Dynamics of robot manipulators by the Newton-Euler formulation

Herman Høifødt

Abstract

This text is about dynamic modeling of robot manipulators using the Newton-Euler formulation, and following simulations of the system responses. The algorithm of the Newton-Euler approach is introduced in general, and applied on a planar elbow manipulator of two degrees of freedom. The approach is also applied on a more complex robot manipulator with three degrees of freedom, showing that the formulation can result in interesting insights if correctly used. Specifically, the computations show that if coordinate frames are attached to the links of the manipulator in a specific manner, there will be simplifications in the equations which ensures more efficient computations of the dynamics.

Simulations are studied in both open loop and closed loop, and it is drawn a system connection between unactuated manipulators and the double pendulum problem. It is shown that there is no dissipation of energy in a released pendulum without friction. A PD-controller is included in both manipulators, and satisfying control of the joint variables are achieved. Towards the end, it is proved with a Lyapunov function that setpoint control with a PD-controller achieves asymptotic tracking of the state variables in this type of nonlinear system models.

In addition, a lab exercise in robotics that is going to be a part of the course *TTK 4100 Kybernetikk Introduksjon* is presented. This exercise includes the very basics of robotic theory, and practical tasks that are mainly meant to be fun and act as motivation to the field of robotics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Previous work and motivation

In the early 1980's, robot manipulators were touted as the ultimate solution to automated manufacturing. Since then the field of robotics has changed in numerous ways, and researchers have discovered that there are several ways to describe the motion of robot manipulators. Different algorithms lead to different insights, and it may not be easy to identify the algorithm which gives the most appreciated results. The physical shape of the manipulator and its composition of links and joints are factors to take into account when choosing how to describe the manipulator.

Computing the dynamics of robot manipulators can be challenging, and in general there are two procedures available; the Euler-Lagrange formulation and the Newton-Euler formulation. An introduction to the most important concepts in fundamental robotics including kinematics, dynamics, and control, are given in [12], [13] and [9]. Both Euler-Lagrange and Newton-Euler formulations are derived from first principles, and examples of how the methods are applied are presented in these three books. The efficiency of the two formulations are shown to be equivalent in [11]. For the case of parallel manipulators, it is shown in [6] that the Newton-Euler formulation gives an advantage for dynamics computations and control.

1.2 Outline

- **Chapter 2:** Basic definitions, theory and identities used in this report are described. The notation introduced, are used throughout the whole report.

- **Chapter 3:** A short description of a dynamic model is given, and different approaches for computing the dynamics are discussed. Moreover, the Newton-Euler formulation is described in depth, and general equations are derived.

- **Chapter 4:** The Newton-Euler formulation is applied on a planar elbow manipulator.
- **Chapter 5:** The Newton-Euler formulation is applied on an actual manipulator with three degrees of freedom, namely the IRB 140 manipulator produced by ABB.
- **Chapter 6:** Several simulations of both manipulators are presented, both in open loop and closed loop. Advantages with the PD-controller is discussed in detail.
- **Chapter 7:** A short description of my own attendance at The National Science Week in Trondheim with the Department of Engineering Cybernetics at NTNU.
- **Chapter 8:** Improvements and further work on the results of this project are discussed.
- **Chapter 9:** A conclusion about the work done in the project.
- **Chapter 10:** A lab exercise in basic robotics.

Chapter 2

Definitions, notation and background theory

This report follows the standard convention of how a manipulator is interpreted. This is explained carefully in [12], and this chapter gives a summary of the most important definitions, notation and background theory needed to understand this report.

2.1 General representation of a robot manipulator

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 how to represent joints in 2D and 3D by symbols.

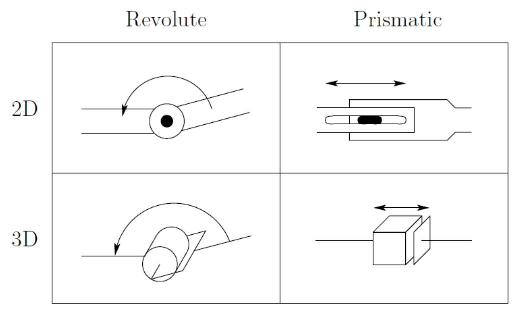

Figure 2.1: Symbolic representation of robot joints

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 of 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 system is rigidly attached to each link, and an inertial coordinate system 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.

2.2 Rotation matrices and skew symmetric matrices

A rotation matrix is used to specify the orientation of one coordinate frame relative to another coordinate frame. To specify the coordinate vectors of frame 1 with respect to frame 0 in three dimensions, the 3×3 rotation matrix is written as

$$R_1^0 = \begin{bmatrix} x_1^0 & y_1^0 & z_1^0 \end{bmatrix} \tag{2.1}$$

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

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
- $\det R = 1$

Vectors and points are denoted with a superscript to denote in which frame they are expressed. The rotation matrix R_1^0 can be used to transform the coordinates of a point or vector from one coordinate frame to another. Introducing a vector v and three frames, this relationship is given as

$$v^0 = R_1^0 v^1 \tag{2.2}$$
$$v^1 = R_2^1 v^2 \tag{2.3}$$
$$v^0 = R_2^0 v^2 \tag{2.4}$$

which gives

$$R_2^0 = R_1^0 R_2^1 \tag{2.5}$$

Four matrix indentities about rotation matrices and skew symmetric matrices should be mentioned. These identities are

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

   $$S(a)p = a \times p \tag{2.6}$$

   where S is a 3×3 skew symmetric matrix.

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

   $$RS(a)R^T = S(Ra) \tag{2.7}$$

   where S is a 3×3 skew symmetric matrix.

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

   $$\dot{R}(t) = S(\omega(t))R(t) \tag{2.8}$$

   where $R = R(t) \in SO(3)$ for every $t \in \mathbb{R}$, S is a 3×3 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 $X \in \mathbb{R}^n$

   $$X^T S X = 0 \tag{2.9}$$

Chapter 3

Newton-Euler Formulation

3.1 Introduction

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).

In the design of robots one needs to derive equations that explicitly describes the relationship between force and motion. These equations of motion are what is called the dynamic equations, and they are important to consider in simulation of robot motion, and in design of control algorithms.

There are mainly two ways to derive the dynamic equations of a robot manipulator. The most common method is known as the Euler-Lagrange formulation where we treat the manipulator as a whole, and analyze the system based on the kinetic and potential energy. The other method is known as the Newton-Euler formulation, which is quite different because each link of the robot in treated in turn. First there is a forward recursion from link 1 to link n to write down equations describing its linear and angular motion. Then there is a backward recursion to calculate the forces and torques.

Both the Euler-Lagrange formulation and the Newton-Euler formulation leads to the same dynamic model of the manipulator. The dynamic model can be written in matrix form as

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

where

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

3.2 Euler-Lagrange versus Newton-Euler

When it comes to the question of which method is better than the other, there is no clear answer. The main goal is to derive efficient formulations of the dynamics. How well this goal is satisfied will depend on the structure of the computation, together with how the rotations are represented. A recursive structure is desirable to achieve quick computations, which is exactly what the Newton-Euler formulation is all about. On the other side, in [7] it is shown that there is also possible to compute the Euler-Lagrange formulation in a recursive procedure, and it is proved that a recursive approach is indeed faster than treating the manipulator as a whole.

How rotations are represented will be determined by the position and orientation of the coordinate frames. In a recursive approach, one is constantly looking from one frame into neighboring frames, such that the recursions will be strictly dependent on the rotation matrices. If the frames are attached in a proper manner, the result is simplified equations and lower computation times. Such simplifications are achievable in both Newton-Euler and the recursive Euler-Lagrange. In [11] it is showed that when attaching frames in a proper manner, the computation time of the two algorithms are equivalent.

So how to decide which algorithm to use? If the manipulator to be analyzed has many degrees of freedom, the Newton-Euler may be a preferred choice since it is solely based on a recursive approach. With few degrees of freedom, one may as well choose the Euler-Lagrange. In the end it is a matter of personal preference, and the main reason for choosing one over the other is that it might provide different insights.

3.3 The general case

In this section the general equations of force and torque in the Newton-Euler formulation will be derived, based on the derivation in [12].

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 τ 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

$$\frac{d(mv)}{dt} = f \tag{3.2}$$

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 (3.2) can be simplified to

$$f = ma \tag{3.3}$$

where a is the acceleration of the center of mass.

The third law gives the relationship

$$\frac{d(I_0\omega_0)}{dt} = \tau_0 \tag{3.4}$$

where I_0 is the moment of inertia of the link, ω_0 is the angular velocity of the link, and τ_0 is the sum of torques applied to 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 avoided by rewriting Equation (3.4) to be valid for a frame rigidly attached to the the link instead of an inertial frame. A similarity transformation of I_0 is given by

$$I = R^{-1} I_0 R \tag{3.5}$$

which gives

$$I_0 = RIR^T \tag{3.6}$$

where R is the rotation matrix that transforms coordinates from the link attached frame to the inertial frame. Equation (3.6) together with the facts

$$\omega_0 = R\omega, \qquad \tau_0 = R\tau \tag{3.7}$$

yields

$$\begin{align} \frac{d(I_0\omega_0)}{dt} &= \frac{d(RIR^TR\omega)}{dt} \tag{3.8} \\ &= \frac{d(RI\omega)}{dt} \tag{3.9} \\ &= \dot{R}I\omega + RI\dot{\omega} \tag{3.10} \end{align}$$

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

$$\begin{aligned} \tau &= R^T \tau_0 && (3.11) \\ &= R^T(\dot{R}I\omega + RI\dot{\omega}) && (3.12) \\ &= R^T\dot{R}I\omega + I\dot{\omega} && (3.13) \end{aligned}$$

Equation (3.13) can be simplified to not include the rotation matrix, by using the indentities (2.6), (2.7) and (2.8). The final torque expression becomes

$$\begin{aligned} \tau &= R^T\dot{R}I\omega + I\dot{\omega} && (3.14) \\ &= R^T S(\omega_0)RI\omega + I\dot{\omega} && (3.15) \\ &= S(R^T\omega_0)I\omega + I\dot{\omega} && (3.16) \\ &= S(\omega)I\omega + I\dot{\omega} && (3.17) \\ &= \omega \times (I\omega) + I\dot{\omega} && (3.18) \end{aligned}$$

3.4 Equations of an n-link manipulator

In this section the general force and torque equations will be used to derive the Newton-Euler formulation of an n-link manipulator. This is done in [12], but due to numerous errors discovered in that derivation (see Appendix A), these equations are derived here from the beginning to the end. However, the procedure is the same.

