5(t) \right. \\
& + \left. \frac{3\pi}{2} + q^4(t) - q^2(t)\right) - 2.16840434497101 \cdot 10^{-19} \cos\left(q^5(t) + \frac{\pi}{2} + q^4(t) + q^2(t)\right) \\
& + 2.16840434497101 \cdot 10^{-19} \cos\left(q^5(t) + \frac{3\pi}{2} - q^4(t) - q^2(t)\right) + 3.93023287525995 \cdot 10^{-19} \sin\left(3q^2(t) + \frac{\pi}{2} + 3q^3(t) \right. \\
& + \left. 2q^5(t) + q^4(t)\right) + 1.08420217248550 \cdot 10^{-19} \sin\left(-3q^2(t) + \frac{5\pi}{2} - 3q^3(t) - 2q^4(t) + q^5(t)\right) \\
& - 2.60886147754325 \cdot 10^{-19} \sin\left(-3q^2(t) + \frac{7\pi}{2} - 3q^3(t) + q^4(t) + 2q^5(t)\right) - 1.28749007982654 \cdot 10^{-19} \sin\left(3q^2(t) \right. \\
& + \left. \frac{\pi}{2} + 2q^5(t) - q^4(t) + q^3(t)\right) + 1.28749007982654 \cdot 10^{-19} \sin\left(q^2(t) + \frac{3\pi}{2} + 3q^3(t) - q^4(t) + 2q^5(t)\right) \\
& - 8.47032947254300 \cdot 10^{-20} \sin\left(-q^2(t) + \frac{5\pi}{2} + 2q^5(t) + q^4(t) + q^3(t)\right) + 8.47032947254300 \cdot 10^{-20} \sin\left(-q^2(t) \right. \\
& + \left. \frac{5\pi}{2} + 2q^5(t) - q^4(t) + q^3(t)\right) + 1.08420217248550 \cdot 10^{-19} \sin\left(2q^4(t) - q^3(t) - q^2(t) + \frac{3\pi}{2} + q^5(t)\right) \\
& - 6.93889390390723 \cdot 10^{-18} \sin\left(-q^2(t) + \frac{\pi}{2} - 3q^3(t)\right) + 3.46944695195361 \cdot 10^{-18} \sin\left(-q^2(t) + \frac{\pi}{2} - 3q^3(t) \right. \\
& - \left. 2q^4(t)\right) - 1.57853060050311 \cdot 10^{-17} \sin\left(-3q^2(t) + \frac{3\pi}{2} - 2q^4(t) - q^3(t)\right) - 1.08420217248550 \cdot 10^{-19} \sin\left( \right. \\
& - \left. 3q^2(t) + \frac{5\pi}{2} - 3q^3(t) + 2q^4(t) + q^5(t)\right) + 1.08420217248550 \cdot 10^{-19} \sin\left(2q^4(t) + \frac{\pi}{2} + q^5(t) + q^3(t) + q^2(t)\right) \\
& - 3.93023287525995 \cdot 10^{-19} \sin\left(3q^2(t) + \frac{\pi}{2} + 3q^3(t) - q^4(t) + 2q^5(t)\right) - 1.08420217248550 \cdot 10^{-19} \sin\left(q^5(t) + \frac{\pi}{2} \right. \\
& - \left. 2q^4(t) + q^3(t) + q^2(t)\right) - 6.59330446142747 \cdot 10^{-18} \sin\left(-3q^2(t) + \frac{3\pi}{2} - 3q^3(t) - 2q^4(t)\right) \\
& + 1.08420217248550 \cdot 10^{-19} \sin\left(-3q^2(t) + \frac{\pi}{2} - 3q^3(t) - 2q^4(t) - q^5(t)\right) + 8.47032947254300 \cdot 10^{-20} \sin\left(q^2(t) \right. \\
& + \left. \frac{3\pi}{2} + 2q^5(t) + q^4(t) - q^3(t)\right) - 3.46944695195361 \cdot 10^{-18} \sin\left(-q^2(t) + \frac{\pi}{2} - 3q^3(t) + 2q^4(t)\right)
\end{aligned}$$

$$\begin{aligned}
& + 6.93889390390723 \cdot 10^{-18} \sin\left(-3q_2(t) + \frac{3\pi}{2} - q_3(t)\right) + 2.16840434497101 \cdot 10^{-19} \cos\left(q_5(t) + \frac{\pi}{2} - q_4(t) \right. \\
& \left. + q_2(t)\right) - 2.16840434497101 \cdot 10^{-19} \cos\left(q_5(t) + \frac{3\pi}{2} + q_4(t) - q_2(t)\right) - 2.60886147754325 \cdot 10^{-19} \sin\left(-3q_2(t) \right. \\
& \left. + \frac{7\pi}{2} + 2q_5(t) + q_4(t) - q_3(t)\right) - 1.08420217248550 \cdot 10^{-19} \sin\left(-3q_2(t) + \frac{\pi}{2} - 3q_3(t) - q_5(t) + q_4(t)\right) \\
& - 4.40457132572236 \cdot 10^{-20} \sin\left(3q_2(t) + \frac{\pi}{2} + q_3(t) + 2q_5(t) + 2q_4(t)\right) - 1.32137139771671 \cdot 10^{-19} \sin\left(2q_5(t) \right. \\
& \left. + \frac{3\pi}{2} + q_4(t) + q_3(t) + q_2(t)\right) + 2.60886147754325 \cdot 10^{-19} \sin\left(-3q_2(t) + \frac{7\pi}{2} - 3q_3(t) - q_4(t) + 2q_5(t)\right) \\
& - 8.63973606199386 \cdot 10^{-20} \sin\left(q_2(t) + \frac{3\pi}{2} + 3q_3(t) + 2q_5(t) - 2q_4(t)\right) - 2.59192081859816 \cdot 10^{-19} \sin\left(3q_2(t) \right. \\
& \left. + \frac{\pi}{2} + 3q_3(t) + 2q_5(t) - 2q_4(t)\right) - 1.08420217248550 \cdot 10^{-19} \sin\left(-3q_2(t) + \frac{5\pi}{2} - 3q_3(t) + q_5(t) - q_4(t)\right) \\
& - 8.47032947254300 \cdot 10^{-20} \sin\left(q_2(t) + \frac{3\pi}{2} + 2q_5(t) - q_4(t) - q_3(t)\right) + 8.63973606199386 \cdot 10^{-20} \sin\left(3q_2(t) + \frac{\pi}{2} \right. \\
& \left. + q_3(t) + 2q_5(t) - 2q_4(t)\right) + 6.93889390390723 \cdot 10^{-18} \sin\left(2q_4(t) - q_3(t) - q_2(t) + \frac{\pi}{2}\right) \\
& + 1.23158590530775 \cdot 10^{-17} \sin\left(-3q_2(t) + \frac{3\pi}{2} + 2q_4(t) - q_3(t)\right) + 2.60886147754325 \cdot 10^{-19} \sin\left(-3q_2(t) + \frac{7\pi}{2} \right. \\
& \left. + 2q_5(t) - q_4(t) - q_3(t)\right) - 1.08420217248550 \cdot 10^{-19} \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) + q_4(t)\right) \\
& + 1.08420217248550 \cdot 10^{-19} \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) - q_4(t)\right) - 2.60886147754325 \cdot 10^{-19} \sin\left(-q_2(t) + \frac{5\pi}{2} \right. \\
& \left. - 3q_3(t) - q_4(t) + 2q_5(t)\right) - 3.45589442479755 \cdot 10^{-19} \sin\left(-3q_2(t) + \frac{3\pi}{2} - 3q_3(t) + 2q_4(t)\right) \Big) \dot{q}I(t)^2 \\
& + \left( \left( 6.50521303491303 \cdot 10^{-19} \sin(2q_4(t) + q_5(t) + 2\pi - 2q_3(t) - 2q_2(t)) + 4.49054400000000 \sin(-2q_2(t) + \pi) \right. \right. \\
& - 0.999311000000000 \sin(-2q_3(t) - 2q_2(t) + \pi) + 0.0001360000000 \sin(2q_5(t) + \pi + q_4(t) + 2q_3(t) + 2q_2(t)) \\
& + 0.00003400000000000005 \sin(2q_5(t) + \pi - 2q_4(t) + 2q_3(t) + 2q_2(t)) + 0.005220000000000000 \cos(q_5(t) + q_4(t) \\
& + q_3(t) + 2q_2(t)) - 0.005220000000000000 \cos(q_5(t) + 2\pi - q_4(t) - q_3(t) - 2q_2(t)) \\
& + 0.010440000000000000 \cos(q_5(t) + q_3(t) + 2q_2(t)) + 0.010440000000000000 \cos(q_5(t) + 2\pi - q_3(t) - 2q_2(t)) \\
& - 0.011020000000000000 \sin(q_5(t) + 2q_3(t) + 2q_2(t)) + 0.0000679999999999722 \sin(2q_4(t) - 2q_3(t) - 2q_2(t) + \pi) \\
& + 6.50521303491303 \cdot 10^{-19} \sin(-2q_4(t) + q_5(t) + 2\pi - 2q_3(t) - 2q_2(t)) - 0.0002039999999999999 \sin(2q_5(t) + 3\pi \\
& - 2q_3(t) - 2q_2(t)) - 1.713600000000000 \cos(-q_3(t) - 2q_2(t) + \pi) + 8.67361737988404 \cdot 10^{-19} \cos(q_5(t) + \pi \\
& - q_3(t)) + 8.67361737988404 \cdot 10^{-19} \cos(q_5(t) + \pi + q_3(t)) + 0.005220000000000000 \cos(q_5(t) - q_4(t) + q_3(t) \\
& + 2q_2(t)) - 0.005220000000000000 \cos(q_5(t) + 2\pi + q_4(t) - q_3(t) - 2q_2(t)) - 0.005510000000000000 \sin(q_5(t) \\
& + 2\pi - 2q_3(t) - 2q_2(t) + q_4(t)) + 0.0001360000000 \sin(q_4(t) + 2q_5(t) + 3\pi - 2q_3(t) - 2q_2(t)) \\
& + 1.929480000000000 \sin\left(-q_2(t) + \frac{\pi}{2}\right) - 4.33680868994202 \cdot 10^{-19} \sin(2q_4(t) + q_5(t) + 2q_3(t) + 2q_2(t)) \\
& - 1.76182853028894 \cdot 10^{-19} \sin(2\pi + 2q_3(t) + 2q_5(t)) + 2.77555756156289 \cdot 10^{-17} \sin(2q_3(t) - 2q_4(t))
\end{aligned}$$

$$\begin{aligned}
& + 1.72794721239877 \cdot 10^{-19} \sin(-2q_4(t) + 2\pi + 2q_5(t)) + 1.11022302462516 \cdot 10^{-16} \cos(q_3(t)) \\
& + 0.00006800000000 \sin(-2q_4(t) - 2q_3(t) - 2q_2(t) + \pi) + 0.0020300000000000 \cos\left(q_5(t) + \frac{3\pi}{2} - q_3(t) \right. \\
& \left. - q_2(t)\right) - 0.000034000000000007 \sin(2q_5(t) + 3\pi - 2q_3(t) - 2q_2(t) - 2q_4(t)) \\
& + 0.0000340000000000007 \sin(2q_5(t) + \pi + 2q_3(t) + 2q_2(t) + 2q_4(t)) + 2.16840434497101 \cdot 10^{-19} \sin(-2q_2(t) \\
& + 2\pi + 2q_4(t) + q_5(t)) + 1.73472347597681 \cdot 10^{-18} \sin(q_5(t) + \pi) + 1.72794721239877 \cdot 10^{-19} \sin(2\pi + 2q_3(t) \\
& + 2q_5(t) - 2q_4(t)) + 1.76182853028894 \cdot 10^{-19} \sin(2q_2(t) + \pi + 2q_5(t)) + 0.00013600000000 \sin(-q_4(t) + 2q_5(t) \\
& + \pi + 2q_3(t) + 2q_2(t)) + 0.001015000000000000 \cos\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) - q_4(t)\right) \\
& - 0.001015000000000000 \cos\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) + q_4(t)\right) + 0.001015000000000000 \cos\left(q_5(t) + \frac{\pi}{2} + q_3(t) \right. \\
& \left. + q_2(t) + q_4(t)\right) - 0.001015000000000000 \cos\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) - q_4(t)\right) \\
& - 4.33680868994202 \cdot 10^{-19} \sin(q_5(t) - 2q_4(t) + 2q_3(t) + 2q_2(t)) - 1.76182853028894 \cdot 10^{-19} \sin(2\pi + 2q_5(t)) \\
& + 0.000203999999999999 \sin(2q_5(t) + \pi + 2q_3(t) + 2q_2(t)) - 4.33680868994202 \cdot 10^{-19} \cos(q_5(t) + \pi - q_4(t) \\
& - q_3(t)) + 4.33680868994202 \cdot 10^{-19} \cos(q_5(t) + \pi + q_4(t) + q_3(t)) - 4.33680868994202 \cdot 10^{-19} \cos(q_5(t) + \pi + q_4(t) \\
& - q_3(t)) + 4.33680868994202 \cdot 10^{-19} \cos(q_5(t) + \pi - q_4(t) + q_3(t)) - 0.005510000000000000 \sin(q_5(t) + 2\pi \\
& - 2q_3(t) - 2q_2(t) - q_4(t)) - 0.3332000000000000 \cos\left(-q_3(t) - q_2(t) + \frac{\pi}{2}\right) + 0.002030000000000000 \cos\left(q_5(t) \right. \\
& \left. + \frac{\pi}{2} + q_3(t) + q_2(t)\right) + 2.16840434497101 \cdot 10^{-19} \sin(\pi - 2q_3(t) + q_5(t) - 2q_4(t)) - 2.16840434497101 \cdot 10^{-19} \sin \\
& (-2q_2(t) + 2\pi - 2q_4(t) + q_5(t)) - 1.72794721239877 \cdot 10^{-19} \sin(2q_2(t) + \pi + 2q_5(t) - 2q_4(t)) \\
& - 2.16840434497101 \cdot 10^{-19} \sin(\pi - 2q_3(t) + 2q_4(t) + q_5(t)) - 2.15078605966812 \cdot 10^{-17} \sin(-2q_2(t) + \pi - 2q_4(t)) \\
& + 0.011020000000000000 \sin(q_5(t) + 2\pi - 2q_3(t) - 2q_2(t)) - 2.16840434497101 \cdot 10^{-19} \sin(q_5(t) + \pi - 2q_4(t)) \\
& - 0.005510000000000000 \sin(q_5(t) + 2q_3(t) + 2q_2(t) - q_4(t)) - 0.005510000000000000 \sin(q_5(t) + 2q_3(t) + 2q_2(t) \\
& + q_4(t)) - 0.0000340000000000007 \sin(2q_5(t) + 3\pi + 2q_4(t) - 2q_3(t) - 2q_2(t)) \\
& - 2.16840434497101 \cdot 10^{-19} \sin(2q_4(t) + \pi + q_5(t)) - 2.15078605966812 \cdot 10^{-17} \sin(-2q_2(t) + \pi + 2q_4(t)) \\
& + 0.00013600000000 \sin(-q_4(t) + 2q_5(t) + 3\pi - 2q_3(t) - 2q_2(t)) \dot{q}_2(t) + \left(6.50521303491303 \cdot 10^{-19} \sin(2q_4(t) \right. \\
& \left. + q_5(t) + 2\pi - 2q_3(t) - 2q_2(t)) - 0.9993110000000000 \sin(-2q_3(t) - 2q_2(t) + \pi) + 0.00013600000000 \sin(2q_5(t) \right. \\
& \left. + \pi + q_4(t) + 2q_3(t) + 2q_2(t)) + 0.0000340000000000005 \sin(2q_5(t) + \pi - 2q_4(t) + 2q_3(t) + 2q_2(t)) \right. \\
& \left. + 0.002610000000000000 \cos(q_5(t) + q_4(t) + q_3(t) + 2q_2(t)) - 0.002610000000000000 \cos(q_5(t) + 2\pi - q_4(t) \right. \\
& \left. - q_3(t) - 2q_2(t)) + 1.72794721239877 \cdot 10^{-19} \sin(2q_5(t) + 2\pi + 2q_4(t)) + 0.005220000000000000 \cos(q_5(t) + q_3(t)) \right)
\end{aligned}$$