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 $i+1$)
ω_i	=	angular velocity of frame i with respect to frame 0
α_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
τ_i	=	torque exerted by link $i-1$ on link i
R^i_{i+1}	=	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
$r_{i,ci}$	=	vector from the origin of frame i 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

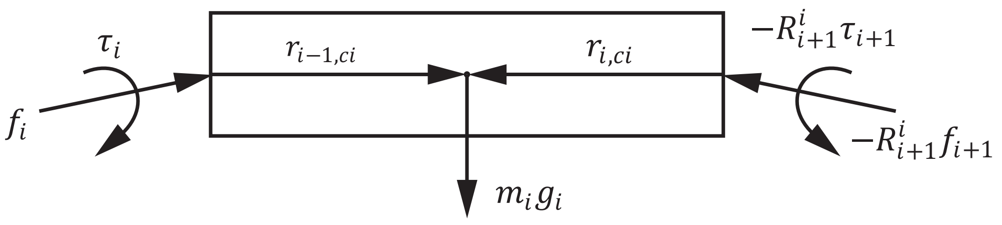

Figure 3.1: Forces and torques acting on a random link

Figure 3.1 shows a link of an unknown manipulator, together with all forces and torques acting on it. By the law of action and reaction, f_i is the force applied by link $i-1$ to link i, and $-f_{i+1}$ is the force applied by link $i+1$ to link i. According to the standard convention, f_i is expressed in frame i while $-f_{i+1}$ is expressed in frame $i+1$, so to express both forces in frame i it is needed to multiply the latter with R^i_{i+1}. The same apply to the torque, again by the law of action and reaction. Then all vectors in Figure 3.1 are expressed in frame i, and the force balance equation for the link can be stated as

$$\sum_{link} f = ma \tag{3.19}$$

$$f_i - R^i_{i+1}f_{i+1} + m_i g_i = m_i a_{c,i} \tag{3.20}$$

$$f_i = R^i_{i+1}f_{i+1} + m_i a_{c,i} - m_i g_i \tag{3.21}$$

Next, the moment balance equation for the link is 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 equation since it is applied directly at the center of mass. The equation becomes

$$\sum_{link} \tau = \omega \times (I\omega) + I\dot{\omega} \tag{3.22}$$

$$\tau_i - R^i_{i+1}\tau_{i+1} + f_i \times r_{i-1,ci} - (R^i_{i+1}f_{i+1}) \times r_{i,ci} = \omega_i \times (I_i\omega_i) + I_i\alpha_i \tag{3.23}$$

$$\tau_i = R^i_{i+1}\tau_{i+1} - f_i \times r_{i-1,ci} + (R^i_{i+1}f_{i+1}) \times r_{i,ci} + \omega_i \times (I_i\omega_i) + I_i\alpha_i \tag{3.24}$$

Solving Equation (3.24) recursively for decreasing i gives the final dynamic equations, but it needs to expressed only by q, $\dot{q}$, $\ddot{q}$ and constant parameters to transform the equation to the general matrix form in (3.1). That means it is necessary to find a relation between q, $\dot{q}$, $\ddot{q}$ and $a_{c,i}$, ω_i and α_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 $a_{c,i}$, ω_i and α_i. However, as a starting point, ω_i and α_i need to be expressed in the inertial frame, and the superscript (0) will be used to denote that. As an example, ω_i denotes the angular velocity of frame i expressed in frame i, while $\omega_i^{(0)}$ denotes the same quantity expressed in an inertial frame. This gives

$$\omega_i^{(0)} = \omega_{i-1}^{(0)} + z_{i-1}\dot{q}_i \tag{3.25}$$

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

$$\omega_i = (R_i^{i-1})^T\omega_{i-1} + b_i\dot{q}_i \tag{3.26}$$

where

$$b_i = (R_i^0)^T R_{i-1}^0 z_0 \tag{3.27}$$

is the rotation of joint i expressed in frame i.

For the angular acceleration it is important to note that

$$\alpha_i = (R_i^0)^T \dot{\omega}_i^{(0)} \tag{3.28}$$

which means $\alpha_i \neq \dot{\omega}_i$! By using Newtons Second Law in a rotating frame (see [14], page 342-343), the time derivative of Equation (3.25) becomes

$$\dot{\omega}_i^{(0)} = \dot{\omega}_{i-1}^{(0)} + z_{i-1}\ddot{q}_i + \omega_i^{(0)} \times z_{i-1}\dot{q}_i \tag{3.29}$$

and expressed in frame i it directly becomes

$$\alpha_i = (R_i^{i-1})^T\alpha_{i-1} + b_i\ddot{q}_i + \omega_i \times b_i\dot{q}_i \tag{3.30}$$

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

$$v_{c,i}^{(0)} = v_{e,i-1}^{(0)} + \omega_i^{(0)} \times r_{i-1,ci}^{(0)} \tag{3.31}$$

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

$$a_{c,i}^{(0)} = a_{e,i-1}^{(0)} \times r_{i-1,ci}^{(0)} + \omega_i^{(0)} \times (\omega_i^{(0)} \times r_{i-1,ci}^{(0)}) \tag{3.32}$$

Multiplying with rotation matrices and using the fact that

$$R(a \times b) = (Ra) \times (Rb) \tag{3.33}$$

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

$$a_{c,i} = (R_i^{i-1})^T a_{e,i-1} + \dot{\omega}_i \times r_{i-1,ci} + \omega_i \times (\omega_i \times r_{i-1,ci}) \tag{3.34}$$

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

$$a_{e,i} = (R_i^{i-1})^T a_{e,i-1} + \dot{\omega}_i \times r_{i-1,i} + \omega_i \times (\omega_i \times r_{i-1,i}) \tag{3.35}$$

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

$$\omega_0 = \alpha_0 = a_{c,0} = a_{e,0} = 0 \tag{3.36}$$

and solve Equations (3.26), (3.30), (3.35) and (3.34) (in that order) to compute ω_i, α_i and $a_{c,i}$ for increasing i from 1 to n.

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

$$f_{n+1} = \tau_{n+1} = 0 \tag{3.37}$$

and solve Equations (3.21) and (3.24) (in that order) for decreasing i from n to 1.

Chapter 4

Newton-Euler formulation: Planar elbow manipulator

In this section the Newton-Euler method is used to find the dynamic equations of the planar elbow manipulator shown in Figure (4.1). This is done in [12], but again due to errors (see Appendix A), the derivation is shown in detail. Later in chapter 6, some simulations are done in Matlab to study the behavior of the system.

This manipulator has two revolute joints, hence there are two joint variables, q_1 and q_2. Frame 0 is the inertial base frame, frame 1 is rigidly attached to link 1, and frame 2 is rigidly attached to link 2. All frames have the z-axis pointing perpendicular out of the paper. l_i is the length of link i, and l_{ci} is the length from the origin of frame $i-1$ to the center of mass of link i.

Some vectors are independent of the configuration of the manipulator and can be written down directly

$$r_{0,c1} = \begin{bmatrix} l_{c1} & 0 & 0 \end{bmatrix}^T \tag{4.1}$$

$$r_{1,c2} = \begin{bmatrix} l_{c2} & 0 & 0 \end{bmatrix}^T \tag{4.2}$$

$$r_{0,1} = \begin{bmatrix} l_1 & 0 & 0 \end{bmatrix}^T \tag{4.3}$$

$$r_{1,2} = \begin{bmatrix} l_2 & 0 & 0 \end{bmatrix}^T \tag{4.4}$$

$$r_{1,c1} = \begin{bmatrix} l_{c1} - l_1 & 0 & 0 \end{bmatrix}^T \tag{4.5}$$

$$r_{2,c2} = \begin{bmatrix} l_{c2} - l_2 & 0 & 0 \end{bmatrix}^T \tag{4.6}$$

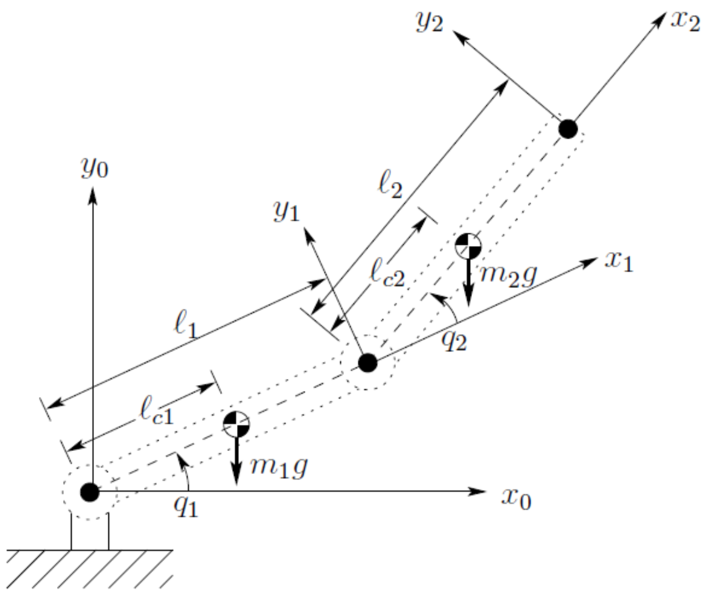

Figure 4.1: Planar elbow manipulator

All rotation matrices describe a rotation around the z-axis, such that

$$R_1^0 = \begin{bmatrix} cos(q_1) & -sin(q_1) & 0 \\ sin(q_1) & cos(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.7}$$

$$R_2^1 = \begin{bmatrix} cos(q_2) & -sin(q_2) & 0 \\ sin(q_2) & cos(q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

$$R_2^0 = R_1^0 R_2^1 = \begin{bmatrix} cos(q_1+q_2) & -sin(q_1+q_2) & 0 \\ sin(q_1+q_2) & cos(q_1+q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

and the gravity vector in the inertial frame is

$$g_0 = \begin{bmatrix} 0 & -g & 0 \end{bmatrix}^T \tag{4.10}$$

In the general case of Newton-Euler, the inertia tensor I_i is defined about a frame parallel to frame i whose origin is at the center of mass of link i. In other words, the general inertia tensor represents rotations in all dimensions. For the planar elbow manipulator, rotations appear only in one dimension. Therefore, in this example, I_i can be simplified to be the scalar moment of inertia of link i about an axis parallel to z_i that goes through the center of mass of link i.

The planar elbow manipulator has only two degrees of freedom, and it is quite easy to see that the angular velocity and angular acceleration of each link (with respect to frame 0) becomes

$$\omega_1 = \begin{bmatrix}0 & 0 & \dot{q}_1\end{bmatrix}^T \tag{4.11}$$

$$\alpha_1 = \begin{bmatrix}0 & 0 & \ddot{q}_1\end{bmatrix}^T \tag{4.12}$$

$$\omega_2 = \begin{bmatrix}0 & 0 & \dot{q}_1 + \dot{q}_2\end{bmatrix}^T \tag{4.13}$$

$$\alpha_2 = \begin{bmatrix}0 & 0 & \ddot{q}_1 + \ddot{q}_2\end{bmatrix}^T \tag{4.14}$$

Note that for manipulators with more degrees of freedom it will not be that easy to see these vectors directly, and we would have to solve Equation (3.26) and (3.30) as part of the forward recursion.