$$\begin{aligned}
& + 2q_2(t) + 0.0052200000000000 \cos(q_5(t) + 2\pi - q_3(t) - 2q_2(t)) - 0.0110200000000000 \sin(q_5(t) + 2q_3(t) \\
& + 2q_2(t) + 0.00006800000000000278 \sin(2q_4(t) - 2q_3(t) - 2q_2(t) + \pi) + 6.50521303491303 \cdot 10^{-19} \sin(-2q_4(t) \\
& + q_5(t) + 2\pi - 2q_3(t) - 2q_2(t)) - 0.00020399999999999999 \sin(2q_5(t) + 3\pi - 2q_3(t) - 2q_2(t)) \\
& - 0.8568000000000000 \cos(-q_3(t) - 2q_2(t) + \pi) - 2.77555756156289 \cdot 10^{-17} \sin(2q_3(t) + 2q_4(t)) \\
& + 0.005220000000000000 \cos(q_5(t) + \pi - q_3(t)) + 0.005220000000000000 \cos(q_5(t) + \pi + q_3(t)) \\
& + 0.002610000000000000 \cos(q_5(t) - q_4(t) + q_3(t) + 2q_2(t)) - 0.002610000000000000 \cos(q_5(t) + 2\pi + q_4(t) \\
& - q_3(t) - 2q_2(t)) - 0.005510000000000000 \sin(q_5(t) + 2\pi - 2q_3(t) - 2q_2(t) + q_4(t)) + 0.00013600000000 \sin(q_4(t) \\
& + 2q_5(t) + 3\pi - 2q_3(t) - 2q_2(t)) - 4.33680868994202 \cdot 10^{-19} \sin(2q_4(t) + q_5(t) + 2q_3(t) + 2q_2(t)) \\
& + 1.62630325872826 \cdot 10^{-19} \sin(2\pi + 2q_3(t) + 2q_5(t)) + 2.77555756156289 \cdot 10^{-17} \sin(2q_3(t) - 2q_4(t)) \\
& - 1.72794721239877 \cdot 10^{-19} \sin(2\pi - 2q_3(t) + 2q_5(t) + 2q_4(t)) + 1.72794721239877 \cdot 10^{-19} \sin(-2q_2(t) + 3\pi \\
& + 2q_5(t) + 2q_4(t)) - 0.8568000000000000 \cos(q_3(t)) + 0.00006800000000000278 \sin(-2q_4(t) - 2q_3(t) - 2q_2(t) \\
& + \pi) + 0.002030000000000000 \cos\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t)\right) - 0.0000340000000000005 \sin(2q_5(t) + 3\pi \\
& - 2q_3(t) - 2q_2(t) - 2q_4(t)) + 0.00003400000000000003 \sin(2q_5(t) + \pi + 2q_3(t) + 2q_2(t) + 2q_4(t)) \\
& - 2.16840434497101 \cdot 10^{-19} \sin(-2q_2(t) + 2\pi + 2q_4(t) + q_5(t)) + 8.67361737988404 \cdot 10^{-19} \sin(q_5(t) + \pi) \\
& - 1.72794721239877 \cdot 10^{-19} \sin(2\pi - 2q_3(t) - 2q_4(t) + 2q_5(t)) + 1.72794721239877 \cdot 10^{-19} \sin(-2q_2(t) + 3\pi \\
& - 2q_4(t) + 2q_5(t)) - 1.72794721239877 \cdot 10^{-19} \sin(2\pi + 2q_3(t) + 2q_5(t) - 2q_4(t)) \\
& - 1.62630325872826 \cdot 10^{-19} \sin(2q_2(t) + \pi + 2q_5(t)) + 0.00013600000000 \sin(-q_4(t) + 2q_5(t) + \pi + 2q_3(t) \\
& + 2q_2(t) + 0.001015000000000000 \cos\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) - q_4(t)\right) - 0.001015000000000000 \cos\left(q_5(t) \\
& + \frac{3\pi}{2} - q_3(t) - q_2(t) + q_4(t)\right) + 0.001015000000000000 \cos\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) + q_4(t)\right) \\
& - 0.001015000000000000 \cos\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) - q_4(t)\right) - 4.33680868994202 \cdot 10^{-19} \sin(q_5(t) - 2q_4(t) \\
& + 2q_3(t) + 2q_2(t)) + 3.38813178901720 \cdot 10^{-19} \sin(2\pi - 2q_3(t) + 2q_5(t)) - 3.38813178901720 \cdot 10^{-19} \sin(-2q_2(t) \\
& + 3\pi + 2q_5(t)) - 1.76182853028894 \cdot 10^{-19} \sin(2\pi + 2q_5(t)) + 0.00020399999999999999 \sin(2q_5(t) + \pi + 2q_3(t) \\
& + 2q_2(t)) - 0.002610000000000000 \cos(q_5(t) + \pi - q_4(t) - q_3(t)) + 0.002610000000000000 \cos(q_5(t) + \pi + q_4(t) \\
& + q_3(t)) - 0.002610000000000000 \cos(q_5(t) + \pi + q_4(t) - q_3(t)) + 0.002610000000000000 \cos(q_5(t) + \pi - q_4(t) \\
& + q_3(t)) - 8.67361737988404 \cdot 10^{-19} \sin(\pi - 2q_3(t) + q_5(t)) + 8.67361737988404 \cdot 10^{-19} \sin(-2q_2(t) + 2\pi + q_5(t)) \\
& - 0.005510000000000000 \sin(q_5(t) + 2\pi - 2q_3(t) - 2q_2(t) - q_4(t)) - 0.3332000000000000 \cos(-q_3(t) - q_2(t)
\end{aligned}$$

$$\begin{aligned}
& + \frac{\pi}{2} \Big) + 0.0020300000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t)\right) + 2.16840434497101 \cdot 10^{-19} \sin(\pi - 2q3(t) + q5(t) \\
& - 2q4(t)) - 2.16840434497101 \cdot 10^{-19} \sin(-2q2(t) + 2\pi - 2q4(t) + q5(t)) + 1.72794721239877 \cdot 10^{-19} \sin(2q2(t) \\
& + \pi + 2q5(t) - 2q4(t)) + 2.16840434497101 \cdot 10^{-19} \sin(\pi - 2q3(t) + 2q4(t) + q5(t)) \\
& - 2.77555756156289 \cdot 10^{-17} \sin(-2q2(t) + \pi - 2q4(t)) + 0.0110200000000000 \sin(q5(t) + 2\pi - 2q3(t) - 2q2(t)) \\
& - 2.16840434497101 \cdot 10^{-19} \sin(q5(t) + \pi - 2q4(t)) - 0.0055100000000000 \sin(q5(t) + 2q3(t) + 2q2(t) - q4(t)) \\
& - 0.0055100000000000 \sin(q5(t) + 2q3(t) + 2q2(t) + q4(t)) - 0.000034000000000005 \sin(2q5(t) + 3\pi + 2q4(t) \\
& - 2q3(t) - 2q2(t)) - 2.16840434497101 \cdot 10^{-19} \sin(2q4(t) + \pi + q5(t)) + 2.77555756156289 \cdot 10^{-17} \sin(-2q2(t) + \pi \\
& + 2q4(t)) + 0.00013600000000 \sin(-q4(t) + 2q5(t) + 3\pi - 2q3(t) - 2q2(t)) \dot{q}3(t) \\
& + \left(1.08420217248550 \cdot 10^{-19} \sin(2q4(t) + q5(t) + 2\pi - 2q3(t) - 2q2(t)) + 0.00006800000000 \sin(2q5(t) + \pi \right. \\
& + q4(t) + 2q3(t) + 2q2(t)) - 0.00003400000000 \sin(2q5(t) + \pi - 2q4(t) + 2q3(t) + 2q2(t)) \\
& + 0.0026100000000000 \cos(q5(t) + q4(t) + q3(t) + 2q2(t)) - 0.0026100000000000 \cos(q5(t) + 2\pi - q4(t) \\
& - q3(t) - 2q2(t)) - 0.00006800000000 \sin(2q5(t) + 2\pi + 2q4(t)) - 0.00006800000000 \sin(2q4(t) - 2q3(t) \\
& - 2q2(t) + \pi) - 1.08420217248550 \cdot 10^{-19} \sin(-2q4(t) + q5(t) + 2\pi - 2q3(t) - 2q2(t)) \\
& - 0.0026100000000000 \cos(q5(t) - q4(t) + q3(t) + 2q2(t)) + 0.0026100000000000 \cos(q5(t) + 2\pi + q4(t) \\
& - q3(t) - 2q2(t)) + 0.0027550000000000 \sin(q5(t) + 2\pi - 2q3(t) - 2q2(t) + q4(t)) \\
& - 0.00006800000000 \sin(q4(t) + 2q5(t) + 3\pi - 2q3(t) - 2q2(t)) + 1.08420217248550 \cdot 10^{-19} \sin(2q4(t) + q5(t) \\
& + 2q3(t) + 2q2(t)) + 0.00006800000000 \sin(-2q4(t) + 2\pi + 2q5(t)) + 0.00006800000000 \sin(-2q4(t) - 2q3(t) \\
& - 2q2(t) + \pi) - 0.00003400000000 \sin(2q5(t) + 3\pi - 2q3(t) - 2q2(t) - 2q4(t)) + 0.00003400000000 \sin(2q5(t) \\
& + \pi + 2q3(t) + 2q2(t) + 2q4(t)) - 0.00006800000000 \sin(-q4(t) + 2q5(t) + \pi + 2q3(t) + 2q2(t)) \\
& - 0.0010150000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right) + 0.0010150000000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \\
& \left. - q2(t) + q4(t)\right) + 0.0010150000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) \\
& - 0.0010150000000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) - q4(t)\right) - 1.08420217248550 \cdot 10^{-19} \sin(q5(t) - 2q4(t) \\
& + 2q3(t) + 2q2(t)) - 0.0026100000000000 \cos(q5(t) + \pi - q4(t) - q3(t)) + 0.0026100000000000 \cos(q5(t) + \pi \\
& + q4(t) + q3(t)) + 0.0026100000000000 \cos(q5(t) + \pi + q4(t) - q3(t)) - 0.0026100000000000 \cos(q5(t) + \pi \\
& - q4(t) + q3(t)) - 0.0027550000000000 \sin(q5(t) + 2\pi - 2q3(t) - 2q2(t) - q4(t)) \\
& + 0.00013600000000 \sin(2q4(t)) + 2.16840434497101 \cdot 10^{-19} \sin(q5(t) + \pi - 2q4(t))
\end{aligned}$$

$$\begin{aligned}
& + 0.0027550000000000 \sin(q5(t) + 2q3(t) + 2q2(t) - q4(t)) - 0.0027550000000000 \sin(q5(t) + 2q3(t) + 2q2(t) \\
& + q4(t)) + 0.00003400000000 \sin(2q5(t) + 3\pi + 2q4(t) - 2q3(t) - 2q2(t)) - 2.16840434497101 \cdot 10^{-19} \sin(2q4(t) \\
& + \pi + q5(t)) + 0.00006800000000 \sin(-q4(t) + 2q5(t) + 3\pi - 2q3(t) - 2q2(t)) \dot{q}4(t) \\
& + \left( 2.16840434497101 \cdot 10^{-19} \sin(2q4(t) + q5(t) + 2\pi - 2q3(t) - 2q2(t)) + 0.00013600000000 \sin(2q5(t) + \pi \right. \\
& + q4(t) + 2q3(t) + 2q2(t)) + 0.00003400000000 \sin(2q5(t) + \pi - 2q4(t) + 2q3(t) + 2q2(t)) \\
& + 0.0026100000000000 \cos(q5(t) + q4(t) + q3(t) + 2q2(t)) + 0.0026100000000000 \cos(q5(t) + 2\pi - q4(t) \\
& - q3(t) - 2q2(t)) - 0.00006800000000 \sin(2q5(t) + 2\pi + 2q4(t)) + 0.0052200000000000 \cos(q5(t) + q3(t) \\
& + 2q2(t)) - 0.0052200000000000 \cos(q5(t) + 2\pi - q3(t) - 2q2(t)) - 0.0055100000000000 \sin(q5(t) + 2q3(t) \\
& + 2q2(t)) + 2.16840434497101 \cdot 10^{-19} \sin(-2q4(t) + q5(t) + 2\pi - 2q3(t) - 2q2(t)) + 0.00020400000000 \sin(2q5(t) \\
& + 3\pi - 2q3(t) - 2q2(t)) - 0.0052200000000000 \cos(q5(t) + \pi - q3(t)) + 0.0052200000000000 \cos(q5(t) + \pi \\
& + q3(t)) + 0.0026100000000000 \cos(q5(t) - q4(t) + q3(t) + 2q2(t)) + 0.0026100000000000 \cos(q5(t) + 2\pi \\
& + q4(t) - q3(t) - 2q2(t)) + 0.0027550000000000 \sin(q5(t) + 2\pi - 2q3(t) - 2q2(t) + q4(t)) \\
& - 0.00013600000000 \sin(q4(t) + 2q5(t) + 3\pi - 2q3(t) - 2q2(t)) + 2.16840434497101 \cdot 10^{-19} \sin(2q4(t) + q5(t) \\
& + 2q3(t) + 2q2(t)) - 0.00006800000000 \sin(-2q4(t) + 2\pi + 2q5(t)) - 0.0020300000000000 \cos\left(q5(t) + \frac{3\pi}{2} \right. \\
& \left. - q3(t) - q2(t)\right) + 0.00003400000000 \sin(2q5(t) + 3\pi - 2q3(t) - 2q2(t) - 2q4(t)) \\
& + 0.00003400000000 \sin(2q5(t) + \pi + 2q3(t) + 2q2(t) + 2q4(t)) + 0.0110200000000000 \sin(q5(t) + \pi) \\
& + 0.00013600000000 \sin(-q4(t) + 2q5(t) + \pi + 2q3(t) + 2q2(t)) + 0.0010150000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) \right. \\
& \left. + q2(t) - q4(t)\right) + 0.0010150000000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) \\
& + 0.0010150000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) + 0.0010150000000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \\
& \left. - q2(t) - q4(t)\right) + 2.16840434497101 \cdot 10^{-19} \sin(q5(t) - 2q4(t) + 2q3(t) + 2q2(t)) + 0.00013600000000 \sin(2\pi \\
& + 2q5(t)) + 0.00020400000000 \sin(2q5(t) + \pi + 2q3(t) + 2q2(t)) + 0.0026100000000000 \cos(q5(t) + \pi - q4(t) \\
& - q3(t)) + 0.0026100000000000 \cos(q5(t) + \pi + q4(t) + q3(t)) + 0.0026100000000000 \cos(q5(t) + \pi + q4(t) \\
& - q3(t)) + 0.0026100000000000 \cos(q5(t) + \pi - q4(t) + q3(t)) + 0.0027550000000000 \sin(q5(t) + 2\pi - 2q3(t) \\
& - 2q2(t) - q4(t)) + 0.0020300000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t)\right) - 0.0055100000000000 \sin(q5(t) + 2\pi \\
& - 2q3(t) - 2q2(t)) - 4.33680868994202 \cdot 10^{-19} \sin(q5(t) + \pi - 2q4(t)) - 0.0027550000000000 \sin(q5(t) + 2q3(t) \\
& + 2q2(t) - q4(t)) - 0.0027550000000000 \sin(q5(t) + 2q3(t) + 2q2(t) + q4(t)) + 0.00003400000000 \sin(2q5(t) \\
& + 3\pi + 2q4(t) - 2q3(t) - 2q2(t)) - 4.33680868994202 \cdot 10^{-19} \sin(2q4(t) + \pi + q5(t)) - 0.00013600000000 \sin(
\end{aligned}$$

$$\begin{aligned}
& -q4(t) + 2q5(t) + 3\pi - 2q3(t) - 2q2(t)) \dot{q}5(t) \dot{q}1(t) + \left( 0.002755000000000000 \sin\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) \right. \right. \\
& \left. \left. + q4(t) \right) + 0.002755000000000000 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t) \right) - 0.00003400000000 \sin\left(-2q4(t) \right. \right. \\
& \left. \left. + 2q5(t) + \frac{5\pi}{2} - q3(t) - q2(t) \right) + 0.00006799999999999993 \sin\left(2q5(t) + \frac{3\pi}{2} + q4(t) + q3(t) + q2(t) \right) \right. \\
& \left. + 0.002610000000000000 \cos\left(q5(t) + \frac{3\pi}{2} + q4(t) - q2(t) \right) + 0.00003400000000 \sin\left(2q4(t) + 2q5(t) + \frac{3\pi}{2} + q3(t) \right. \right. \\
& \left. \left. + q2(t) \right) - 0.002755000000000000 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t) \right) - 0.0026100000000000 \cos\left(q5(t) \right. \right. \\
& \left. \left. + \frac{3\pi}{2} - q4(t) - q2(t) \right) + 0.002610000000000000 \cos\left(q5(t) + \frac{\pi}{2} + q4(t) + q2(t) \right) - 0.00003400000000 \sin\left(2q5(t) \right. \right. \\
& \left. \left. + \frac{3\pi}{2} - 2q4(t) + q3(t) + q2(t) \right) - 0.309540 \cos\left(-q2(t) + \frac{\pi}{2} \right) - 0.002755000000000000 \sin\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \right. \\
& \left. \left. - q2(t) - q4(t) \right) - 2.15078605966812 \cdot 10^{-17} \sin\left(-q2(t) + \frac{\pi}{2} - 2q4(t) + q3(t) \right) + 0.00006799999999999993 \sin\left( \right. \\
& \left. -q4(t) + \frac{5\pi}{2} + 2q5(t) - q3(t) - q2(t) \right) - 0.00006799999999999993 \sin\left(2q4(t) - q3(t) - q2(t) + \frac{\pi}{2} \right) \\
& - 0.00006799999999999993 \sin\left(2q5(t) + \frac{5\pi}{2} + q4(t) - q3(t) - q2(t) \right) + 0.00006800000000000000 \sin\left(-2q4(t) \right. \\
& \left. - q3(t) - q2(t) + \frac{\pi}{2} \right) - 0.00006799999999999993 \sin\left(-q4(t) + \frac{3\pi}{2} + 2q5(t) + q3(t) + q2(t) \right) \\
& + 0.0000340000000000 \sin\left(2q5(t) + \frac{5\pi}{2} + 2q4(t) - q3(t) - q2(t) \right) - 0.002610000000000000 \cos\left(q5(t) + \frac{\pi}{2} - q4(t) \right. \\
& \left. + q2(t) \right) \dot{q}2(t)^2 + \left( \left( 0.0000680000000000 \sin\left(2q4(t) + 2q5(t) + \frac{3\pi}{2} + q3(t) + q2(t) \right) \right. \right. \\
& \left. \left. - 0.00013599999999999999 \sin\left(-q4(t) + \frac{3\pi}{2} + 2q5(t) + q3(t) + q2(t) \right) - 0.0000680000000000 \sin\left(2q5(t) + \frac{3\pi}{2} \right. \right. \right. \\
& \left. \left. - 2q4(t) + q3(t) + q2(t) \right) - 3.38813178901720 \cdot 10^{-19} \sin\left(-q2(t) + \frac{5\pi}{2} + 2q5(t) + q4(t) + q3(t) \right) \right. \\
& \left. - 3.38813178901720 \cdot 10^{-19} \sin\left(q2(t) + \frac{3\pi}{2} + 2q5(t) - q4(t) - q3(t) \right) + 0.0000680000000000 \sin\left(2q5(t) + \frac{5\pi}{2} \right. \right. \\
& \left. \left. + 2q4(t) - q3(t) - q2(t) \right) + 3.38813178901720 \cdot 10^{-19} \sin\left(q2(t) + \frac{3\pi}{2} + 2q5(t) + q4(t) - q3(t) \right) \right. \\
& \left. - 0.00013600000000 \sin\left(2q4(t) - q3(t) - q2(t) + \frac{\pi}{2} \right) - 0.0000680000000000 \sin\left(-2q4(t) + 2q5(t) + \frac{5\pi}{2} - q3(t) \right. \right. \\
& \left. \left. - q2(t) \right) + 3.38813178901720 \cdot 10^{-19} \sin\left(-q2(t) + \frac{5\pi}{2} + 2q5(t) - q4(t) + q3(t) \right) + 0.00013600000000 \sin\left(-2q4(t) \right. \right. \\
& \left. \left. - q3(t) - q2(t) + \frac{\pi}{2} \right) + 0.00013599999999999999 \sin\left(2q5(t) + \frac{3\pi}{2} + q4(t) + q3(t) + q2(t) \right) \right. \\
& \left. + 0.00013599999999999999 \sin\left(-q4(t) + \frac{5\pi}{2} + 2q5(t) - q3(t) - q2(t) \right) - 0.00013599999999999999 \sin\left(2q5(t) + \frac{5\pi}{2} \right. \right. \\
& \left. \left. + q4(t) - q3(t) - q2(t) \right) - 0.005510000000000000 \sin\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) - q4(t) \right) \right. \\
& \left. - 0.005510000000000000 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t) \right) + 0.005510000000000000 \sin\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \right. \\
& \left. \left. - q2(t) + q4(t) \right) + 0.005510000000000000 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t) \right) \right) \dot{q}3(t) + \left( \right. \\
& \left. -0.00013600000000 \sin\left(2q5(t) + \frac{3\pi}{2} + q3(t) + q2(t) \right) - 0.0000680000000000 \sin\left(-2q4(t) + 2q5(t) + \frac{5\pi}{2} - q3(t) \right. \right. \\
& \left. \left. - q2(t) \right) + 0.00013600000000 \sin\left(2q5(t) + \frac{5\pi}{2} - q3(t) - q2(t) \right) - 0.1121800000 \sin\left(-q3(t) - q2(t) + \frac{\pi}{2} \right) \right)
\end{aligned}$$

$$\begin{aligned}
& + 0.0001360000000 \sin\left(-2q_4(t) - q_3(t) - q_2(t) + \frac{\pi}{2}\right) + 0.0001360000000 \sin\left(2q_4(t) - q_3(t) - q_2(t) + \frac{\pi}{2}\right) \\
& + 0.00006800000000 \sin\left(2q_5(t) + \frac{3\pi}{2} - 2q_4(t) + q_3(t) + q_2(t)\right) - 0.00006800000000 \sin\left(2q_5(t) + \frac{5\pi}{2} + 2q_4(t) \right. \\
& \left. - q_3(t) - q_2(t)\right) + 0.00006800000000 \sin\left(2q_4(t) + 2q_5(t) + \frac{3\pi}{2} + q_3(t) + q_2(t)\right) \dot{q}_4(t) + \left( \right. \\
& -0.0001360000000 \sin\left(-q_4(t) + \frac{5\pi}{2} + 2q_5(t) - q_3(t) - q_2(t)\right) - 0.0001360000000 \sin\left(-q_4(t) + \frac{3\pi}{2} + 2q_5(t) \right. \\
& \left. + q_3(t) + q_2(t)\right) - 0.0010010000000 \sin\left(q_4(t) - q_3(t) - q_2(t) + \frac{\pi}{2}\right) + 0.00006800000000 \sin\left(-2q_4(t) + 2q_5(t) \right. \\
& \left. + \frac{5\pi}{2} - q_3(t) - q_2(t)\right) - 0.00006800000000 \sin\left(2q_5(t) + \frac{5\pi}{2} + 2q_4(t) - q_3(t) - q_2(t)\right) \\
& + 0.00006800000000 \sin\left(2q_4(t) + 2q_5(t) + \frac{3\pi}{2} + q_3(t) + q_2(t)\right) + 4.33680868994202 \cdot 10^{-19} \sin\left(2q_4(t) - q_3(t) \right. \\
& \left. - q_2(t) + \frac{3\pi}{2} + q_5(t)\right) - 4.33680868994202 \cdot 10^{-19} \sin\left(2q_4(t) + \frac{\pi}{2} + q_5(t) + q_3(t) + q_2(t)\right) \\
& + 0.0001360000000 \sin\left(2q_5(t) + \frac{5\pi}{2} + q_4(t) - q_3(t) - q_2(t)\right) + 0.0001360000000 \sin\left(2q_5(t) + \frac{3\pi}{2} + q_4(t) \right. \\
& \left. + q_3(t) + q_2(t)\right) + 4.33680868994202 \cdot 10^{-19} \sin\left(q_5(t) + \frac{\pi}{2} - 2q_4(t) + q_3(t) + q_2(t)\right) + 0.0010010000000 \sin\left( \right. \\
& \left. -q_4(t) - q_3(t) - q_2(t) + \frac{\pi}{2}\right) - 0.00006800000000 \sin\left(2q_5(t) + \frac{3\pi}{2} - 2q_4(t) + q_3(t) + q_2(t)\right) \\
& - 4.33680868994202 \cdot 10^{-19} \sin\left(q_5(t) + \frac{3\pi}{2} - 2q_4(t) - q_3(t) - q_2(t)\right) \dot{q}_5(t) + \left( 0.0007290000000 \sin\left(q_5(t) + \frac{3\pi}{2} \right. \right. \\
& \left. \left. - q_3(t) - q_2(t)\right) - 0.0007290000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t)\right) - 0.0003645000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) \right. \right. \\
& \left. \left. - q_2(t) - q_4(t)\right) - 0.0003645000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) + q_4(t)\right) - 0.0003645000000 \sin\left(q_5(t) + \frac{3\pi}{2} \right. \right. \\
& \left. \left. - q_3(t) - q_2(t) + q_4(t)\right) - 0.0003645000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) - q_4(t)\right) \dot{q}_6(t) \dot{q}_2(t) + \left( \right. \\
& -2.16840434497101 \cdot 10^{-19} \sin\left(q_3(t) + q_5(t) + \frac{3\pi}{2} - q_4(t) - q_2(t)\right) - 2.16840434497101 \cdot 10^{-19} \sin\left(-q_3(t) + q_5(t) \right. \\
& \left. + \frac{\pi}{2} + q_4(t) + q_2(t)\right) + 2.16840434497101 \cdot 10^{-19} \sin\left(-q_3(t) + q_5(t) + \frac{\pi}{2} - q_4(t) + q_2(t)\right) \\
& + 0.00003400000000 \sin\left(2q_4(t) + 2q_5(t) + \frac{3\pi}{2} + q_3(t) + q_2(t)\right) + 2.16840434497101 \cdot 10^{-19} \sin\left(q_3(t) + q_5(t) \right. \\
& \left. + \frac{3\pi}{2} + q_4(t) - q_2(t)\right) - 0.0000679999999999996 \sin\left(2q_5(t) + \frac{5\pi}{2} + q_4(t) - q_3(t) - q_2(t)\right) \\
& - 0.0000679999999999996 \sin\left(-q_4(t) + \frac{3\pi}{2} + 2q_5(t) + q_3(t) + q_2(t)\right) + 0.00003400000000 \sin\left(2q_5(t) + \frac{5\pi}{2} \right. \\
& \left. + 2q_4(t) - q_3(t) - q_2(t)\right) + 0.0000679999999999996 \sin\left(2q_5(t) + \frac{3\pi}{2} + q_4(t) + q_3(t) + q_2(t)\right) \\
& - 0.00003400000000 \sin\left(-2q_4(t) + 2q_5(t) + \frac{5\pi}{2} - q_3(t) - q_2(t)\right) - 0.00003400000000 \sin\left(2q_5(t) + \frac{3\pi}{2} \right. \\
& \left. - 2q_4(t) + q_3(t) + q_2(t)\right) + 0.00275500000000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) + q_4(t)\right) \\
& - 0.00275500000000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) - q_4(t)\right) + 0.00275500000000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) \right. \\
& \left. + q_2(t) - q_4(t)\right) + 0.0000679999999999996 \sin\left(-q_4(t) + \frac{5\pi}{2} + 2q_5(t) - q_3(t) - q_2(t)\right) \\
& - 0.000068000000000 \sin\left(2q_4(t) - q_3(t) - q_2(t) + \frac{\pi}{2}\right) + 0.000068000000000 \sin\left(-2q_4(t) - q_3(t) - q_2(t) + \frac{\pi}{2}\right)
\end{aligned}$$