4.1 Forward recursion: Link 1

Using Equation (3.35), noting that $a_{e,0} = 0$ and $\dot{\omega}_1 = \alpha_1$, the acceleration of the end of the link becomes

$$a_{e,1} = (R_1^0)^T a_{e,0} + \dot{\omega}_1 \times r_{0,1} + \omega_1 \times (\omega_1 \times r_{0,1}) \tag{4.15}$$

$$= \alpha_1 \times r_{0,1} + \omega_1 \times (\omega_1 \times r_{0,1}) \tag{4.16}$$

$$= \begin{bmatrix}-l_1\dot{q}_1^2 \\ l_1\ddot{q}_1 \\ 0\end{bmatrix} \tag{4.17}$$

Using Equation (3.34), the acceleration of the center of mass of the link becomes

$$a_{c,1} = (R_1^0)^T a_{e,0} + \dot{\omega}_1 \times r_{0,c1} + \omega_1 \times (\omega_1 \times r_{0,c1}) \tag{4.18}$$

$$= \alpha_1 \times r_{0,c1} + \omega_1 \times (\omega_1 \times r_{0,c1}) \tag{4.19}$$

$$= \begin{bmatrix}-l_{c1}\dot{q}_1^2 \\ l_{c1}\ddot{q}_1 \\ 0\end{bmatrix} \tag{4.20}$$

The gravity vector is

$$g_1 = (R_1^0)^T g_0 = -g \begin{bmatrix}sin(q_1) \\ cos(q_1) \\ 0\end{bmatrix} \tag{4.21}$$

4.2 Forward recursion: Link 2

Since there are only two links, there is no need to compute $a_{e,2}$ (this term only appears in $a_{c,3}$ which is zero). The acceleration of the center of mass of the link becomes

$$\begin{aligned} a_{c,2} &= (R_2^1)^T a_{e,1} + \dot{\omega}_2 \times r_{1,c2} + \omega_2 \times (\omega_2 \times r_{1,c2}) && (4.22)\\ &= (R_2^1)^T a_{e,1} + \alpha_2 \times r_{1,c2} + \omega_2 \times (\omega_2 \times r_{1,c2}) && (4.23)\\ &= \begin{bmatrix} -l_1\dot{q}_1^2 cos(q_2) + l_1\ddot{q}_1 sin(q_2) - l_{c2}(\dot{q}_1+\dot{q}_2)^2 \\ l_1\dot{q}_1^2 sin(q_2) + l_1\ddot{q}_1 cos(q_2) + l_{c2}(\ddot{q}_1+\ddot{q}_2) \\ 0 \end{bmatrix} && (4.24) \end{aligned}$$

The gravity vector is

$$g_2 = (R_2^0)^T g_0 = -g \begin{bmatrix} sin(q_1+q_2) \\ cos(q_1+q_2) \\ 0 \end{bmatrix} \tag{4.25}$$

4.3 Backward recursion: Link 2

Using Equation (3.21), noting that f_3=0, the force exerted on the link becomes

$$f_2 = m_2 a_{c2} - m_2 g_2 \tag{4.26}$$

Using Equation (3.24) the torque equation for the link can be stated as

$$\tau_2 = R_3^2 \tau_3 - f_2 \times r_{1,c2} + (R_3^2 f_3) \times r_{2,c2} + \omega_2 \times (I_2\omega_2) + I_2\alpha_2 \tag{4.27}$$

This equation can be simplified by noting that $\tau_3 = f_3 = 0$, and $\omega_2 \times (I_2\omega_2) = 0$ because both vectors in the cross product are aligned in the z-direction. Substituting for f_2 then gives the final torque equation as

$$\begin{aligned} \tau_2 &= -f_2 \times r_{1,c2} + I_2\alpha_2 && (4.28)\\ &= (m_2 g_2 - m_2 a_{c,2}) \times r_{1,c2} + I_2\alpha_2 && (4.29)\\ &= \begin{bmatrix} 0 \\ 0 \\ [m_2 l_{c2}^2 + I_2 + m_2 l_1 l_{c2} cos(q_2)]\ddot{q}_1 + (I_2 + m_2 l_{c2}^2)\ddot{q}_2 \\ +m_2 l_1 l_{c2} sin(q_2)\dot{q}_1^2 + m_2 l_{c2} g cos(q_1+q_2) \end{bmatrix} && (4.30) \end{aligned}$$

4.4 Backward recursion: Link 1

The force exerted on the link becomes

$$f_1 = R_2^1 f_2 + m_1 a_{c,1} - m_1 g_1 \tag{4.31}$$

and the torque equation becomes

$$\tau_1 = R_2^1 \tau_2 - f_1 \times r_{0,c1} + (R_2^1 f_2) \times r_{1,c1} + \omega_1 \times (I_1 \omega_1) + I_1 \alpha_1 \tag{4.32}$$

This equation can be simplified by noting that $R_2^1 \tau_2 = \tau_2$ since the rotation matrix does not affect the third components of the vectors. In addition, $\omega_1 \times (I_1 \omega_1) = 0$ for the same reason as for link 2. Substituting for f_1 then gives the final torque equation as

$$\begin{aligned}
\tau_1 &= \tau_2 - f_1 \times r_{0,c1} + (R_2^1 f_2) \times r_{1,c1} + I_1 \alpha_1 && (4.33)\\
&= \tau_2 + (m_1 g_1 - R_2^1 f_2 - m_1 a_{c,1}) \times r_{0,c1} + (R_2^1 f_2) \times r_{1,c1} + I_1 \alpha_1 && (4.34)\\
&= \tau_2 + m_1 g_1 \times r_{0,c1} - m_1 a_{c,1} \times r_{0,c1} + (R_2^1 f_2) \times r_{1,c1} - (R_2^1 f_2) \times r_{0,c1} + I_1 \alpha_1 && (4.35)\\
&= \tau_2 + m_1 g_1 \times r_{0,c1} - m_1 a_{c,1} \times r_{0,c1} - (R_2^1 f_2) \times r_{0,1} + I_1 \alpha_1 && (4.36)
\end{aligned}$$

where the transition from Equation (4.35) to Equation (4.36) comes from algebraic properties of the cross product, and noting that $r_{1,c1} - r_{0,c1} = -r_{0,1}$, giving

$$\begin{aligned}
(R_2^1 f_2) \times r_{1,c1} - (R_2^1 f_2) \times r_{0,c1} &= && (4.37)\\
(R_2^1 f_2) \times r_{1,c1} + (R_2^1 f_2) \times -r_{0,c1} &= && (4.38)\\
(R_2^1 f_2) \times (r_{1,c1} - r_{0,c1}) &= && (4.39)\\
(R_2^1 f_2) \times -r_{0,1} &= && (4.40)\\
-(R_2^1 f_2) \times r_{0,1} & && (4.41)
\end{aligned}$$

Calculating Equation X then gives the final torque equation as

$$\tau_1 = \begin{bmatrix} 0 \\ 0 \\ [2m_2 l_1 l_{c2} cos(q_2) + m_2 l_{c2}^2 + m_2 l_1^2 + m_1 l_{c1}^2 + I_1 + I_2]\ddot{q}_1 \\ +[m_2 l_1 l_{c2} cos(q_2) + m_2 l_{c2}^2 + I_2]\ddot{q}_2 - 2m_2 l_1 l_{c2} sin(q_2)\dot{q}_1\dot{q}_2 \\ -m_2 l_1 l_{c2} sin(q_2)\dot{q}_2^2 + m_2 l_1 g cos(q_1) \\ +m_1 l_{c1} g cos(q_1) + m_2 l_{c2} g cos(q_1 + q_2) \end{bmatrix} \tag{4.42}$$

4.5 Setting up the model

Both τ_1 and τ_2 are zero in the x-direction and y-direction. This comes from the fact that the manipulator is planar, and all rotations appear in parallel axes with the z-direction whatever configuration the manipulator is in. Furthermore it means that the joints do not need to be physically built to resist any torque in the x-direction and y-direction.

The dynamic model can be written in the general matrix form of Equation (3.1) by rearranging terms. The nonlinear system becomes

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u \tag{4.43}$$

with

$$q = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T, \qquad u = \begin{bmatrix} \tau_1 & \tau_2 \end{bmatrix}^T \tag{4.44}$$

and

$$M = \begin{bmatrix} 2m_2l_1l_{c2}cos(q_2) + m_2l_{c2}^2 + m_2l_1^2 + m_1l_{c1}^2 + I_1 + I_2 & m_2l_1l_{c2}cos(q_2) + m_2l_{c2}^2 + I_2 \\ m_2l_{c2}^2 + I_2 + m_2l_1l_{c2}cos(q_2) & I_2 + m_2l_{c2}^2 \end{bmatrix} \tag{4.45}$$

$$C = \begin{bmatrix} -2m_2l_1l_{c2}sin(q_2)\dot{q}_2 & -m_2l_1l_{c2}sin(q_2)\dot{q}_2 \\ m_2l_1l_{c2}sin(q_2)\dot{q}_1 & 0 \end{bmatrix} \tag{4.46}$$

$$g = \begin{bmatrix} m_2l_1gcos(q_1) + m_1l_{c1}gcos(q_1) + m_2l_{c2}gcos(q_1+q_2) \\ m_2l_{c2}gcos(q_1+q_2) \end{bmatrix} \tag{4.47}$$

All recursive calculations are done in Maple, and details can be found in Appendix B.

Chapter 5

6-DOF manipulator IRB 140

ABB has produced an industrial robot manipulator named IRB 140-6/0.8. It has a total of six revolute joints that are controlled by AC-motors. The manipulator 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 link 3. Figure 5.1 shows the manipulator with its base, links and joints in the reference configuration as described in the product specification [4].

In this section the Newton-Euler formulation is applied to compute the dynamics of the first three degrees of freedom. The physical shape of the manipulator is very irregular, and it is chosen to represent all links as rectangular prisms. Figure 5.2 shows a symbolic representation of the links and joints, and note that link 1 has an offset that complicates the model. Frames are attached following the standard convention such that the z-axis of frame i is the axis of actuation for joint $i+1$. In addition, the frames are oriented such that my reference configuration is the same as defined by ABB.

As for the planar elbow manipulator, the first step is to write down the vectors

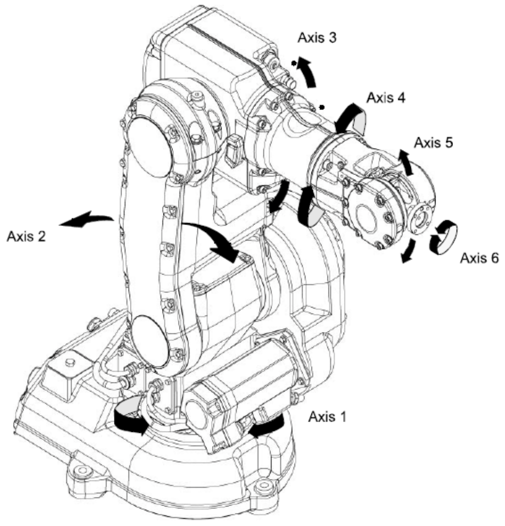

Figure 5.1: Sketch of the IRB 140

that are independent of the configuration.

$$r_{0,c1} = \begin{bmatrix} l_{c1,x} & -l_{c1,y} & 0 \end{bmatrix}^T \tag{5.1}$$

$$r_{1,c2} = \begin{bmatrix} 0 & l_{c2} & 0 \end{bmatrix}^T \tag{5.2}$$

$$r_{2,c3} = \begin{bmatrix} l_{c3} & 0 & 0 \end{bmatrix}^T \tag{5.3}$$

$$r_{0,1} = \begin{bmatrix} l_{1,x} & -l_{1,y} & 0 \end{bmatrix}^T \tag{5.4}$$

$$r_{1,2} = \begin{bmatrix} 0 & l_2 & 0 \end{bmatrix}^T \tag{5.5}$$

$$r_{2,3} = \begin{bmatrix} l_3 & 0 & 0 \end{bmatrix}^T \tag{5.6}$$

$$r_{1,c1} = \begin{bmatrix} l_{c1,x} - l_{1,x} & l_{1,y} - l_{c1,y} & 0 \end{bmatrix}^T \tag{5.7}$$

$$r_{2,c2} = \begin{bmatrix} 0 & l_{c2} - l_2 & 0 \end{bmatrix}^T \tag{5.8}$$

$$r_{3,c3} = \begin{bmatrix} l_{c3} - l_3 & 0 & 0 \end{bmatrix}^T \tag{5.9}$$

The gravity vector in the inertial frame is

$$g_0 = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T \tag{5.10}$$

The rotation matrices are products of basic rotations around the z-axis and the x-axis. It can be summarized as

- R_1^0: Rotate q_1 around the z-axis, rotate $\frac{\pi}{2}$ around the x-axis, rotate $\frac{\pi}{2}$ around the z-axis (in that order).