$$\begin{aligned}
& -0.0027550000000000 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) \dot{q}3(t)^2 + \left( \left( -0.00013600000000 \sin\left(2q5(t) + \frac{3\pi}{2} \right. \right. \right. \\
& \left. \left. \left. + q3(t) + q2(t)\right) - 0.00006800000000 \sin\left(-2q4(t) + 2q5(t) + \frac{5\pi}{2} - q3(t) - q2(t)\right) \right. \\
& \left. + 0.00013600000000 \sin\left(2q5(t) + \frac{5\pi}{2} - q3(t) - q2(t)\right) - 0.1121800000 \sin\left(-q3(t) - q2(t) + \frac{\pi}{2}\right) \right. \\
& \left. + 0.00013600000000 \sin\left(-2q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) + 0.00013600000000 \sin\left(2q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) \right. \\
& \left. + 0.00006800000000 \sin\left(2q5(t) + \frac{3\pi}{2} - 2q4(t) + q3(t) + q2(t)\right) - 0.00006800000000 \sin\left(2q5(t) + \frac{5\pi}{2} + 2q4(t) \right. \right. \\
& \left. \left. - q3(t) - q2(t)\right) + 0.00006800000000 \sin\left(2q4(t) + 2q5(t) + \frac{3\pi}{2} + q3(t) + q2(t)\right) \dot{q}4(t) + \left( \right. \\
& \left. -0.00013600000000 \sin\left(-q4(t) + \frac{5\pi}{2} + 2q5(t) - q3(t) - q2(t)\right) - 0.00013600000000 \sin\left(-q4(t) + \frac{3\pi}{2} + 2q5(t) \right. \right. \\
& \left. \left. + q3(t) + q2(t)\right) - 0.00100100000000 \sin\left(q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) + 0.00006800000000 \sin\left(-2q4(t) + 2q5(t) \right. \right. \\
& \left. \left. + \frac{5\pi}{2} - q3(t) - q2(t)\right) - 0.00006800000000 \sin\left(2q5(t) + \frac{5\pi}{2} + 2q4(t) - q3(t) - q2(t)\right) \right. \\
& \left. + 0.00006800000000 \sin\left(2q4(t) + 2q5(t) + \frac{3\pi}{2} + q3(t) + q2(t)\right) + 4.33680868994202 \cdot 10^{-19} \sin\left(2q4(t) - q3(t) \right. \right. \\
& \left. \left. - q2(t) + \frac{3\pi}{2} + q5(t)\right) - 4.33680868994202 \cdot 10^{-19} \sin\left(2q4(t) + \frac{\pi}{2} + q5(t) + q3(t) + q2(t)\right) \right. \\
& \left. + 0.00013600000000 \sin\left(2q5(t) + \frac{5\pi}{2} + q4(t) - q3(t) - q2(t)\right) + 0.00013600000000 \sin\left(2q5(t) + \frac{3\pi}{2} + q4(t) \right. \right. \\
& \left. \left. + q3(t) + q2(t)\right) + 4.33680868994202 \cdot 10^{-19} \sin\left(q5(t) + \frac{\pi}{2} - 2q4(t) + q3(t) + q2(t)\right) + 0.00100100000000 \sin\left( \right. \right. \\
& \left. \left. -q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) - 0.00006800000000 \sin\left(2q5(t) + \frac{3\pi}{2} - 2q4(t) + q3(t) + q2(t)\right) \right. \\
& \left. - 4.33680868994202 \cdot 10^{-19} \sin\left(q5(t) + \frac{3\pi}{2} - 2q4(t) - q3(t) - q2(t)\right) \dot{q}5(t) + \left( 0.00072900000000 \sin\left(q5(t) + \frac{3\pi}{2} \right. \right. \right. \\
& \left. \left. - q3(t) - q2(t)\right) - 0.00072900000000 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t)\right) - 0.00036450000000 \sin\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \right. \\
& \left. \left. - q2(t) - q4(t)\right) - 0.00036450000000 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) - 0.00036450000000 \sin\left(q5(t) + \frac{3\pi}{2} \right. \right. \\
& \left. \left. - q3(t) - q2(t) + q4(t)\right) - 0.00036450000000 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right) \dot{q}6(t) \dot{q}3(t) \right. \\
& \left. + \left( 0.0026100000000000 \cos\left(q5(t) + \frac{\pi}{2} - q4(t) + q2(t)\right) - 0.0026100000000000 \cos\left(q5(t) + \frac{\pi}{2} + q4(t) \right. \right. \right. \\
& \left. \left. + q2(t)\right) - 0.0026100000000000 \cos\left(q5(t) + \frac{3\pi}{2} + q4(t) - q2(t)\right) - 0.0027550000000000 \sin\left(q5(t) + \frac{\pi}{2} \right. \right. \\
& \left. \left. + q3(t) + q2(t) - q4(t)\right) - 0.0027550000000000 \sin\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) \right. \\
& \left. + 0.0026100000000000 \cos\left(q5(t) + \frac{3\pi}{2} - q4(t) - q2(t)\right) + 0.0027550000000000 \sin\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \right. \\
& \left. \left. - q2(t) - q4(t)\right) + 0.00006800000000 \sin\left(-q4(t) + \frac{3\pi}{2} + 2q5(t) + q3(t) + q2(t)\right) + 0.00006800000000 \sin\left(2q5(t) \right. \right. \\
& \left. \left. + \frac{5\pi}{2} + q4(t) - q3(t) - q2(t)\right) + 0.0027550000000000 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) \right. \\
& \left. - 0.00006800000000 \sin\left(-q4(t) + \frac{5\pi}{2} + 2q5(t) - q3(t) - q2(t)\right) + 0.0010150000000000 \cos\left(q5(t) + \pi - q4(t)\right) \right. \\
& \left. - 0.0010150000000000 \cos\left(q5(t) + \pi + q4(t)\right) - 0.00006800000000 \sin\left(2q5(t) + \frac{3\pi}{2} + q4(t) + q3(t) + q2(t)\right) \right)
\end{aligned}$$

$$\begin{aligned}
& \dot{q}_4(t)^2 + \left( \left( -0.0002720000000 \sin\left(2q_5(t) + \frac{5\pi}{2} - q_3(t) - q_2(t)\right) - 0.005220000000000000 \cos\left(q_5(t) + \frac{\pi}{2} + q_4(t) \right. \right. \right. \\
& \left. \left. \left. + q_2(t)\right) - 0.002030000000000000 \cos(q_5(t) + \pi - q_4(t)) - 0.002030000000000000 \cos(q_5(t) + \pi + q_4(t)) \right. \right. \\
& \left. \left. - 0.0002720000000 \sin\left(2q_5(t) + \frac{3\pi}{2} + q_3(t) + q_2(t)\right) - 0.005220000000000000 \cos\left(q_5(t) + \frac{3\pi}{2} + q_4(t) - q_2(t)\right) \right. \right. \\
& \left. \left. + 0.005510000000000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) + q_4(t)\right) + 0.001001000000 \sin\left(q_4(t) - q_3(t) - q_2(t) \right. \right. \\
& \left. \left. + \frac{\pi}{2}\right) + 0.005510000000000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) - q_4(t)\right) - 0.005510000000000000 \sin\left(q_5(t) + \frac{3\pi}{2} \right. \right. \\
& \left. \left. - q_3(t) - q_2(t) + q_4(t)\right) - 0.005220000000000000 \cos\left(q_5(t) + \frac{\pi}{2} - q_4(t) + q_2(t)\right) - 0.00013600000000 \sin\left(2q_5(t) \right. \right. \\
& \left. \left. + \frac{3\pi}{2} + q_4(t) + q_3(t) + q_2(t)\right) + 0.00013600000000 \sin\left(-q_4(t) + \frac{5\pi}{2} + 2q_5(t) - q_3(t) - q_2(t)\right) \right. \\
& \left. - 0.00013600000000 \sin\left(-q_4(t) + \frac{3\pi}{2} + 2q_5(t) + q_3(t) + q_2(t)\right) - 0.005510000000000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) \right. \right. \\
& \left. \left. - q_2(t) - q_4(t)\right) - 0.005220000000000000 \cos\left(q_5(t) + \frac{3\pi}{2} - q_4(t) - q_2(t)\right) + 0.001001000000 \sin\left(-q_4(t) - q_3(t) \right. \right. \\
& \left. \left. - q_2(t) + \frac{\pi}{2}\right) + 0.00013600000000 \sin\left(2q_5(t) + \frac{5\pi}{2} + q_4(t) - q_3(t) - q_2(t)\right) \right) \dot{q}_5(t) \\
& + \left( 0.0003645000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) + q_4(t)\right) - 0.0003645000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) \right. \right. \\
& \left. \left. - q_2(t) - q_4(t)\right) - 0.0003645000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) + q_4(t)\right) + 0.0003645000000 \sin\left(q_5(t) + \frac{\pi}{2} \right. \right. \\
& \left. \left. + q_3(t) + q_2(t) - q_4(t)\right) \right) \dot{q}_6(t) \dot{q}_4(t) + \left( 0.002610000000000000 \cos\left(q_5(t) + \frac{3\pi}{2} - q_4(t) - q_2(t)\right) \right. \\
& \left. + 0.002610000000000000 \cos\left(q_5(t) + \frac{\pi}{2} - q_4(t) + q_2(t)\right) - 0.002610000000000000 \cos\left(q_5(t) + \frac{\pi}{2} + q_4(t) + q_2(t)\right) \right. \\
& \left. - 0.002610000000000000 \cos\left(q_5(t) + \frac{3\pi}{2} + q_4(t) - q_2(t)\right) + 0.002755000000000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) \right. \right. \\
& \left. \left. + q_4(t)\right) + 0.001015000000000000 \cos(q_5(t) + \pi - q_4(t)) - 0.001015000000000000 \cos(q_5(t) + \pi + q_4(t)) \right. \\
& \left. + 0.002755000000000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) - q_4(t)\right) - 0.002755000000000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) \right. \right. \\
& \left. \left. + q_2(t) - q_4(t)\right) - 0.002755000000000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) + q_4(t)\right) \right) \dot{q}_5(t)^2 + \left( \right. \\
& \left. -0.0007290000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t)\right) - 0.0007290000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t)\right) \right. \\
& \left. + 0.0003645000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) - q_2(t) - q_4(t)\right) - 0.0003645000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) \right. \right. \\
& \left. \left. - q_4(t)\right) - 0.0003645000000 \sin\left(q_5(t) + \frac{\pi}{2} + q_3(t) + q_2(t) + q_4(t)\right) + 0.0003645000000 \sin\left(q_5(t) + \frac{3\pi}{2} - q_3(t) \right. \right. \\
& \left. \left. - q_2(t) + q_4(t)\right) \right) \dot{q}_6(t) \dot{q}_5(t) + \left( 0.00006800000000 \cos(-q_4(t) + 2q_5(t) + 3\pi - 2q_3(t) - 2q_2(t)) \right. \\
& \left. + 2.16840434497101 \cdot 10^{-19} \cos(-2q_4(t) + q_5(t) + 2\pi - 2q_3(t) - 2q_2(t)) - 0.00001700000000 \cos(2q_5(t) + \pi \right. \\
& \left. + 2q_3(t) + 2q_2(t) + 2q_4(t)) + 2.245272000000000 \cos(-2q_2(t) + \pi) + 0.00003400000000 \cos(2q_5(t) + 2\pi \right. \\
& \left. + 2q_4(t)) + 0.002755000000000000 \cos(q_5(t) + 2q_3(t) + 2q_2(t) + q_4(t)) - 0.002755000000000000 \cos(q_5(t) + 2\pi \right. \\
& \left. - 2q_3(t) - 2q_2(t) - q_4(t)) + 0.8568000000000000 \sin(-q_3(t) - 2q_2(t) + \pi) + 0.005220000000000000 \sin(q_5(t) + \pi \right.
\end{aligned}$$

$$\begin{aligned}
& + q^3(t) - 0.0052200000000000 \sin(q^5(t) + \pi - q^3(t)) - 0.0000680000000000 \cos(-q^4(t) + 2q^5(t) + \pi + 2q^3(t) \\
& + 2q^2(t)) - 0.0000680000000000 \cos(2\pi + 2q^5(t)) - 4.33680868994202 \cdot 10^{-19} \cos(2q^4(t) + \pi + q^5(t)) \\
& + 0.002610000000000000 \sin(q^5(t) + \pi + q^4(t) + q^3(t)) + 0.002610000000000000 \sin(q^5(t) + \pi - q^4(t) - q^3(t)) \\
& + 1.9294800000000000 \cos\left(-q^2(t) + \frac{\pi}{2}\right) + 0.001015000000000000 \sin\left(q^5(t) + \frac{\pi}{2} + q^3(t) + q^2(t) - q^4(t)\right) \\
& + 0.001015000000000000 \sin\left(q^5(t) + \frac{\pi}{2} + q^3(t) + q^2(t) + q^4(t)\right) + 0.002610000000000000 \sin(q^5(t) + \pi - q^4(t) \\
& + q^3(t)) + 0.002610000000000000 \sin(q^5(t) + \pi + q^4(t) - q^3(t)) + 0.002755000000000000 \cos(q^5(t) + 2q^3(t) \\
& + 2q^2(t) - q^4(t)) - 0.002755000000000000 \cos(q^5(t) + 2\pi - 2q^3(t) - 2q^2(t) + q^4(t)) \\
& + 2.16840434497101 \cdot 10^{-19} \cos(q^5(t) - 2q^4(t) + 2q^3(t) + 2q^2(t)) - 0.0000170000000000 \cos(2q^5(t) + 3\pi + 2q^4(t) \\
& - 2q^3(t) - 2q^2(t)) - 0.8568000000000000 \sin(q^3(t)) - 4.33680868994202 \cdot 10^{-19} \cos(q^5(t) + \pi - 2q^4(t)) \\
& - 0.0000170000000000 \cos(2q^5(t) + 3\pi - 2q^3(t) - 2q^2(t) - 2q^4(t)) - 0.0001020000000000 \cos(2q^5(t) + \pi + 2q^3(t) \\
& + 2q^2(t)) - 0.011020000000000000 \cos(q^5(t) + \pi) + 0.0000340000000000 \cos(-2q^4(t) + 2\pi + 2q^5(t)) \\
& + 1.38777878078145 \cdot 10^{-17} \cos(-2q^2(t) + \pi - 2q^4(t)) + 2.16840434497101 \cdot 10^{-19} \cos(2q^4(t) + q^5(t) + 2q^3(t) \\
& + 2q^2(t)) + 0.002610000000000000 \sin(q^5(t) + 2\pi - q^4(t) - q^3(t) - 2q^2(t)) + 0.002610000000000000 \sin(q^5(t) \\
& + q^4(t) + q^3(t) + 2q^2(t)) + 0.002610000000000000 \sin(q^5(t) + 2\pi + q^4(t) - q^3(t) - 2q^2(t)) \\
& + 0.002610000000000000 \sin(q^5(t) - q^4(t) + q^3(t) + 2q^2(t)) + 0.0000340000000000062 \cos(-2q^4(t) - 2q^3(t) \\
& - 2q^2(t) + \pi) + 0.0000680000000000 \cos(q^4(t) + 2q^5(t) + 3\pi - 2q^3(t) - 2q^2(t)) - 0.0000170000000000 \cos(2q^5(t) \\
& + \pi - 2q^4(t) + 2q^3(t) + 2q^2(t)) + 0.3332000000000000 \sin\left(-q^3(t) - q^2(t) + \frac{\pi}{2}\right) \\
& + 2.16840434497101 \cdot 10^{-19} \cos(2q^4(t) + q^5(t) + 2\pi - 2q^3(t) - 2q^2(t)) + 3.7466280000000000 \\
& - 0.005220000000000000 \sin(q^5(t) + 2\pi - q^3(t) - 2q^2(t)) + 0.005220000000000000 \sin(q^5(t) + q^3(t) + 2q^2(t)) \\
& + 0.005510000000000000 \cos(q^5(t) + 2q^3(t) + 2q^2(t)) + 0.005510000000000000 \cos(q^5(t) + 2\pi - 2q^3(t) - 2q^2(t)) \\
& - 0.0000679999999999709 \cos(2q^4(t)) - 0.4996555000000000 \cos(-2q^3(t) - 2q^2(t) + \pi) \\
& + 0.0000339999999999785 \cos(2q^4(t) - 2q^3(t) - 2q^2(t) + \pi) + 0.002030000000000000 \sin\left(q^5(t) + \frac{\pi}{2} + q^3(t) \\
& + q^2(t)\right) - 0.002030000000000000 \sin\left(q^5(t) + \frac{3\pi}{2} - q^3(t) - q^2(t)\right) + 0.001015000000000000 \sin\left(q^5(t) + \frac{3\pi}{2} \\
& - q^3(t) - q^2(t) + q^4(t)\right) + 0.001015000000000000 \sin\left(q^5(t) + \frac{3\pi}{2} - q^3(t) - q^2(t) - q^4(t)\right) \\
& - 0.0000680000000000 \cos(2q^5(t) + \pi + q^4(t) + 2q^3(t) + 2q^2(t)) - 0.0001020000000000 \cos(2q^5(t) + 3\pi - 2q^3(t) \\
& - 2q^2(t)) \ddot{q}l(t) + \left(4.33680868994202 \cdot 10^{-19} \cos\left(2q^4(t) + \frac{\pi}{2} + q^5(t) + q^3(t) + q^2(t)\right)\right)
\end{aligned}$$

$$\begin{aligned}
& -4.33680868994202 \cdot 10^{-19} \cos\left(2q4(t) - q3(t) - q2(t) + \frac{3\pi}{2} + q5(t)\right) + 0.002610000000 \sin\left(q5(t) + \frac{\pi}{2} + q4(t) \right. \\
& \left. + q2(t)\right) - 0.00006800000000 \cos\left(2q5(t) + \frac{5\pi}{2} + q4(t) - q3(t) - q2(t)\right) + 0.002755000000 \cos\left(q5(t) + \frac{\pi}{2} \right. \\
& \left. + q3(t) + q2(t) + q4(t)\right) - 0.002610000000 \sin\left(q5(t) + \frac{3\pi}{2} + q4(t) - q2(t)\right) - 0.00006800000000 \cos\left(2q5(t) \right. \\
& \left. + \frac{3\pi}{2} + q4(t) + q3(t) + q2(t)\right) - 0.00003400000000 \cos\left(2q4(t) + 2q5(t) + \frac{3\pi}{2} + q3(t) + q2(t)\right) \\
& + 0.002755000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) + 0.002610000000 \sin\left(q5(t) + \frac{3\pi}{2} - q4(t) - q2(t)\right) \\
& - 0.002610000000 \sin\left(q5(t) + \frac{\pi}{2} - q4(t) + q2(t)\right) + 0.309540 \sin\left(-q2(t) + \frac{\pi}{2}\right) - 0.00003400000000 \cos\left(-2q4(t) \right. \\
& \left. + 2q5(t) + \frac{5\pi}{2} - q3(t) - q2(t)\right) - 0.002755000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) - q4(t)\right) \\
& + 0.00003400000000 \cos\left(2q5(t) + \frac{3\pi}{2} - 2q4(t) + q3(t) + q2(t)\right) + 0.00006800000000 \cos\left(-q4(t) + \frac{5\pi}{2} + 2q5(t) \right. \\
& \left. - q3(t) - q2(t)\right) - 4.33680868994202 \cdot 10^{-19} \cos\left(q5(t) + \frac{\pi}{2} - 2q4(t) + q3(t) + q2(t)\right) \\
& + 4.33680868994202 \cdot 10^{-19} \cos\left(q5(t) + \frac{3\pi}{2} - 2q4(t) - q3(t) - q2(t)\right) + 0.00006800000000 \cos\left(-q4(t) + \frac{3\pi}{2} \right. \\
& \left. + 2q5(t) + q3(t) + q2(t)\right) - 0.000068000000000403 \cos\left(2q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) \\
& - 2.77555756156289 \cdot 10^{-17} \cos\left(-q2(t) + \frac{\pi}{2} - 2q4(t) + q3(t)\right) + 0.00003400000000 \cos\left(2q5(t) + \frac{5\pi}{2} + 2q4(t) \right. \\
& \left. - q3(t) - q2(t)\right) + 0.000068000000000125 \cos\left(-2q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) - 0.002755000000 \cos\left(q5(t) + \frac{\pi}{2} \right. \\
& \left. + q3(t) + q2(t) - q4(t)\right) \Big) \ddot{q}_2(t) + \left(-0.002755000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right)\right) \\
& - 2.77555756156289 \cdot 10^{-17} \cos\left(-q2(t) + \frac{\pi}{2} - 2q4(t) + q3(t)\right) - 4.33680868994202 \cdot 10^{-19} \cos\left(q5(t) + \frac{\pi}{2} - 2q4(t) \right. \\
& \left. + q3(t) + q2(t)\right) + 2.77555756156289 \cdot 10^{-17} \cos\left(-q2(t) + \frac{\pi}{2} + 2q4(t) + q3(t)\right) + 0.002755000000 \cos\left(q5(t) \right. \\
& \left. + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) + 0.00006800000000 \cos\left(-q4(t) + \frac{3\pi}{2} + 2q5(t) + q3(t) + q2(t)\right) \\
& - 0.002755000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) - q4(t)\right) + 0.000067999999999847 \cos\left(-2q4(t) - q3(t) \right. \\
& \left. - q2(t) + \frac{\pi}{2}\right) - 0.000067999999999847 \cos\left(2q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) + 4.33680868994202 \cdot 10^{-19} \cos\left(2q4(t) \right. \\
& \left. + \frac{\pi}{2} + q5(t) + q3(t) + q2(t)\right) + 0.00006800000000 \cos\left(-q4(t) + \frac{5\pi}{2} + 2q5(t) - q3(t) - q2(t)\right) \\
& + 0.002755000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) + 0.00003400000000 \cos\left(2q5(t) + \frac{3\pi}{2} - 2q4(t) \right. \\
& \left. + q3(t) + q2(t)\right) - 0.00003400000000 \cos\left(-2q4(t) + 2q5(t) + \frac{5\pi}{2} - q3(t) - q2(t)\right) \\
& - 4.33680868994202 \cdot 10^{-19} \cos\left(2q4(t) - q3(t) - q2(t) + \frac{3\pi}{2} + q5(t)\right) + 4.33680868994202 \cdot 10^{-19} \cos\left(q5(t) + \frac{3\pi}{2} \right. \\
& \left. - 2q4(t) - q3(t) - q2(t)\right) - 0.00003400000000 \cos\left(2q4(t) + 2q5(t) + \frac{3\pi}{2} + q3(t) + q2(t)\right) \\
& + 0.00003400000000 \cos\left(2q5(t) + \frac{5\pi}{2} + 2q4(t) - q3(t) - q2(t)\right) - 0.00006800000000 \cos\left(2q5(t) + \frac{3\pi}{2} + q4(t) \right. \\
& \left. + q3(t) + q2(t)\right) - 0.00006800000000 \cos\left(2q5(t) + \frac{5\pi}{2} + q4(t) - q3(t) - q2(t)\right) \Big) \ddot{q}_3(t) + \left(
\end{aligned}$$

$$\begin{aligned}
& -0.0027550000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right) - 0.0026100000000000 \sin\left(q5(t) + \frac{\pi}{2} - q4(t) \right. \\
& \left. + q2(t)\right) - 0.0010150000000000 \sin(q5(t) + \pi - q4(t)) - 0.0026100000000000 \sin\left(q5(t) + \frac{3\pi}{2} + q4(t) - q2(t)\right) \\
& + 0.0027550000000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) - q4(t)\right) + 0.00013600000000 \cos\left(2q5(t) + \frac{3\pi}{2} + q3(t) \right. \\
& \left. + q2(t)\right) + 0.00013600000000 \cos\left(2q5(t) + \frac{5\pi}{2} - q3(t) - q2(t)\right) - 0.0026100000000000 \sin\left(q5(t) + \frac{\pi}{2} + q4(t) \right. \\
& \left. + q2(t)\right) - 0.0026100000000000 \sin\left(q5(t) + \frac{3\pi}{2} - q4(t) - q2(t)\right) - 0.1121800000 \cos\left(-q3(t) - q2(t) + \frac{\pi}{2}\right) \\
& + 0.0000680000000000 \cos\left(2q5(t) + \frac{3\pi}{2} + q4(t) + q3(t) + q2(t)\right) - 0.0000680000000000 \cos\left(2q5(t) + \frac{5\pi}{2} + q4(t) \right. \\
& \left. - q3(t) - q2(t)\right) + 0.0000680000000000 \cos\left(-q4(t) + \frac{3\pi}{2} + 2q5(t) + q3(t) + q2(t)\right) - 0.0000680000000000 \cos\left( \right. \\
& \left. -q4(t) + \frac{5\pi}{2} + 2q5(t) - q3(t) - q2(t)\right) - 0.0010150000000000 \sin(q5(t) + \pi + q4(t)) \\
& - 0.0027550000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) + 0.0027550000000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \\
& \left. - q2(t) + q4(t)\right) \ddot{q4}(t) + \left(0.0026100000000000 \sin\left(q5(t) + \frac{\pi}{2} - q4(t) + q2(t)\right) \right. \\
& \left. - 0.0026100000000000 \sin\left(q5(t) + \frac{\pi}{2} + q4(t) + q2(t)\right) - 0.001001000000 \cos\left(q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) \right. \\
& \left. + 0.001001000000 \cos\left(-q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) - 0.0026100000000000 \sin\left(q5(t) + \frac{3\pi}{2} + q4(t) - q2(t)\right) \right. \\
& \left. - 0.0027550000000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) - q4(t)\right) + 0.0026100000000000 \sin\left(q5(t) + \frac{3\pi}{2} \right. \right. \\
& \left. \left. - q4(t) - q2(t)\right) + 0.0010150000000000 \sin(q5(t) + \pi - q4(t)) + 0.0027550000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) \right. \right. \\
& \left. \left. + q2(t) - q4(t)\right) - 0.0027550000000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) \right. \\
& \left. + 0.0027550000000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) - 0.0010150000000000 \sin(q5(t) + \pi + q4(t)) \right) \\
& \ddot{q5}(t) + \left(0.00036450000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right) - 0.00036450000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \right. \\
& \left. \left. - q2(t) + q4(t)\right) + 0.00072900000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t)\right) + 0.00072900000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \right. \\
& \left. \left. - q2(t)\right) + 0.00036450000000 \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) - 0.00036450000000 \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) \right. \right. \\
& \left. \left. - q2(t) - q4(t)\right) \right) \ddot{q6}(t)
\end{aligned}$$