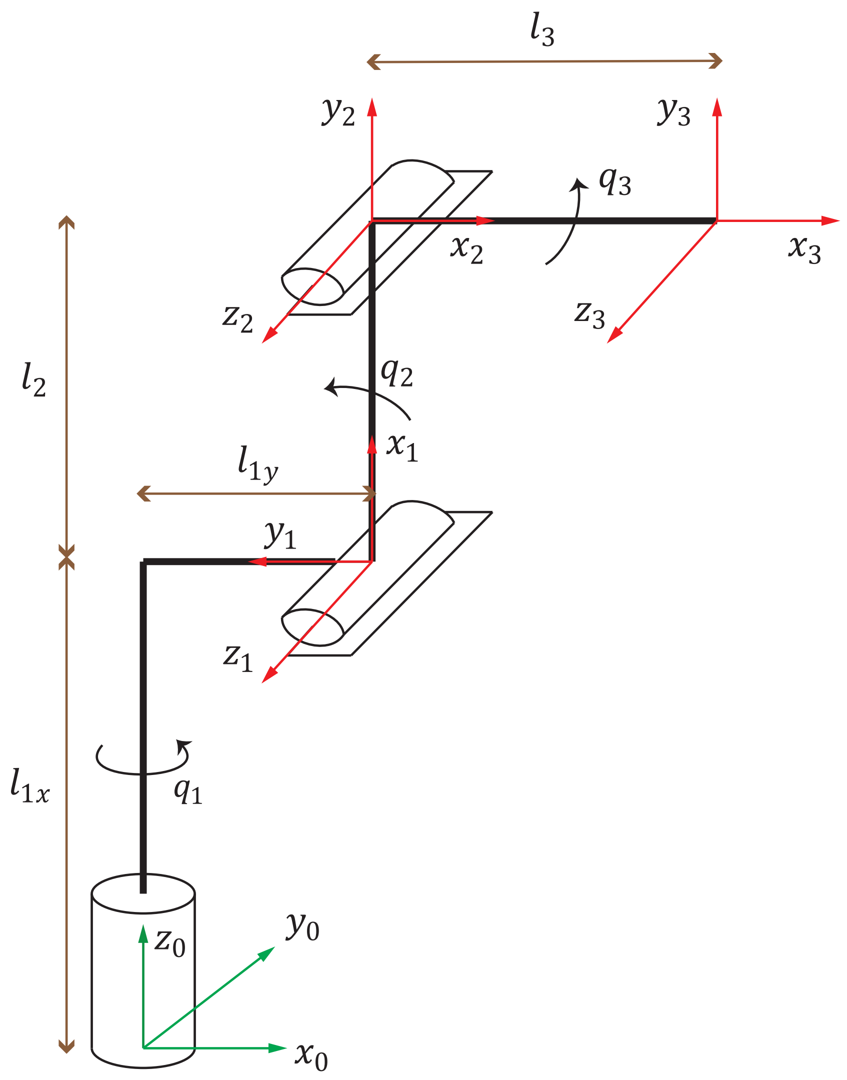

Figure 5.2: Symbolic representation of the first 3-DOF

- R_2^1: Rotate $q_2 - \frac{\pi}{2}$ around the z-axis.

- R_3^2: Rotate q_3 around the z-axis.

The result becomes

$$R_1^0 = \begin{bmatrix} cos(q_1) & -sin(q_1) & 0 \\ sin(q_1) & cos(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.11}$$

$$= \begin{bmatrix} 0 & -cos(q_1) & sin(q_1) \\ 0 & -sin(q_1) & -cos(q_1) \\ 1 & 0 & 0 \end{bmatrix} \tag{5.12}$$

$$R_2^1 = \begin{bmatrix} cos(q_2 - \frac{\pi}{2}) & -sin(q_2 - \frac{\pi}{2}) & 0 \\ sin(q_2 - \frac{\pi}{2}) & cos(q_2 - \frac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.13}$$

$$R_3^2 = \begin{bmatrix} cos(q_3) & -sin(q_3) & 0 \\ sin(q_3) & cos(q_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.14}$$

The inertia tensors for the links are defined about a frame parallel to frame i, whose origin is at the center of mass of link i. By assuming that the mass distribution of the link is symmetric with respect to this frame, all cross products of the inertia tensors become zero. This gives

$$I_1 = \begin{bmatrix} I_{1xx} & 0 & 0 \\ 0 & I_{1yy} & 0 \\ 0 & 0 & I_{1zz} \end{bmatrix}, I_2 = \begin{bmatrix} I_{2xx} & 0 & 0 \\ 0 & I_{2yy} & 0 \\ 0 & 0 & I_{2zz} \end{bmatrix}, I_3 = \begin{bmatrix} I_{3xx} & 0 & 0 \\ 0 & I_{3yy} & 0 \\ 0 & 0 & I_{3zz} \end{bmatrix} \tag{5.15}$$

To carry out the forward recursions it is necessary to know what the axis of rotation is for each joint i expressed in frame i. This can be computed right away for all joints by Equation (3.27). The rotation axis in frame 0 is given directly as

$$z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{5.16}$$

and then the rotation axes can be computed as

$$b_1 = (R_1^0)^T z_0 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \tag{5.17}$$

$$b_2 = (R_2^0)^T R_1^0 z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{5.18}$$

$$b_3 = (R_3^0)^T R_2^0 z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{5.19}$$

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 5.2, it can be seen that when looking from frame i into frame $i-1$, the angular velocity ω_i does not depend on q_i itself, but on the axis of rotation. Consequently, the rotation axes for this case are not depending on q, and this underlines one of the great advantages of the Newton-Euler formulation.

After the rotation matrices, the link length vectors, and the inertia tensors are defined correctly, the forward and backward recursions are actually just a matter of substituting in the general formulas of an n-link manipulator derived in chapter 3. For the planar elbow manipulator in chapter 4 the recursions were carefully computed showing every step in the procedure. The computations are not followed in such detail for the IRB 140 manipulator, due to the size of the equations. Important equations and results are shown, and details are stated in Appendix B.

5.1 Forward recursion: Link 1

Angular velocity and acceleration becomes

$$\omega_1 = b_1\dot{q}_1 \tag{5.20}$$
$$\alpha_1 = b_1\ddot{q}_1 + \omega_1 \times b_1\dot{q}_1 \tag{5.21}$$

Acceleration of the center and the end of the link becomes

$$a_{e,1} = \dot{\omega}_1 \times r_{0,1} + \omega_1 \times (\omega_1 \times r_{0,1}) \tag{5.22}$$
$$a_{c,1} = \dot{\omega}_1 \times r_{0,c1} + \omega_1 \times (\omega_1 \times r_{0,c1}) \tag{5.23}$$

The gravity vector becomes

$$g_1 = (R_1^0)^T g_0 \tag{5.24}$$

5.2 Forward recursion: Link 2

Angular velocity and acceleration are computed as

$$\omega_2 = (R_2^1)^T\omega_1 + b_2\dot{q}_2 \tag{5.25}$$
$$\alpha_2 = (R_2^1)^T\alpha_1 + b_2\ddot{q}_2 + \omega_2 \times b_2\dot{q}_2 \tag{5.26}$$

Acceleration of the center and the end of the link becomes

$$a_{e,2} = (R_2^1)^T a_{e,1} + \dot{\omega}_2 \times r_{1,2} + \omega_2 \times (\omega_2 \times r_{1,2}) \tag{5.27}$$
$$a_{c,2} = (R_2^1)^T a_{e,1} + \dot{\omega}_2 \times r_{1,c2} + \omega_2 \times (\omega_2 \times r_{1,c2}) \tag{5.28}$$

The gravity vector becomes

$$g_2 = (R_2^0)^T g_0 \tag{5.29}$$

5.3 Forward recursion: Link 3

Angular velocity and acceleration are computed as

$$\omega_3 = (R_3^2)^T\omega_2 + b_3\dot{q}_3 \tag{5.30}$$

$$\alpha_3 = (R_3^2)^T\alpha_2 + b_3\ddot{q}_3 + \omega_3 \times b_3\dot{q}_3 \tag{5.31}$$

Acceleration of the center and the end of the link becomes

$$a_{e,3} = (R_3^2)^T a_{e,2} + \dot{\omega}_3 \times r_{2,3} + \omega_3 \times (\omega_3 \times r_{2,3}) \tag{5.32}$$

$$a_{c,3} = (R_3^2)^T a_{e,2} + \dot{\omega}_3 \times r_{2,c3} + \omega_3 \times (\omega_3 \times r_{2,c3}) \tag{5.33}$$

The gravity vector becomes

$$g_3 = (R_3^0)^T g_0 \tag{5.34}$$

5.4 Backward recursion: Link 3

The force exerted on the link becomes

$$f_3 = m_3 a_{c,3} - m_3 g_3 \tag{5.35}$$

and the torque becomes

$$\tau_3 = -f_3 \times r_{2,c3} + I_3\alpha_3 + \omega_3 \times (I_3\omega_3) \tag{5.36}$$

Note that only the component in the z-direction of the torque is affecting q_3. The x-component and the y-component express the torque load in the x-direction and y-direction respectively. In other words, the joint has to physically resist torque induced by the motion of other joints, in directions different by its own rotation axis. If this is not fulfilled, the joint will break.

The dynamics will only contain the component that induces motion. The whole vector τ_3 is given in Appendix B because of the big size.

5.5 Backward recursion: Link 2

The force exerted on the link becomes

$$f_2 = R_3^2 f_3 + m_2 a_{c,2} - m_2 g_2 \tag{5.37}$$

and the torque becomes

$$\tau_2 = R_3^2\tau_3 - f_2 \times r_{1,c2} + R_3^2 f_3 \times r_{2,c2} + I_2\alpha_2 + \omega_2 \times (I_2\omega_2) \tag{5.38}$$

About the the torque components, the same applies to τ_2 as for τ_3.

5.6 Backward recursion: Link 1

The force exerted on the link becomes

$$f_1 = R_2^1 f_2 + m_1 a_{c,1} - m_1 g_1 \tag{5.39}$$

and the torque becomes

$$\tau_1 = R_2^1 \tau_2 - f_1 \times r_{0,c1} + R_2^1 f_2 \times r_{1,c1} + I_1 \alpha_1 + \omega_1 \times (I_1 \omega_1) \tag{5.40}$$

By the definition of coordinate frames, it is the component in the x-direction of τ_1 that affects q_1. The same principle is the case, the difference is that another component of the torque is affecting the dynamics.