>

> #Setting up the matrix elements

> m11 := eval(τlx, {DDq[1] = 1, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :

> m12 := eval(τlx, {DDq[1] = 0, DDq[2] = 1, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :

> m13 := eval(τlx, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 1, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :

> m14 := eval(τlx, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 1, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :

> m15 := eval(τlx, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 1, DDq[6] = 0, Dq[1] = 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :



```

> m54 := eval(τ5y, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 1, DDq[5] = 0, DDq[6] = 0, Dq[1]
= 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :
> m55 := eval(τ5y, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 1, DDq[6] = 0, Dq[1]
= 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :
> m56 := eval(τ5y, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 1, Dq[1]
= 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :
>
> m61 := eval(τ6z, {DDq[1] = 1, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1]
= 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :
> m62 := eval(τ6z, {DDq[1] = 0, DDq[2] = 1, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1]
= 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :
> m63 := eval(τ6z, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 1, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1]
= 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :
> m64 := eval(τ6z, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 1, DDq[5] = 0, DDq[6] = 0, Dq[1]
= 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :
> m65 := eval(τ6z, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 1, DDq[6] = 0, Dq[1]
= 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :
> m66 := eval(τ6z, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 1, Dq[1]
= 0, Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0, g = 0}) :
>
> g6 := eval(τ6z, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0,
Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0}) :
> g5 := eval(τ5y, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0,
Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0}) :
> g4 := eval(τ4x, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0,
Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0}) :
> g3 := eval(τ3y, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0,
Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0}) :
> g2 := eval(τ2z, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0,
Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0}) :
> g1 := eval(τ1x, {DDq[1] = 0, DDq[2] = 0, DDq[3] = 0, DDq[4] = 0, DDq[5] = 0, DDq[6] = 0, Dq[1] = 0,
Dq[2] = 0, Dq[3] = 0, Dq[4] = 0, Dq[5] = 0, Dq[6] = 0}) :

```

### #Setting up the dynamic system

```

> M0 := Matrix([ [m11, m12, m13, m14, m15, m16], [m21, m22, m23, m24, m25, m26], [m31, m32, m33, m34,
m35, m36], [m41, m42, m43, m44, m45, m46], [m51, m52, m53, m54, m55, m56], [m61, m62, m63, m64,
m65, m66]]) :
> G0 := Vector([ [g1], [g2], [g3], [g4], [g5], [g6]]) :
> τ0 := Vector([ [τ1x], [τ2z], [τ3y], [τ4x], [τ5y], [τ6z]]) :
> C0Dq := combine(τ0 - MatrixVectorMultiply(M0, DDq) - G0, trig) :
>

```

### #Kinetic energy

```

> Ekin := combine(1/2 * VectorMatrixMultiply(Transpose(Dq), MatrixVectorMultiply(M0, Dq)), trig)

```

$$\begin{aligned}
E_{kin} := & 0.964740000000000 \cos\left(-2q_2(t) + \frac{\pi}{2}\right) \dot{q}_1(t)^2 + 2.865169500000000 \dot{q}_2(t)^2 - 8.500000000 10^{-6} \dot{q}_1(t)^2 \cos(2q_5(t) + \pi) \\
& + 2q_3(t) + 2q_2(t) + 2q_4(t) - 0.000034000000000 \dot{q}_3(t)^2 \cos(2q_5(t) + 2\pi + 2q_4(t)) \\
& + 0.000068000000000 \dot{q}_3(t)^2 \cos(2\pi + 2q_5(t)) - 0.000051000000000 \dot{q}_1(t)^2 \cos(2q_5(t) + 3\pi - 2q_3(t) - 2q_2(t)) \\
& + 1.122636000000000 \dot{q}_1(t)^2 \cos(-2q_2(t) + \pi) - 0.000034000000000 \dot{q}_1(t) \dot{q}_2(t) \cos\left(2q_4(t) + 2q_5(t) + \frac{3\pi}{2} + q_3(t)\right)
\end{aligned} \quad (2)$$

$$\begin{aligned}
& + q2(t) \Big) - 0.00003400000000 \dot{q}1(t) \dot{q}3(t) \cos\left(2q4(t) + 2q5(t) + \frac{3\pi}{2} + q3(t) + q2(t)\right) + 0.00006800000000 \dot{q}4(t) \\
& \dot{q}1(t) \cos\left(2q5(t) + \frac{3\pi}{2} + q4(t) + q3(t) + q2(t)\right) - 0.00006800000000 \dot{q}2(t) \dot{q}3(t) \cos(2q5(t) + 2\pi + 2q4(t)) \\
& + 0.00013600000000 \dot{q}2(t) \dot{q}3(t) \cos(2\pi + 2q5(t)) - 0.00006800000000 \dot{q}1(t) \dot{q}2(t) \cos\left(2q5(t) + \frac{5\pi}{2} + q4(t) - q3(t) \right. \\
& \left. - q2(t)\right) - 2.16840434497101 \cdot 10^{-19} \dot{q}1(t)^2 \cos(q5(t) + \pi - 2q4(t)) + 6.93889390390723 \cdot 10^{-18} \dot{q}1(t)^2 \cos(-2q2(t) \\
& + \pi - 2q4(t)) - 6.93889390390723 \cdot 10^{-18} \dot{q}1(t)^2 \cos(2q3(t) + 2q4(t)) + 1.08420217248550 \cdot 10^{-19} \dot{q}1(t)^2 \cos(2q4(t) \\
& + q5(t) + 2q3(t) + 2q2(t)) - 1.38777878078145 \cdot 10^{-17} \dot{q}1(t)^2 \cos(2q3(t) - 2q4(t)) \\
& + 1.08420217248550 \cdot 10^{-19} \dot{q}1(t)^2 \cos(2q4(t) + q5(t) + 2\pi - 2q3(t) - 2q2(t)) \\
& + 4.33680868994202 \cdot 10^{-19} \dot{q}3(t)^2 \cos(2q4(t) + \pi + q5(t)) + 4.33680868994202 \cdot 10^{-19} \dot{q}3(t)^2 \cos(q5(t) + \pi - 2q4(t)) \\
& + 1.08420217248550 \cdot 10^{-19} \dot{q}1(t)^2 \cos(-2q4(t) + q5(t) + 2\pi - 2q3(t) - 2q2(t)) \\
& + 4.33680868994202 \cdot 10^{-19} \dot{q}2(t)^2 \cos(q5(t) + \pi - 2q4(t)) + 4.33680868994202 \cdot 10^{-19} \dot{q}2(t)^2 \cos(2q4(t) + \pi + q5(t)) \\
& + 1.38777878078145 \cdot 10^{-17} \dot{q}1(t)^2 \cos(-2q2(t) + \pi + 2q4(t)) - 2.16840434497101 \cdot 10^{-19} \dot{q}1(t)^2 \cos(2q4(t) + \pi \\
& + q5(t)) + 1.08420217248550 \cdot 10^{-19} \dot{q}1(t)^2 \cos(q5(t) - 2q4(t) + 2q3(t) + 2q2(t)) + 0.5557455000000000 \dot{q}3(t)^2 \\
& + 0.00003400000000 \dot{q}1(t) \dot{q}2(t) \cos\left(2q5(t) + \frac{3\pi}{2} - 2q4(t) + q3(t) + q2(t)\right) + 0.00003400000000 \dot{q}1(t) \\
& \dot{q}3(t) \cos\left(2q5(t) + \frac{3\pi}{2} - 2q4(t) + q3(t) + q2(t)\right) + 0.00006800000000 \dot{q}4(t) \dot{q}1(t) \cos\left(-q4(t) + \frac{3\pi}{2} + 2q5(t) \right. \\
& \left. + q3(t) + q2(t)\right) - 0.00003400000000 \dot{q}1(t) \dot{q}2(t) \cos\left(-2q4(t) + 2q5(t) + \frac{5\pi}{2} - q3(t) - q2(t)\right) \\
& - 0.00003400000000 \dot{q}1(t) \dot{q}3(t) \cos\left(-2q4(t) + 2q5(t) + \frac{5\pi}{2} - q3(t) - q2(t)\right) - 0.00006800000000 \dot{q}4(t) \dot{q}1(t) \cos\left( \right. \\
& \left. -q4(t) + \frac{5\pi}{2} + 2q5(t) - q3(t) - q2(t)\right) - 0.0110200000000000 \cos(q5(t) + \pi) \dot{q}2(t)^2 \\
& - 0.0110200000000000 \cos(q5(t) + \pi) \dot{q}3(t)^2 + 0.05609000000000 \dot{q}4(t)^2 - 0.4284000000000000 \sin(q3(t)) \dot{q}1(t)^2 \\
& - 0.8568000000000000 \sin(q3(t)) \dot{q}2(t)^2 + 0.1666000000000000 \dot{q}1(t)^2 \sin\left(-q3(t) - q2(t) + \frac{\pi}{2}\right) \\
& + 0.4284000000000000 \dot{q}1(t)^2 \sin(-q3(t) - 2q2(t) + \pi) - 0.0000680000000000 \dot{q}1(t) \dot{q}3(t) \cos\left(2q5(t) + \frac{5\pi}{2} + q4(t) \right. \\
& \left. - q3(t) - q2(t)\right) - 0.0000680000000000 \dot{q}1(t) \dot{q}3(t) \cos\left(2q5(t) + \frac{3\pi}{2} + q4(t) + q3(t) + q2(t)\right) + 0.00013600000000 \\
& \dot{q}4(t) \dot{q}3(t) \cos(2q5(t) + 2\pi + q4(t)) + 0.0000680000000000 \dot{q}1(t) \dot{q}3(t) \cos\left(-q4(t) + \frac{5\pi}{2} + 2q5(t) - q3(t) - q2(t)\right) \\
& + 0.001015000000000000 \dot{q}1(t) \dot{q}5(t) \sin(q5(t) + \pi - q4(t)) - 0.001015000000000000 \dot{q}1(t) \dot{q}5(t) \sin(q5(t) + \pi + q4(t)) \\
& + 0.00036450000000 \dot{q}1(t) \dot{q}6(t) \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right) - 0.00036450000000 \dot{q}1(t) \dot{q}6(t) \cos\left(q5(t) \right. \\
& \left. + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) + 0.00072900000000 \dot{q}1(t) \dot{q}6(t) \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t)\right) \\
& + 0.00072900000000 \dot{q}1(t) \dot{q}6(t) \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t)\right) + 0.00072900000000 \dot{q}6(t)^2 \\
& - 0.022040000000000000 \cos(q5(t) + \pi) \dot{q}2(t) \dot{q}3(t) + 0.0000680000000000125 \dot{q}1(t) \dot{q}3(t) \cos\left(-2q4(t) - q3(t) - q2(t) \right. \\
& \left. + \frac{\pi}{2}\right) - 0.0000679999999999986 \dot{q}1(t) \dot{q}2(t) \cos\left(2q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) - 0.0000680000000000125 \dot{q}1(t) \\
& \dot{q}3(t) \cos\left(2q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) + 0.000135999999999969 \dot{q}2(t) \dot{q}3(t) \cos(2q4(t)) \\
& - 0.0000340000000000 \dot{q}1(t)^2 \cos(-q4(t) + 2q5(t) + \pi + 2q3(t) + 2q2(t)) - 8.500000000 \cdot 10^{-6} \dot{q}1(t)^2 \cos(2q5(t) + 3\pi \\
& - 2q3(t) - 2q2(t) - 2q4(t)) + 0.0000680000000000 \dot{q}1(t) \dot{q}2(t) \cos\left(-q4(t) + \frac{5\pi}{2} + 2q5(t) - q3(t) - q2(t)\right)
\end{aligned}$$

$$\begin{aligned}
& + 0.00006800000000 \dot{q}1(t) \dot{q}2(t) \cos\left(-q4(t) + \frac{3\pi}{2} + 2q5(t) + q3(t) + q2(t)\right) - 0.00013600000000 \dot{q}4(t) \\
& \dot{q}2(t) \cos(2q5(t) + 2\pi - q4(t)) + 0.001001000000 \dot{q}5(t) \dot{q}1(t) \cos\left(-q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) - 0.001001000000 \\
& \dot{q}5(t) \dot{q}1(t) \cos\left(q4(t) - q3(t) - q2(t) + \frac{\pi}{2}\right) + 0.002002000000 \dot{q}5(t) \dot{q}2(t) \cos(q4(t)) + 0.002002000000 \dot{q}5(t) \\
& \dot{q}3(t) \cos(q4(t)) + 0.00003400000000 \dot{q}1(t) \dot{q}2(t) \cos\left(2q5(t) + \frac{5\pi}{2} + 2q4(t) - q3(t) - q2(t)\right) + 0.00003400000000 \\
& \dot{q}1(t) \dot{q}3(t) \cos\left(2q5(t) + \frac{5\pi}{2} + 2q4(t) - q3(t) - q2(t)\right) - 0.00006800000000 \dot{q}4(t) \dot{q}1(t) \cos\left(2q5(t) + \frac{5\pi}{2} + q4(t) \right. \\
& \left. - q3(t) - q2(t)\right) + 0.0000169999999999962 \dot{q}1(t)^2 \cos(-2q4(t) - 2q3(t) - 2q2(t) + \pi) + 1.87331400000000 \dot{q}1(t)^2 \\
& + 0.001305000000000000 \dot{q}1(t)^2 \sin(q5(t) + 2\pi + q4(t) - q3(t) - 2q2(t)) + 0.001305000000000000 \dot{q}1(t)^2 \sin(q5(t) \\
& - q4(t) + q3(t) + 2q2(t)) + 0.000507500000000000 \dot{q}1(t)^2 \sin\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) - q4(t)\right) \\
& + 0.000507500000000000 \dot{q}1(t)^2 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) + 0.000507500000000000 \dot{q}1(t)^2 \sin\left(q5(t) \right. \\
& \left. + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) + 0.000507500000000000 \dot{q}1(t)^2 \sin\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right) \\
& + 0.002610000000000000 \dot{q}2(t)^2 \sin(q5(t) + \pi - q4(t) + q3(t)) + 0.002610000000000000 \dot{q}2(t)^2 \sin(q5(t) + \pi + q4(t) \\
& - q3(t)) + 0.001377500000000000 \dot{q}1(t)^2 \cos(q5(t) + 2q3(t) + 2q2(t) - q4(t)) \\
& - 0.001377500000000000 \dot{q}1(t)^2 \cos(q5(t) + 2\pi - 2q3(t) - 2q2(t) + q4(t)) + 0.001305000000000000 \dot{q}1(t)^2 \sin(q5(t) \\
& + \pi + q4(t) + q3(t)) + 0.001305000000000000 \dot{q}1(t)^2 \sin(q5(t) + \pi - q4(t) - q3(t)) + 0.00036450000000 \dot{q}1(t) \\
& \dot{q}6(t) \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) + q4(t)\right) - 0.00036450000000 \dot{q}1(t) \dot{q}6(t) \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) \right. \\
& \left. - q4(t)\right) + 8.67361737988404 \cdot 10^{-19} \dot{q}2(t) \dot{q}3(t) \cos(2q4(t) + \pi + q5(t)) - 0.8568000000000000 \dot{q}2(t) \dot{q}3(t) \sin(q3(t)) \\
& + 8.67361737988404 \cdot 10^{-19} \dot{q}2(t) \dot{q}3(t) \cos(q5(t) + \pi - 2q4(t)) + 0.002610000000000000 \dot{q}2(t) \dot{q}3(t) \sin(q5(t) + \pi \\
& - q4(t) + q3(t)) + 0.002610000000000000 \dot{q}2(t) \dot{q}3(t) \sin(q5(t) + \pi + q4(t) + q3(t)) + 0.002610000000000000 \dot{q}2(t) \\
& \dot{q}3(t) \sin(q5(t) + \pi + q4(t) - q3(t)) + 0.002610000000000000 \dot{q}2(t) \dot{q}3(t) \sin(q5(t) + \pi - q4(t) - q3(t)) \\
& + 0.005220000000000000 \dot{q}2(t) \dot{q}3(t) \sin(q5(t) + \pi + q3(t)) - 0.005220000000000000 \dot{q}2(t) \dot{q}3(t) \sin(q5(t) + \pi - q3(t)) \\
& + 0.002610000000000000 \dot{q}2(t) \dot{q}4(t) \sin(q5(t) + \pi + q4(t) + q3(t)) - 0.002610000000000000 \dot{q}2(t) \dot{q}4(t) \sin(q5(t) + \pi \\
& + q4(t) - q3(t)) - 0.002610000000000000 \dot{q}2(t) \dot{q}4(t) \sin(q5(t) + \pi - q4(t) + q3(t)) + 0.002610000000000000 \dot{q}2(t) \\
& \dot{q}4(t) \sin(q5(t) + \pi - q4(t) - q3(t)) - 0.002610000000000000 \dot{q}2(t) \dot{q}5(t) \sin(q5(t) + \pi + q4(t) - q3(t)) \\
& + 0.002610000000000000 \dot{q}2(t) \dot{q}5(t) \sin(q5(t) + \pi - q4(t) + q3(t)) + 0.002610000000000000 \dot{q}2(t) \dot{q}5(t) \sin(q5(t) + \pi \\
& + q4(t) + q3(t)) + 0.005220000000000000 \dot{q}2(t) \dot{q}5(t) \sin(q5(t) + \pi + q3(t)) - 0.002610000000000000 \dot{q}2(t) \\
& \dot{q}5(t) \sin(q5(t) + \pi - q4(t) - q3(t)) + 0.005220000000000000 \dot{q}2(t) \dot{q}5(t) \sin(q5(t) + \pi - q3(t)) - 0.00072900000000 \\
& \dot{q}2(t) \dot{q}6(t) \cos(q5(t) + \pi - q4(t)) + 0.00072900000000 \dot{q}2(t) \dot{q}6(t) \cos(q5(t) + \pi + q4(t)) - 0.00072900000000 \dot{q}3(t) \\
& \dot{q}6(t) \cos(q5(t) + \pi - q4(t)) + 0.00072900000000 \dot{q}3(t) \dot{q}6(t) \cos(q5(t) + \pi + q4(t)) - 0.001458000000 \dot{q}4(t) \cos(q5(t) \\
& + \pi) \dot{q}6(t) + 4.33680868994202 \cdot 10^{-19} \dot{q}1(t) \dot{q}3(t) \cos\left(q5(t) + \frac{3\pi}{2} - 2q4(t) - q3(t) - q2(t)\right) \\
& - 0.002610000000000000 \dot{q}1(t) \dot{q}4(t) \sin\left(q5(t) + \frac{\pi}{2} - q4(t) + q2(t)\right) - 0.001015000000000000 \dot{q}1(t) \dot{q}4(t) \sin(q5(t) \\
& + \pi - q4(t)) + 4.33680868994202 \cdot 10^{-19} \dot{q}1(t) \dot{q}3(t) \cos\left(2q4(t) + \frac{\pi}{2} + q5(t) + q3(t) + q2(t)\right) \\
& - 1.38777878078145 \cdot 10^{-17} \dot{q}1(t) \dot{q}2(t) \cos\left(-q2(t) + \frac{\pi}{2} - 2q4(t) + q3(t)\right) + 4.33680868994202 \cdot 10^{-19} \dot{q}1(t) \\
& \dot{q}2(t) \cos\left(2q4(t) + \frac{\pi}{2} + q5(t) + q3(t) + q2(t)\right) + 1.38777878078145 \cdot 10^{-17} \dot{q}1(t) \dot{q}2(t) \cos\left(-q2(t) + \frac{\pi}{2} + 2q4(t) \right. \\
& \left. + q3(t)\right) - 0.002610000000000000 \dot{q}1(t) \dot{q}4(t) \sin\left(q5(t) + \frac{3\pi}{2} + q4(t) - q2(t)\right) - 0.002610000000000000 \dot{q}1(t)
\end{aligned}$$



$$\begin{aligned}
& \dot{q}1(t) \dot{q}2(t) \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) + 0.002755000000000000 \dot{q}1(t) \dot{q}3(t) \cos\left(q5(t) + \frac{\pi}{2} + q3(t) \right. \\
& \left. + q2(t) + q4(t)\right) - 0.002755000000000000 \dot{q}1(t) \dot{q}3(t) \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) - q4(t)\right) \\
& - 0.002755000000000000 \dot{q}1(t) \dot{q}3(t) \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right) + 0.002755000000000000 \dot{q}1(t) \\
& \dot{q}3(t) \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) - 0.002755000000000000 \dot{q}5(t) \dot{q}1(t) \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) \right. \\
& \left. + q4(t)\right) - 0.002755000000000000 \dot{q}5(t) \dot{q}1(t) \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) - q4(t)\right) - 0.005510000000000000 \\
& \dot{q}5(t) \dot{q}2(t) \cos(q5(t) + \pi - q4(t)) - 0.005510000000000000 \dot{q}5(t) \dot{q}2(t) \cos(q5(t) + \pi + q4(t)) \\
& - 0.005510000000000000 \dot{q}5(t) \dot{q}3(t) \cos(q5(t) + \pi - q4(t)) - 0.005510000000000000 \dot{q}5(t) \dot{q}3(t) \cos(q5(t) + \pi \\
& + q4(t)) + 0.0000340000000000 \dot{q}1(t)^2 \cos(q4(t) + 2q5(t) + 3\pi - 2q3(t) - 2q2(t)) \\
& - 8.5000000000 \cdot 10^{-6} \dot{q}1(t)^2 \cos(2q5(t) + \pi - 2q4(t) + 2q3(t) + 2q2(t)) + 0.00001700000000 \dot{q}1(t)^2 \cos(-2q4(t) \\
& + 2\pi + 2q5(t)) + 0.00006800000000 \dot{q}1(t) \dot{q}3(t) \cos\left(-q4(t) + \frac{3\pi}{2} + 2q5(t) + q3(t) + q2(t)\right) - 0.00013600000000 \\
& \dot{q}4(t) \dot{q}3(t) \cos(2q5(t) + 2\pi - q4(t)) - 0.002755000000000000 \dot{q}4(t) \dot{q}1(t) \cos\left(q5(t) + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right) \\
& + 0.002755000000000000 \dot{q}4(t) \dot{q}1(t) \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) + 0.005510000000000000 \dot{q}4(t) \\
& \dot{q}2(t) \cos(q5(t) + \pi - q4(t)) - 0.005510000000000000 \dot{q}4(t) \dot{q}2(t) \cos(q5(t) + \pi + q4(t)) + 0.005510000000000000 \\
& \dot{q}4(t) \dot{q}3(t) \cos(q5(t) + \pi - q4(t)) - 0.005510000000000000 \dot{q}4(t) \dot{q}3(t) \cos(q5(t) + \pi + q4(t)) \\
& - 0.005510000000000000 \cos(q5(t) + \pi) \dot{q}1(t)^2 - 0.0000340000000000 \dot{q}1(t)^2 \cos(2q5(t) + \pi + q4(t) + 2q3(t) \\
& + 2q2(t)) - 0.0000340000000000 \dot{q}1(t)^2 \cos(2\pi + 2q5(t)) + 0.0000340000000000 \dot{q}1(t)^2 \cos(-q4(t) + 2q5(t) + 3\pi \\
& - 2q3(t) - 2q2(t)) + 0.002610000000000000 \dot{q}1(t) \dot{q}2(t) \sin\left(q5(t) + \frac{3\pi}{2} - q4(t) - q2(t)\right) + 0.002610000000000000 \\
& \dot{q}1(t) \dot{q}2(t) \sin\left(q5(t) + \frac{\pi}{2} + q4(t) + q2(t)\right) - 0.002610000000000000 \dot{q}1(t) \dot{q}2(t) \sin\left(q5(t) + \frac{3\pi}{2} + q4(t) - q2(t)\right) \\
& - 0.002610000000000000 \dot{q}1(t) \dot{q}2(t) \sin\left(q5(t) + \frac{\pi}{2} - q4(t) + q2(t)\right) + 0.002755000000000000 \dot{q}5(t) \dot{q}1(t) \cos\left(q5(t) \right. \\
& \left. + \frac{\pi}{2} + q3(t) + q2(t) - q4(t)\right) + 0.002755000000000000 \dot{q}5(t) \dot{q}1(t) \cos\left(q5(t) + \frac{3\pi}{2} - q3(t) - q2(t) + q4(t)\right) \\
& + 0.001305000000000000 \dot{q}1(t)^2 \sin(q5(t) + \pi - q4(t) + q3(t)) + 0.001305000000000000 \dot{q}1(t)^2 \sin(q5(t) + \pi + q4(t) \\
& - q3(t)) + 0.002610000000000000 \dot{q}2(t)^2 \sin(q5(t) + \pi + q4(t) + q3(t)) + 0.002610000000000000 \dot{q}2(t)^2 \sin(q5(t) + \pi \\
& - q4(t) - q3(t)) + 0.001305000000000000 \dot{q}1(t)^2 \sin(q5(t) + 2\pi - q4(t) - q3(t) - 2q2(t)) \\
& + 0.001305000000000000 \dot{q}1(t)^2 \sin(q5(t) + q4(t) + q3(t) + 2q2(t)) + 0.001377500000000000 \dot{q}1(t)^2 \cos(q5(t) \\
& + 2q3(t) + 2q2(t) + q4(t)) - 0.001377500000000000 \dot{q}1(t)^2 \cos(q5(t) + 2\pi - 2q3(t) - 2q2(t) - q4(t))
\end{aligned}$$

```
> #Matlab conversion
```

```
> with( CodeGeneration ) :
```

```
> Matlab( m11, resultname = "m11" );
```

```
> Matlab( m12, resultname = "m12" );
```

```
> Matlab( m13, resultname = "m13" );
```

```
> Matlab( m14, resultname = "m14" );
```

```
> Matlab( m15, resultname = "m15" );
```

```
> Matlab( m16, resultname = "m16" );
```

```
> Matlab( m21, resultname = "m21" );
```

```
> Matlab( m22, resultname = "m22" );
```

```
> Matlab( m23, resultname = "m23" );
```

```
L> Matlab(m24, resultname = "m24");
L> Matlab(m25, resultname = "m25");
L> Matlab(m26, resultname = "m26");
L>
L> Matlab(m31, resultname = "m31");
L> Matlab(m32, resultname = "m32");
L> Matlab(m33, resultname = "m33");
L> Matlab(m34, resultname = "m34");
L> Matlab(m35, resultname = "m35");
L> Matlab(m36, resultname = "m36");
L>
L>
L> Matlab(m41, resultname = "m41");
L> Matlab(m42, resultname = "m42");
L> Matlab(m43, resultname = "m43");
L> Matlab(m44, resultname = "m44");
L> Matlab(m45, resultname = "m45");
L> Matlab(m46, resultname = "m46");
L>
L>
L> Matlab(m51, resultname = "m51");
L> Matlab(m52, resultname = "m52");
L> Matlab(m53, resultname = "m53");
L> Matlab(m54, resultname = "m54");
L> Matlab(m55, resultname = "m55");
L> Matlab(m56, resultname = "m56");
L>
L>
L> Matlab(m61, resultname = "m61");
L> Matlab(m62, resultname = "m62");
L> Matlab(m63, resultname = "m63");
L> Matlab(m64, resultname = "m64");
L> Matlab(m65, resultname = "m65");
L> Matlab(m66, resultname = "m66");
L>
L>
L> Matlab(g1, resultname = "g1");
L> Matlab(g2, resultname = "g2");
L> Matlab(g3, resultname = "g3");
L> Matlab(g4, resultname = "g4");
L> Matlab(g5, resultname = "g5");
L> Matlab(g6, resultname = "g6");
L>
L>
L> Matlab(C0Dq(1), resultname = "cdq1");
L> Matlab(C0Dq(2), resultname = "cdq2");
L> Matlab(C0Dq(3), resultname = "cdq3");
L> Matlab(C0Dq(4), resultname = "cdq4");
L> Matlab(C0Dq(5), resultname = "cdq5");
L> Matlab(C0Dq(6), resultname = "cdq6");
L>
L>
L> Matlab(Ekin, resultname = "K")
```