All recursive calculations are done in Maple, and all equation and details can be found in Appendix B.

Chapter 6

Simulations

6.1 Introduction

All simulations are done in Simulink, a tool for modeling, simulating and analyzing dynamic systems. In Simulink a so-called "Level-2 Matlab S-Function" is used, which is a block with multiple input and output ports. Input 1 is the state vector x, input 2 is the torque vector u, and the output is the vector of state derivatives $\dot{x}$. In the M-file representing the S-Function block all constant parameters in the system are defined (e.g. masses, lengths and gravity), and for each time step in the simulation the updated vector of state derivatives is computed from the inputs and the constant parameters. The Dormand-Prince method ode45 with relative tolerance 1^{-10} is used for solving the differential equations.

6.1.1 Reducing system order

To simulate the systems in Matlab its necessary to express them in the first-order nonlinear form

$$\dot{x} = f(x, u) \tag{6.1}$$

where x is the state variable vector and u is the torque vector. Rearranging terms gives

$$\ddot{q} = M^{-1}(-C\dot{q} - g + u) \tag{6.2}$$

where it is assumed that the inertia matrix M is invertible. The inertia matrix is the main part of the kinetic energy expression $\frac{1}{2}\dot{q}^T M(q)\dot{q}$. 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. This is described in more detail in [12] page 250-254.

The system order needs to be reduced from n second-order equations to $2n$ first-order equations. Defining

$$\begin{aligned} x_1 &= q_1, & x_2 &= \dot{x}_1 = \dot{q}_1 && (6.3)\\ x_3 &= q_2, & x_4 &= \dot{x}_3 = \dot{q}_2 && (6.4)\\ &\vdots & &\vdots & \\ x_{2n-1} &= q_n, & x_{2n} &= \dot{x}_{2n-1} = \dot{q}_n && (6.5) \end{aligned}$$

I can express the system in the form (6.1) as

$$\begin{aligned} \dot{x}_1 &= x_2 && (6.6)\\ \dot{x}_2 &= f_2(x,u) && (6.7)\\ \dot{x}_3 &= x_4 && (6.8)\\ \dot{x}_4 &= f_4(x,u) && (6.9)\\ &\vdots \\ \dot{x}_{2n-1} &= x_{2n} && (6.10)\\ \dot{x}_{2n} &= f_{2n}(x,u) && (6.11) \end{aligned}$$

where $f_i(x,u)$ is the expressions in Equation (6.2) for $\ddot{q}_{\frac{i}{2}}$, substituting q and $\dot{q}$ with x.

6.2 Planar elbow manipulator

6.2.1 Reduced system

By defining

$$\begin{aligned} x_1 &= q_1, & x_2 &= \dot{x}_1 = \dot{q}_1 && (6.12)\\ x_3 &= q_2, & x_4 &= \dot{x}_3 = \dot{q}_2 && (6.13) \end{aligned}$$

the system becomes

$$\begin{aligned} \dot{x}_1 &= x_2 && (6.14)\\ \dot{x}_2 &= f_2(x,u) && (6.15)\\ \dot{x}_3 &= x_4 && (6.16)\\ \dot{x}_4 &= f_4(x,u) && (6.17) \end{aligned}$$

where $f_2(x,u)$ and $f_4(x,u)$ are big expressions (See appendix B).

6.2.2 Parameters

For the simulations, realistic parameters need to be chosen. It is assumed that both links are identical with the parameters shown in Table 6.1, and the shape of the links are rectangular prisms of steel with uniform mass density ρ. This implies that the geometric center in each link is also its center of mass. Dropping the

Symbol	Description	Value
m	Mass	15 kg
l	Length	30 cm
ρ	Mass density (steel)	7850 $\frac{kg}{m^3}$

Table 6.1: Planar elbow manipulator link parameters

indices for notation simplicity, the width w of the links will be

$$w = \sqrt{\frac{m}{\rho l}} = \sqrt{\frac{15}{7850 \cdot 0.3}} = 7.98 \text{ cm} \tag{6.18}$$

which gives an idea of how the manipulator looks. The inertia tensors for the links will be identical, and can be calculated from the formula[1]

$$I_{zz} = I_1 = I_2 = \iiint (x^2 + y^2)\rho(x, y, z) dx\, dy\, dz \tag{6.19}$$

$$= \rho \int_{-\frac{w}{2}}^{\frac{w}{2}} \int_{-\frac{w}{2}}^{\frac{w}{2}} \int_{-\frac{l}{2}}^{\frac{l}{2}} (x^2 + y^2) dx\, dy\, dz \tag{6.20}$$

$$= \frac{1}{12} m(l^2 + w^2) \tag{6.21}$$

where I have used the fact that $m = \rho V$.

To achieve realistic simulations it is necessary to know about the maximum torques of the motors. Some specifications for the IRB 140 manipulator is given in its manual [4], and some of these specifications will be applied to the planar manipulator as well. The maximum velocity of axes 1 and 2 in the IRB 140 is 200° per second, and the joints are driven by AC-motors. Unfortunately, the manual does not say anything about the specifications of the motors, but through another trustworthy source[2]the nominal torque of the AC-motors in joint 1 and 2 is given

[1] see [5]

[2] Uwe Mettin, Reasearch Scientist in Robotics and Control Engineering. http://www.itk.ntnu.no/ansatte/Mettin_Uwe

as 1.9 Nm. This kind of motors will be used for both joints in the planar elbow manipulator. Further it is assumed that the motors has 0.25 horsepowers each, as this is a common size for standard AC-motors[3]. Horsepower related to torque are defined as[4]:

$$\text{Horsepower} = \frac{\text{Torque} \cdot \text{RPM}}{5252} \tag{6.22}$$

such that the maximum velocity of the motor is found to be

$$\text{RPM} = \frac{5252 \cdot \text{Horsepower}}{\text{Torque}} \tag{6.23}$$

$$= \frac{5252 \cdot 0.25}{1.9} = 691 \tag{6.24}$$

which equals 4146° per second. The gear ratio is then

$$\frac{4146}{200} = 20.73 \tag{6.25}$$

and finally the maximum input torque in the model becomes

$$20.73 \cdot 1.9 \text{ Nm} = 39.4 \text{ Nm} \tag{6.26}$$

6.2.3 Open loop

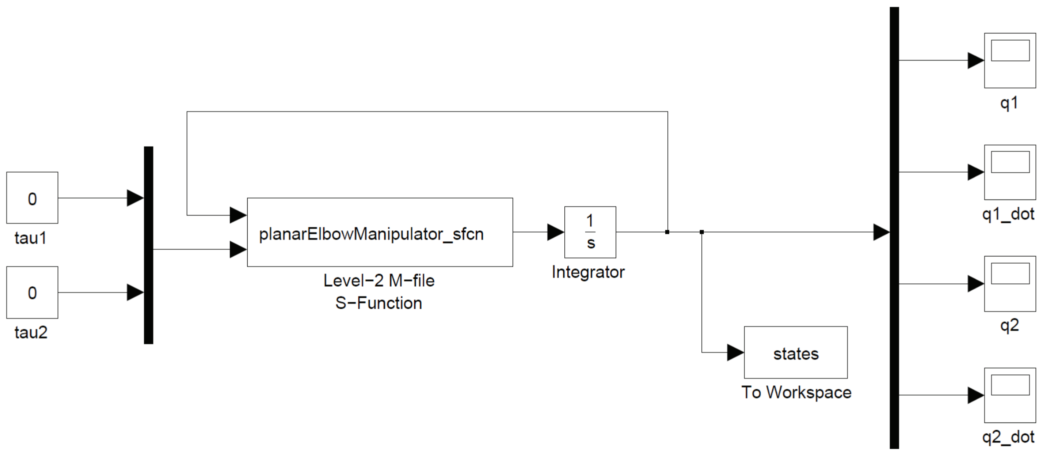

Figure 6.1: Open-loop simulink model

In open loop there are no feedback from the system output. The behavior of the system can be studied by driving the system with the desired torque, that

[3]see [2]
[4]see [1]

is the torque derived when substituting in the dynamic equations for the desired joint variables and derivatives. Because the gravity is dependent on the actual position, which there is no information about, it is quite intuitive that controlling the system in open loop should be impossible. This is verified by the following simulations.

First, let the initial conditions be $q = \dot{q} = 0$, and the desired position be $q_1 = \pi, q_2 = 0$ with $\dot{q} = 0$. By substituting the desired positions and velocities in the dynamic equations, the control torque becomes

$$\tau_1 = -m_2 l_1 g - l_{c2} m_2 g - m_1 l_{c1} g \tag{6.27}$$

$$\tau_2 = -l_{c2} m_2 g \tag{6.28}$$

Simulating the system gives the response shown in Figure 6.2 which becomes un-

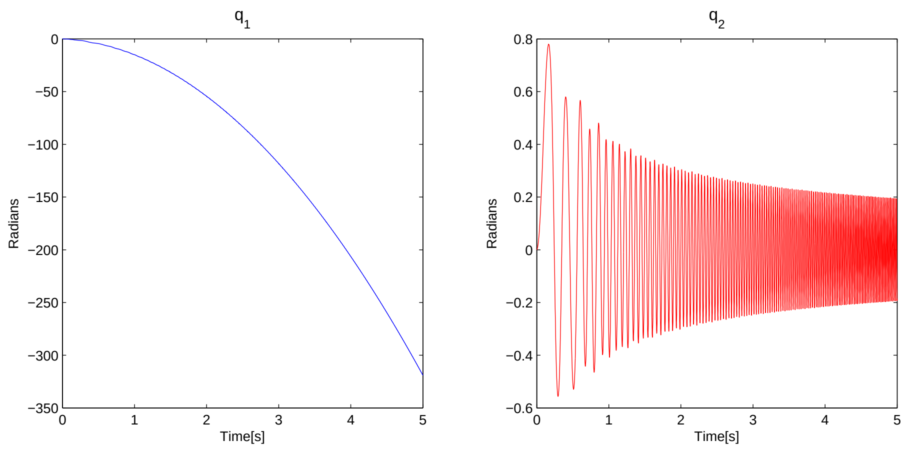

Figure 6.2: Open-loop response, simulation 1

stable almost instantly.

The second simulation has initial conditions $q_1 = \pi - 0.01, \quad q_2 = 0, \quad \dot{q} = 0$ and desired position be $q_1 = \pi, \quad q_2 = 0$ with $\dot{q} = 0$. That means the desired position is only a rotation of 0.01 radians away from the initial position in q_1. The control torque is the same as in the first simulation, and the response is shown in Figure 6.3. The plots show that the system actually manages to stay in vicinity to the desired joint variables for a short time, but it collapses and becomes unstable pretty fast.

Double pendulum comparison without external forces

Another interesting open loop simulation can be done by letting the equilibrium point $q_1 = -\pi/2, \quad q_2 = 0$ with $\dot{q} = 0$ be the desired joint variables and derivatives.