# Appendix D

## Errata in Chapter 7.6 in [20]

Chapter 7.6 *Newton-Euler Formulation* in [20] has some misleading notations and errata. These are the findings, and the suggested corrections.

- Page 275, definition of  $R_{i+1}^i$  should be:  
 $R_{i+1}^i$  : the rotation matrix from frame  $i$  to frame  $i + 1$ .
- Page 276, definition of  $I_i$  should be:  
 $I_i$  : the inertia tensor of link  $i$  about a frame parallel to frame  $i$  whose origin is at the center of mass of link  $i$ .
- Page 276, definition of  $r_{i,ci}$  should be:  
 $r_{i-1,ci}$  : the vector from the origin of frame  $i - 1$  to the center of mass of link  $i$ .
- Page 276, definition of  $r_{i+1,ci}$  should be:  
 $r_{i,ci}$  : the vector from the origin of frame  $i$  to the center of mass of link  $i$ .
- Page 276, definition of  $r_{i,i+1}$  should be:  
 $r_{i-1,i}$  : the vector from the origin of frame  $i - 1$  to the origin of frame  $i$ .
- Page 277, Equation (7.145) should be:

$$\tau_i - R_{i+1}^i \tau_{i+1} + f_i \times r_{i-1,ci} - (R_{i+1}^i f_{i+1}) \times r_{i,ci} = I_i \alpha_i + \omega_i \times (I_i \omega_i) \quad (\text{D.1})$$

- Page 277, Equation (7.147) should be:

$$\tau_i = R_{i+1}^i \tau_{i+1} - f_i \times r_{i-1,ci} + (R_{i+1}^i f_{i+1}) \times r_{i,ci} + I_i \alpha_i + \omega_i \times (I_i \omega_i) \quad (\text{D.2})$$

- Page 278, Equation (7.153) should be:

$$\alpha_i = (R_i^{i-1})^T \alpha_{i-1} + b_i \ddot{q}_i + \omega_i \times b_i \dot{q}_i \quad (\text{D.3})$$

- Page 278, Equation (7.154) should be:

$$v_{c,i}^{(0)} = v_{e,i-1}^{(0)} + \omega_i^{(0)} \times r_{i-1,ci}^{(0)} \quad (\text{D.4})$$

- Page 278, the sentence between (7.154) and (7.155) should be:

To obtain an expression for the acceleration, we note that the vector  $r_{i-1,ci}^{(0)}$  is constant in frame  $i$ .

- Page 278, Equation (7.155) should be:

$$a_{c,i}^{(0)} = a_{e,i-1}^{(0)} \times r_{i-1,ci}^{(0)} + \omega_i^{(0)} \times (\omega_i^{(0)} \times r_{i-1,ci}^{(0)}) \quad (\text{D.5})$$

- Page 279, Equation (7.158) should be:

$$a_{c,i} = (R_i^{i-1})^T a_{e,i-1} + \dot{\omega}_i \times r_{i-1,ci} + \omega_i \times (\omega_i \times r_{i-1,ci}) \quad (\text{D.6})$$

- Page 279, the sentence between Equation (7.158) and Equation (7.159) should be:

Now, to find the acceleration of the end of link  $i$ , we can use Equation (7.158) with  $r_{i-1,i}$  replacing  $r_{i-1,ci}$ .

- Page 279, Equation (7.159) should be:

$$a_{e,i} = (R_i^{i-1})^T a_{e,i-1} + \dot{\omega}_i \times r_{i-1,i} + \omega_i \times (\omega_i \times r_{i-1,i}) \quad (\text{D.7})$$

- Page 279, Equation (7.162) for  $\omega_2$  should be:

$$\omega_2 = (\dot{q}_1 + \dot{q}_2)k \quad (\text{D.8})$$

- Page 280, Equation (7.163) should be:

$$r_{0,c1} = l_{c1}i, \quad r_{1,c1} = (l_{c1} - l_1)i, \quad r_{0,1} = l_1i \quad (\text{D.9})$$

- Page 280, Equation (7.164) should be:

$$r_{1,c2} = l_{c2}i, \quad r_{2,c2} = (l_{c2} - l_2)i, \quad r_{1,2} = l_2i \quad (\text{D.10})$$

- Page 280, Equation (7.166) should be:

$$g_1 = (R_1^0)^T g_j = -g \begin{bmatrix} \sin(q_1) \\ \cos(q_1) \\ 0 \end{bmatrix} \quad (\text{D.11})$$

- Page 280, Equation (7.168) should be:

$$a_{c,2} = (R_2^1)^T a_{e,1} + (\ddot{q}_1 + \ddot{q}_2)k \times l_{c2}i + (\dot{q}_1 + \dot{q}_2)k \times [(\dot{q}_1 + \dot{q}_2)k \times l_{c2}i] \quad (\text{D.12})$$

- Page 281, Equation (7.169) should be:

$$(R_2^1)^T a_{e,1} = \begin{bmatrix} \cos(q_2) & \sin(q_2) \\ -\sin(q_2) & \cos(q_2) \end{bmatrix} \begin{bmatrix} -l_1 \dot{q}_1^2 \\ l_1 \ddot{q}_1 \end{bmatrix} \quad (\text{D.13})$$

$$= \begin{bmatrix} -l_1 \dot{q}_1^2 \cos(q_2) + l_1 \ddot{q}_1 \sin(q_2) \\ l_1 \dot{q}_1^2 \sin(q_2) + l_1 \ddot{q}_1 \cos(q_2) \end{bmatrix} \quad (\text{D.14})$$

- Page 281, Equation (7.170) should be:

$$a_{c,2} = \begin{bmatrix} -l_1 \dot{q}_1^2 \cos(q_2) + l_1 \ddot{q}_1 \sin(q_2) - l_{c2}(\dot{q}_1 + \dot{q}_2)^2 \\ l_1 \dot{q}_1^2 \sin(q_2) + l_1 \ddot{q}_1 \cos(q_2) + l_{c2}(\ddot{q}_1 + \ddot{q}_2) \end{bmatrix} \quad (\text{D.15})$$

- Page 281, Equation (7.171) should be:

$$g_2 = -g \begin{bmatrix} \sin(q_1 + q_2) \\ \cos(q_1 + q_2) \end{bmatrix} \quad (\text{D.16})$$

- Page 281, Equation (7.174) should be:

$$\begin{aligned} \tau_2 = I_2(\ddot{q}_1 + \ddot{q}_2)k + [m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1^2 + m_2 l_1 l_{c2} \cos(q_2) \ddot{q}_1 \\ + m_2 l_{c2}^2 (\ddot{q}_1 + \ddot{q}_2) + m_2 l_{c2} g \cos(q_1 + q_2)]k \end{aligned} \quad (\text{D.17})$$

- Page 282, Equation (7.175) should be:

$$f_1 = R_2^1 f_2 + m_1 a_{c,1} - m_1 g_1 \quad (\text{D.18})$$

- Page 282, Equation (7.176) should be:

$$\tau_1 = R_2^1 \tau_2 - f_1 \times l_{c1} i + (R_2^1 f_2) \times (l_{c1} - l_1) i + I_1 \alpha_1 + \omega_1 \times (I_1 \omega_1) \quad (\text{D.19})$$

- Page 282, second sentence after Equation (7.176) should be:  
First,  $R_2^1 \tau_2 = \tau_2$ , since the rotation matrix does not affect the third components of vectors.

- Page 282, Equation (7.177) should be:

$$\tau_1 = \tau_2 - m_1 a_{c,1} \times l_{c1} i + m_1 g_1 \times l_{c1} i - (R_2^1 f_2) \times l_1 i + I_1 \alpha_1 \quad (\text{D.20})$$



# Appendix E

## Contents of Attached ZIP-File

### PDF Files

- Master's Thesis: A digital copy of this thesis.
- Project Work 2010: A digital copy of [6].
- IRB 140 Data Sheet: A digital copy of [15].
- IRB 140 Product Specification: A digital copy of [16].

### Maple Files

- 3dofFramework.mw: The automated framework.
- 6dofIRB140.mw: Derivation of the dynamic model of the IRB 140.

### Matlab and Simulink Files

- SixDofOpenLoop.mdl: The open loop Simulink model.
- SixDofPDreg.mdl: The closed loop Simulink model with PD controllers.
- irb140.m: Matlab code of the dynamic model (the *irb140* block).
- plot6dof.m: Matlab code creating the plots after a simulation.



# Bibliography

- [1] T. Bajd, M. Mihelj, J. Lenarčič, A. Stanovnik, and M. Munih. *Robotics*. Springer Verlag, 2010.
- [2] G.L. Baker and J.A. Blackburn. *The pendulum: a case study in physics*. Oxford University Press, USA, 2005.
- [3] J.G. Balchen, T. Andresen, and B.A. Foss. *Reguleringsteknikk*. Institutt for teknisk kybernetikk, 2004.
- [4] M.F. Beatty. *Principles of engineering mechanics*. Springer Verlag, 2006.
- [5] B. Dasgupta and P. Choudhury. A general strategy based on the Newton-Euler approach for the dynamic formulation of parallel manipulators. *Mechanism and Machine Theory*, 34(6):801–824, 1999.
- [6] H. Høifødt. *Dynamics of robot manipulators by the Newton-Euler formulation*, 5th year project work, 2010.
- [7] J.M. Hollerbach. A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity. *Systems, Man and Cybernetics, IEEE Transactions on*, 10(11):730–736, 2007.
- [8] H.K. Khalil. *Nonlinear Systems 3rd*. Prentice Hall, 2002.
- [9] Maplesoft. *Maple 15 - The Essential Tools for Mathematics and Modeling*, Last accessed June 13, 2011.  
<http://www.maplesoft.com/products/maple/index.aspx>.
- [10] Mathworks. *MATLAB - The Language Of Technical Computing*, Last accessed June 13, 2011.  
<http://www.mathworks.com/products/matlab/>.
- [11] Mathworks. *Simulink - Simulation and Model-Based Design*, Last accessed June 13, 2011.  
<http://www.mathworks.com/products/simulink/>.
- [12] SI metric. *Density of metals*, Last accessed May 25, 2011.  
[http://www.simetric.co.uk/si\\_metals.htm](http://www.simetric.co.uk/si_metals.htm).

- 
- [13] S. Pchelkin. Dynamic model of the irb 140 by the euler-lagrange formulation.
- [14] ABB Robotics. *ABB Product Guide for IRB 140*, Last accessed June 14, 2011.  
<http://www.abb.com/product/seitp327/7c4717912301eb02c1256efc00278a26.aspx?productLanguage=no&country=00>.
- [15] ABB Robotics. *Data sheet for IRB 140*, Retrieved September 21, 2010.
- [16] ABB Robotics. *Product specification for IRB 140*, Retrieved September 21, 2010.
- [17] L. Sciavicco and B. Siciliano. *Modelling and control of robot manipulators*. Springer Verlag, 2000.
- [18] T. Shinbrot, C. Grebogi, J. Wisdom, and J.A. Yorke. Chaos in a double pendulum. *American Journal of Physics*, 60:491, 1992.
- [19] W.M. Silver. On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators. *The International Journal of Robotics Research*, 1(2):60, 1982.
- [20] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*. Wiley New Jersey, 2006.
- [21] Adobe Systems. *Adobe Illustrator*, Last accessed June 13, 2011.  
<http://www.adobe.com/products/illustrator.html>.
- [22] J.R. Taylor. *Classical mechanics*. Univ Science Books, 2005.