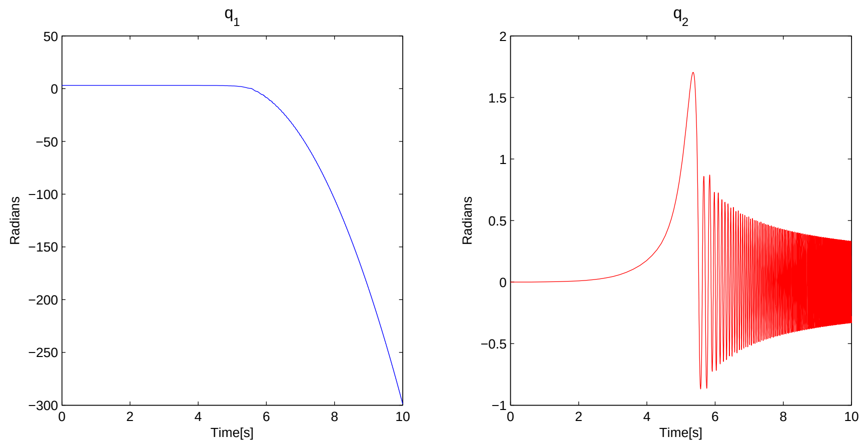

Figure 6.3: Open-loop response, simulation 2

Substituting in the dynamic equations gives

$$\tau_1 = \tau_2 = 0 \tag{6.29}$$

which means the system now can be interpreted as a double pendulum. This kind of system has been studied a lot, and is well known to be a chaotic system as demonstrated in [10] and [3]. A chaotic system is sensitive to initial conditions, meaning that infinitely small changes to the initial conditions in an otherwise perfect experiment will produce widely different results. It is not possible to predict the results, and its a complex question to determine when any of the links will flip (if possible). This is shown below where Figure 6.4 shows the response when the pendulum is released from initial conditions $q = 0, \quad \dot{q} = 0$, and Figure 6.5 shows the response when released from $q_1 = 0.000001, \quad q_2 = 0, \quad \dot{q} = 0$.

A closer look at the response when releasing from $q = 0$ gives even more insight. Figure 6.6 shows clearly the effect behind the law of action and reaction. Link 1 swings back and forth making q_1 oscillate, but the exact motion depends strongly on the motion of link 2. Note that for this experiment to be realistic there has to be a small offset in link 1, to make it possible for link 2 to flip.

Since joint friction and air resistance are not included in the model, there is no dissipation of energy, and the internal energy of the system (sum of kinetic and potential energy) is constant during motion. Figure 6.8 shows the kinetic energy, the potential energy, and the total energy of the simulation in Figure 6.4. By definition the internal energy is initially zero, and as expected, the graph shows it stays constantly zero for all time. This simulation concludes that the behavior of the manipulator is satisfying, and the model seems to be correct.

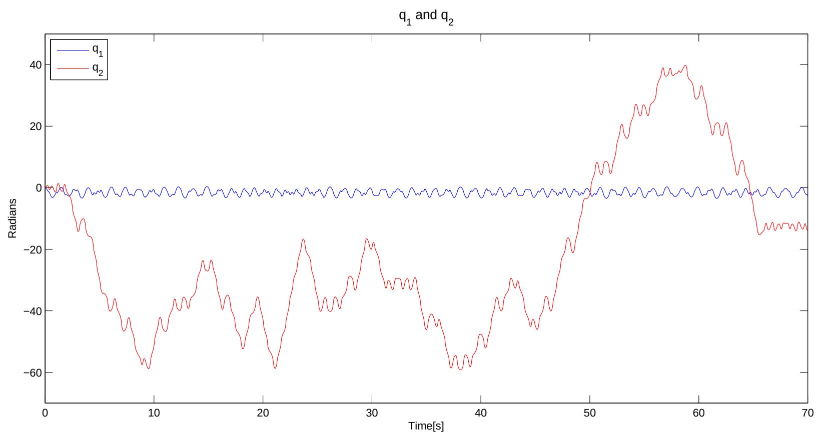

Figure 6.4: Pendulum released from $q = 0$

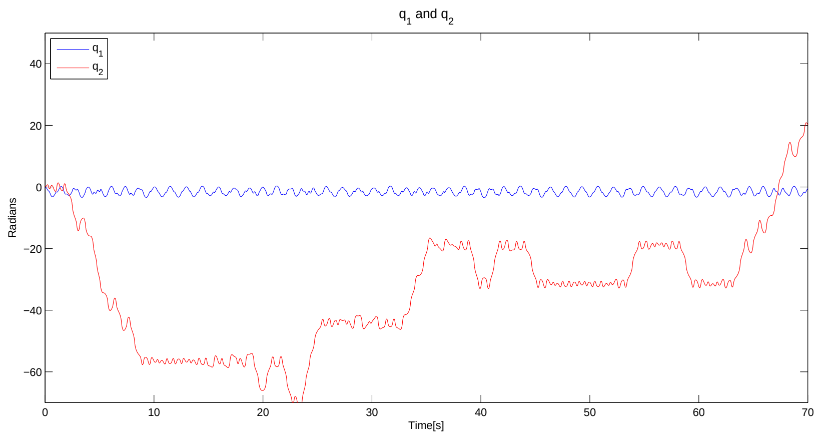

Figure 6.5: Pendulum released from $q_1 = 0.000001, \quad q_2 = 0$

6.2.4 Closed loop

The system is studied in closed loop by including a feedback controller in the system. Controllers can take one or more of three standard control elements[5], which can briefly be summarized as

- **P - proportional term**: The input is proportional to the error between the reference of the controlled states and the current output. K_p is the proportional gain.

[1]A complete description can be found in most books about control theory, e.g. [5]

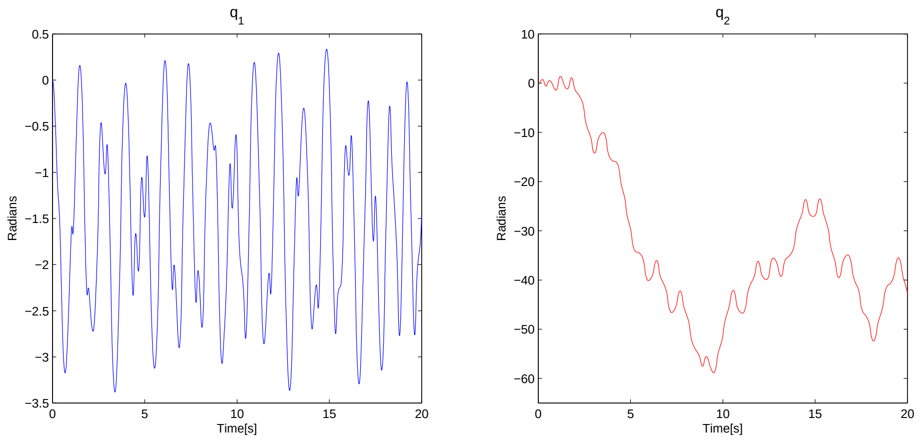

Figure 6.6: Joint movements the first 20 seconds

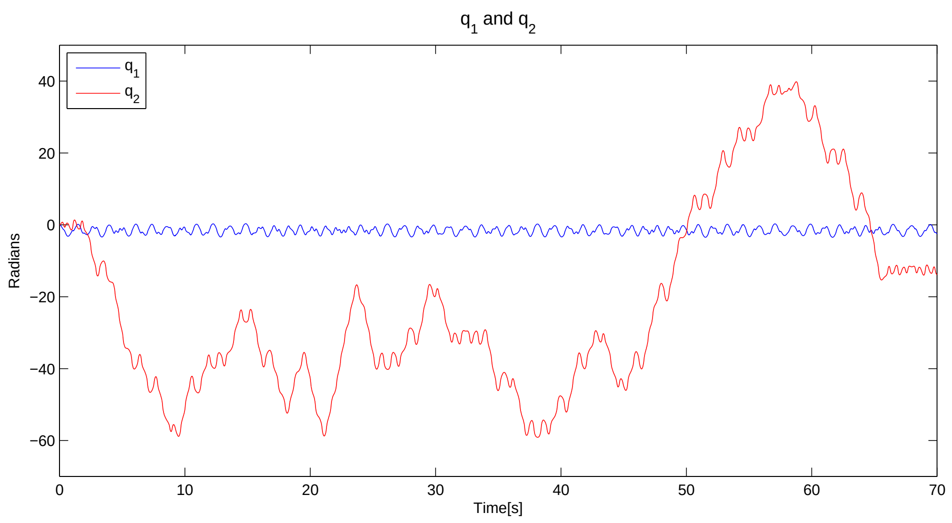

Figure 6.7: Joint movements the first 70 seconds

- **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.

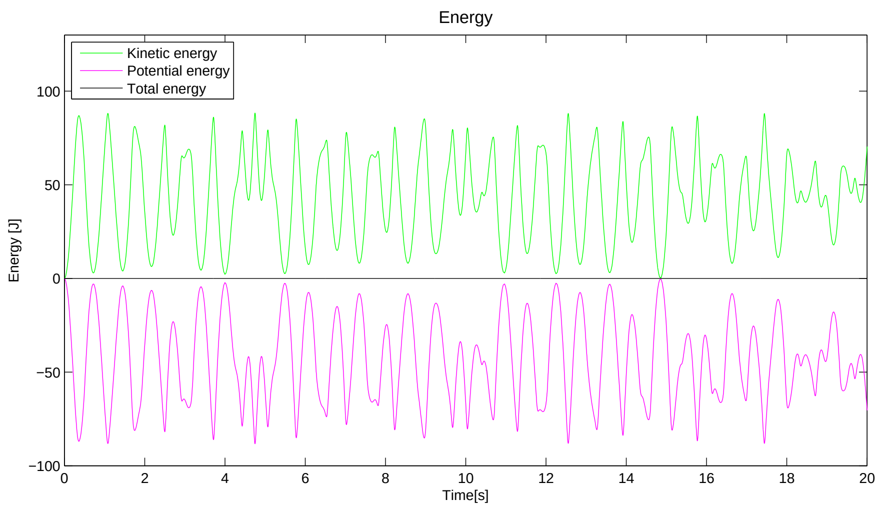

Figure 6.8: The energy in the system

Double pendulum comparison with damping

Damping is included in the model to simulate the how friction will influence the behavior. The input is now given by

$$u = -K_d\dot{q} \tag{6.30}$$

and the complete model becomes

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = -K_d\dot{q} \tag{6.31}$$

The simulink model is shown in Figure 6.9. The initial conditions are $q = 0, \dot{q} = 0$

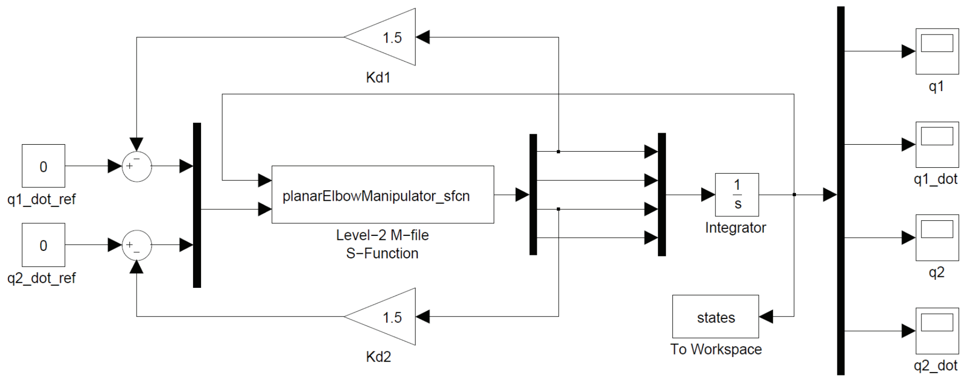

Figure 6.9: Simulink model with damping

and the derivative gain is selected to be $K_d = \begin{bmatrix} 1.5 & 1.5 \end{bmatrix}^T$. $q1$ and $q2$ are oscillating and converges to the equilibrium $q = \begin{bmatrix} -\frac{\pi}{2} & 0 \end{bmatrix}^T$ in about 25 seconds, and the internal energy is not constant anymore because of the damping. Responses are shown in Figure 6.10. The kinetic energy decreases as the oscillations decrease, such that the total energy ends up being equal to the potential energy in the equilibrium point.

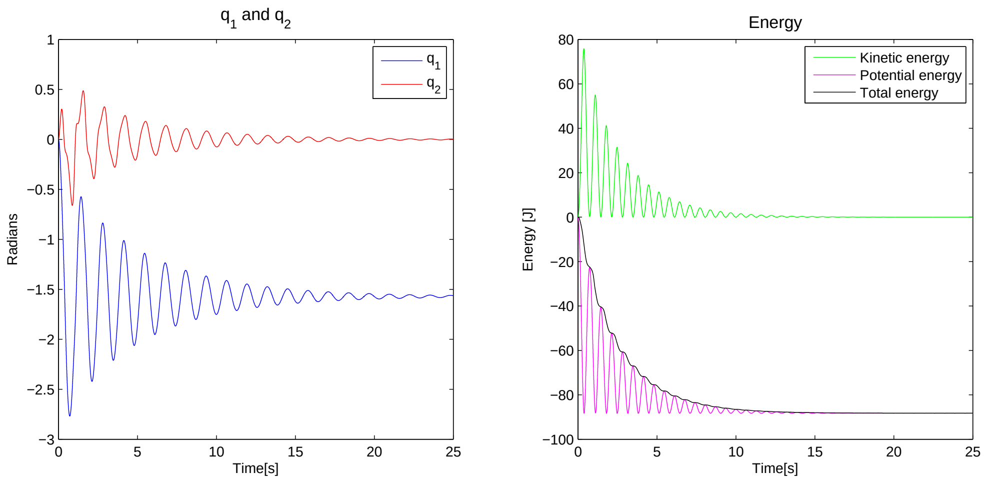

Figure 6.10: q_1, q_2 and the energy in a damped double pendulum

6.2.5 PD-controller

The challenge in control of a manipulator is to design a controller with a quick response and high accuracy, such that the manipulator is able to follow trajectories in a desired way. In open loop it is not possible to control the planar elbow manipulator as shown in section 6.2.3. The simulations have also indicated that the system has intense oscillations without friction, thus a PD-controller seems to be a good choice for control. In addition, the gravity vector $g(q)$ is added to the input since it can be assumed that gravity is constant and known. This simplification is done by removing $g(q)$ directly in the dynamic model to increase efficiency of the simulations in Simulink.

With this controller the input becomes

$$u = -K_p(q_{ref} - q) - K_d\dot{q} = -K_p\tilde{q} - K_d\dot{q} \tag{6.32}$$

where $\tilde{q}$ is the error between the joint references and the actual joint variables. The complete system model is given by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = -K_p\tilde{q} - K_d\dot{q} \tag{6.33}$$

and Figure 6.11 shows the Simulink model. Tuning the regulators are done by

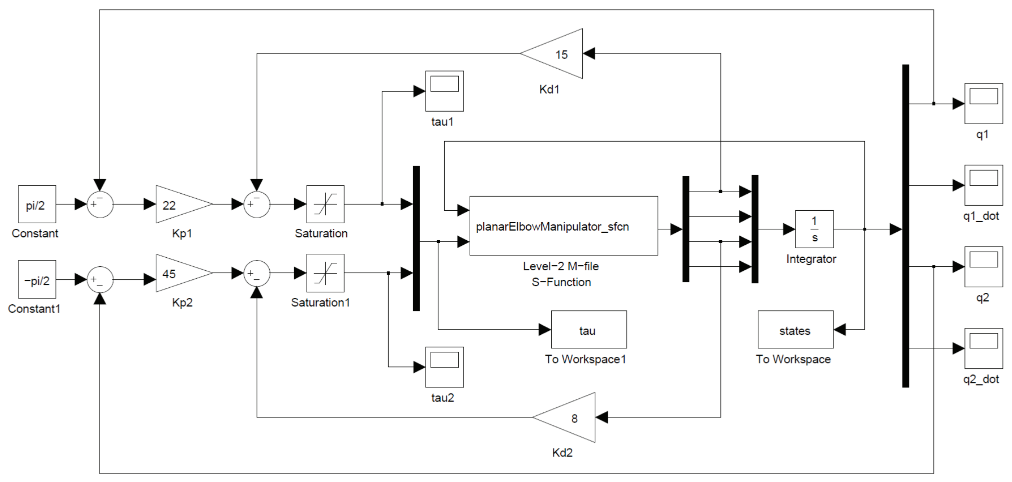

Figure 6.11: Simulink model with PD-regulator and input saturation

a trial-and-error approach, and several key points are taken into account. First, it is not desirable for a robot manipulator to overshoot a lot when moving to a reference position, and this can be avoided by increasing the derivate gain K_d. The drawbacks that follow is that the response will be slower, and the energy consumption increases. Secondly, the user of a robot will most likely require high accuracy which means that steady-state error has to be eliminated. This is usually done by including the integral term in the controller, but it will be proved in section 6.4 that the PD-controller alone achieves asymptotic tracking of the joint variables, thus no steady-state error.

The propertional gain K_p acts directly on the error between the reference and the measured state. Consequently, the input torque will be strongly dependent on K_p, such that a high gain results in a faster response. In section 6.2.2, the maximum torque available for each joint has been computed to be 39.4 Nm. Therefore, saturation at ± 39.4 has been added at both inputs. A well known problem with saturation is that the integral term in a controller continues to increase even after the input is saturated, so-called wind-up. A PD-controller does not include the integral term, thus we can add saturation without expecting any unwanted behavior. Rate limiters are not added, because it is assumed that electric motors create their electric fields very fast.

Five simulations are done to show the responses for different reference values, and the same controller parameters are used for all simulations. It was found that the parameters

$$K_p = \begin{bmatrix}22 & 45\end{bmatrix}^T \tag{6.34}$$

$$K_d = \begin{bmatrix}15 & 8\end{bmatrix}^T \tag{6.35}$$

were giving a satisfying response overall. Marginal overshooting and no steady-state error can be observed in the simulations, and all steady states are reached in 3-4 seconds.

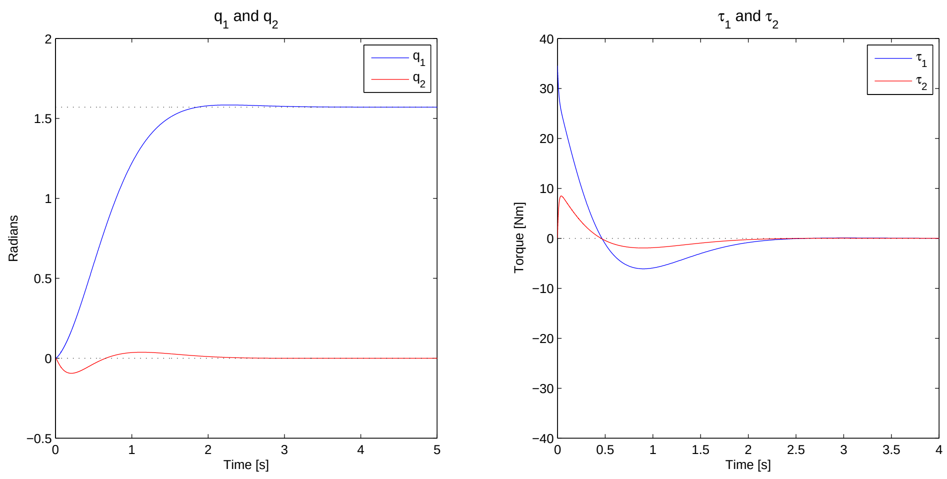

Figure 6.12: Response with PD-controller, simulation 1

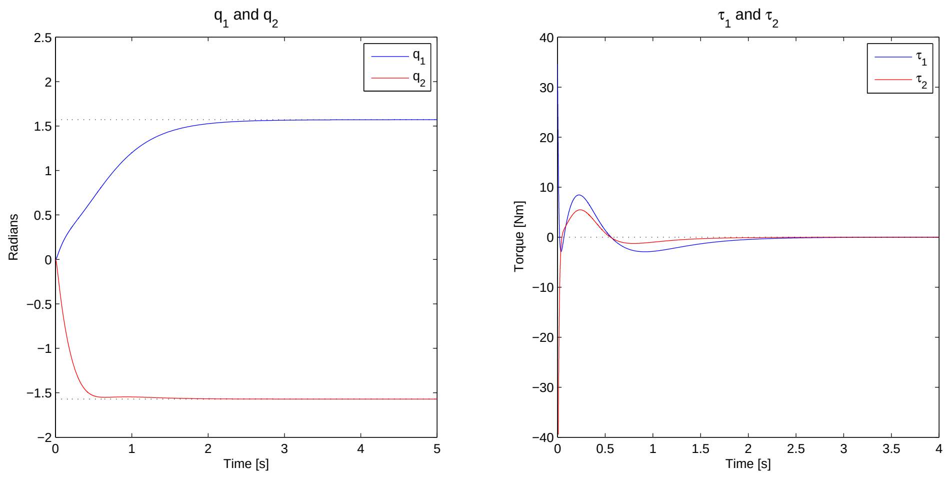

Figure 6.13: Response with PD-controller, simulation 2

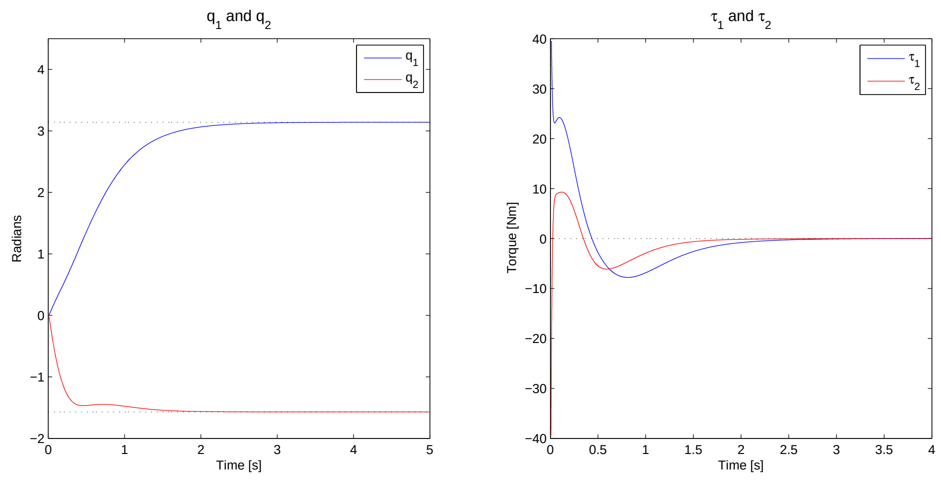

Figure 6.14: Response with PD-controller, simulation 3

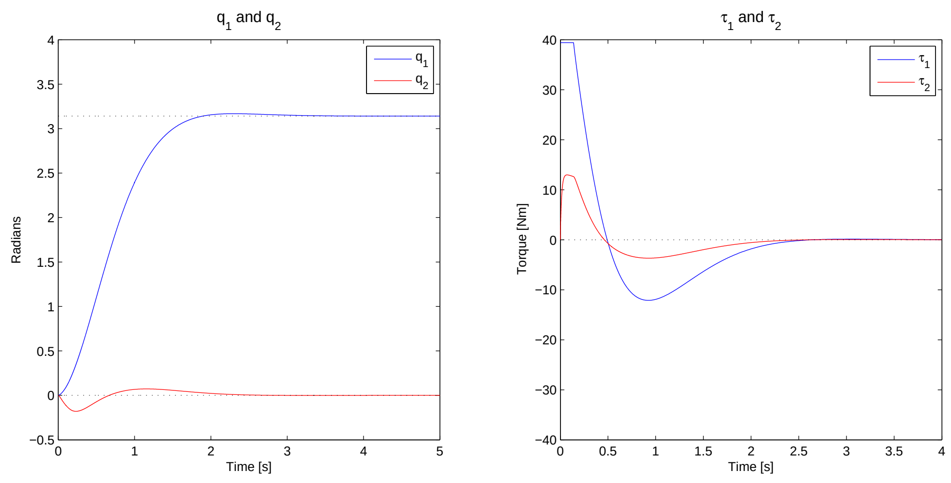

Figure 6.15: Response with PD-controller, simulation 4

6.3 IRB 140

6.3.1 Reduced system

By defining

$$x_1 = q_1, \qquad x_2 = \dot{x}_1 = \dot{q}_1 \tag{6.36}$$
$$x_3 = q_2, \qquad x_4 = \dot{x}_3 = \dot{q}_2 \tag{6.37}$$
$$x_5 = q_3, \qquad x_6 = \dot{x}_3 = \dot{q}_3 \tag{6.38}$$

the system becomes

$$\dot{x}_1 = x_2 \tag{6.39}$$
$$\dot{x}_2 = f_2(x,u) \tag{6.40}$$
$$\dot{x}_3 = x_4 \tag{6.41}$$
$$\dot{x}_4 = f_4(x,u) \tag{6.42}$$
$$\dot{x}_5 = x_6 \tag{6.43}$$
$$\dot{x}_6 = f_6(x,u) \tag{6.44}$$

where $f_2(x,u)$, $f_4(x,u)$ and $f_6(x,u)$ are very big expressions (See appendix B).

6.3.2 Parameters

Not many parameters of the IRB140 are given in the manual [4], out of consideration for trade secrets in the producer company ABB. However, the manual gives a few numbers that can be used to estimate the parameters needed. The complete mass is 98kg, and values of link lengths and link widths are given with varying accuracy. Estimates have been done, and the results are shown in Table 6.2.

Symbol	Description	Value
m_1	Mass of link 1	23 kg
m_2	Mass of link 2	25 kg
m_3	Mass of link 3	22 kg
–	Mass of base	28 kg
l_{1x}	Length of link 1, expressed in the x-direction of frame 1	0.352 m
l_{1y}	Offset, expressed in frame 1	0.070 m
l_2	Length of link 2, expressed in frame 2	0.360 m
l_3	Length of link 3, expressed in frame 3	0.380 m
w_1	Width of link 1	0.300 m
w_2	Width of link 2	0.125 m
w_3	Width of link 3	0.135 m

Table 6.2: Parameters in IRB 140

Inertia tensors for the links can be computed from the formulas[6]

$$I_{xx} = \iiint (y^2 + z^2)\rho(x, y, z)dx\,dy\,dz \tag{6.45}$$

$$I_{yy} = \iiint (x^2 + z^2)\rho(x, y, z)dx\,dy\,dz \tag{6.46}$$

$$I_{zz} = \iiint (x^2 + y^2)\rho(x, y, z)dx\,dy\,dz \tag{6.47}$$

where

$$\rho = \frac{m}{lw^2} \tag{6.48}$$

The integrals are solved in Maple and can be found in Appendix B.

As for the planar elbow manipulator, the motor torque has to be accounted for in the simulations. The manual [4] states that axis 1 and 2 have a maximum velocity of 200° per second, while axis 3 has a maximum velocity of 260° per second. Nominal torques for the electric motors has been given from a trustworthy source[7], and are stated as 1.9 Nm for joint 1 and 2, and 0.87 Nm for joint 3. Moreover, it is assumed again that all motors have 0.25 horsepowers. In section 6.2.2 it is already showed that the maximum input torque for joint 1 and 2 is 39.4 Nm, but it is still necessary to solve for joint 3 by the same formula.

$$\text{RPM} = \frac{5252 \cdot \text{Horsepower}}{\text{Torque}} \tag{6.49}$$

$$= \frac{5252 \cdot 0.25}{0.87} = 1509 \tag{6.50}$$

which equals 9054° per second. The gear ratio is then

$$\frac{9054}{260} = 34.82 \tag{6.51}$$

and the maximum input torque in the model becomes

$$34.82 \cdot 0.87 \text{ Nm} = 30.3 \text{ Nm} \tag{6.52}$$

The motor parameters are summarized in Table 6.3 for the sake of clarity.

[6]see [12], page 252

[7]Uwe Mettin, Reasearch Scientist in Robotics and Control Engineering. http://www.itk.ntnu.no/ansatte/Mettin_Uwe

Axis	Maximum velocity	Maximum input torque in model
1	200° per second	39.4 Nm
2	200° per second	39.4 Nm
3	260° per second	30.3 Nm

Table 6.3: Motor parameters in IRB 140

6.3.3 Open loop

To begin with, the energy in the system is studied in open loop without any input. Joint 1 and 2 are locked in position by setting $x_1 = x_2 = x_3 = x_4 = 0$, and link 3 are released from the reference $q_3 = 0$ to act like a pendulum. As for the planar elbow manipulator, it is expected that the total energy is constant for all time, since no energy is dissipated. Surprisingly, Figure 6.16 and 6.17 shows that this is not the case. At first glance it is not easy to discover where the strange behavior comes from, but by zooming closely in at q_3 it is discovered that the oscillations are not identical, and actually seems to vary in a periodic manner. The angle error seems to be less than 0.003 radians for all periods, and the plot of total energy shows that the strange behavior results in oscillations with an amplitude of about 2 J. It is not easy to understand what the cause of this error is, but it may very well be numerical error in the solver Matlab uses to solve differential equations. These solvers will not be completely accurate.

6.3.4 PD-controller

For the planar elbow manipulator it was observed that a PD-controller stabilized the system and the simulations obtained satisfying responses. Accordingly, the PD-controller should be tested for the 3-DOF manipulator as well. It is assumed that the gravity is known such that it can be removed from the model, such that model becomes.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = -K_p\tilde{q} - K_d\dot{q} \tag{6.53}$$

where $\tilde{q}$ is the error between the joint references and the actual joint variables. The Simulink model is shown in Figure 6.18. In the simulations, saturation are added according to Table 6.3. The gains are selected by a trial-and-error approach, and a set of satisfying gains was found as

$$K_p = \begin{bmatrix}25 & 18 & 100\end{bmatrix}^T \tag{6.54}$$

$$K_d = \begin{bmatrix}27 & 18 & 20\end{bmatrix}^T \tag{6.55}$$

Figures 6.19, 6.20 and 6.21 shows three different simulations where all initial conditions are $q = \dot{q} = 0$. All responses converge to the reference in about 5 seconds with marginal overshooting and no steady-state error.

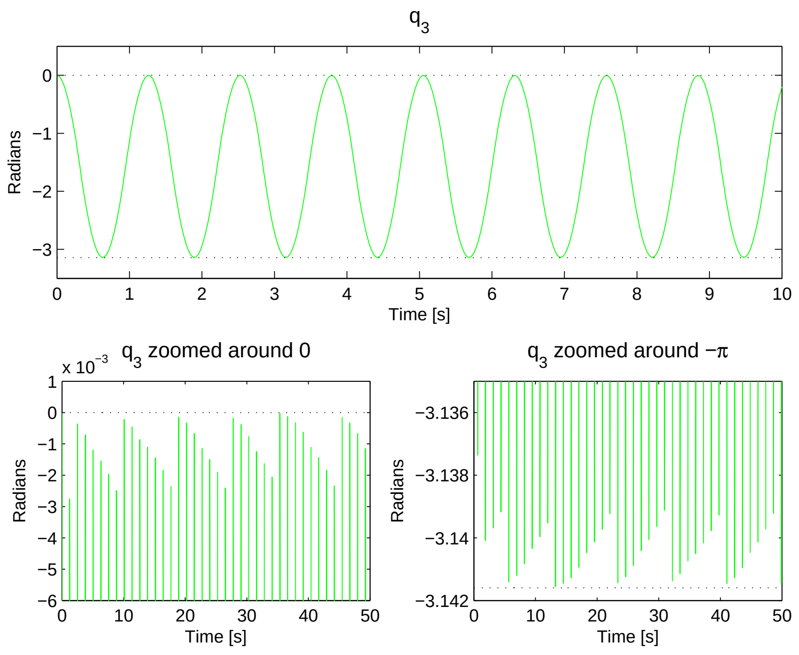

Figure 6.16: Oscillations in q_3 in open loop

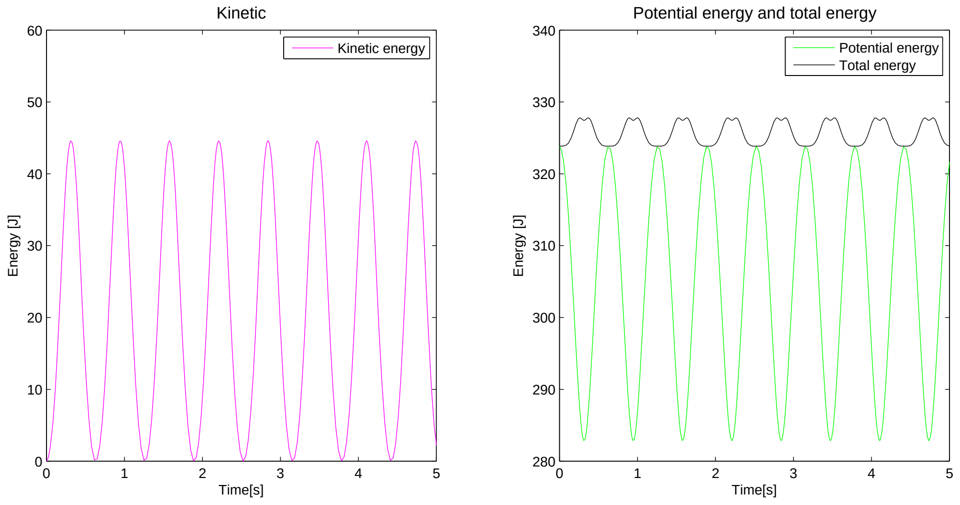

Figure 6.17: Energy in open loop

6.4 PD Control and Lyapunov stability

The simulations of both the planar elbow manipulator and the IRB 140 have been done without the use of integral term in the controller, because steady-state

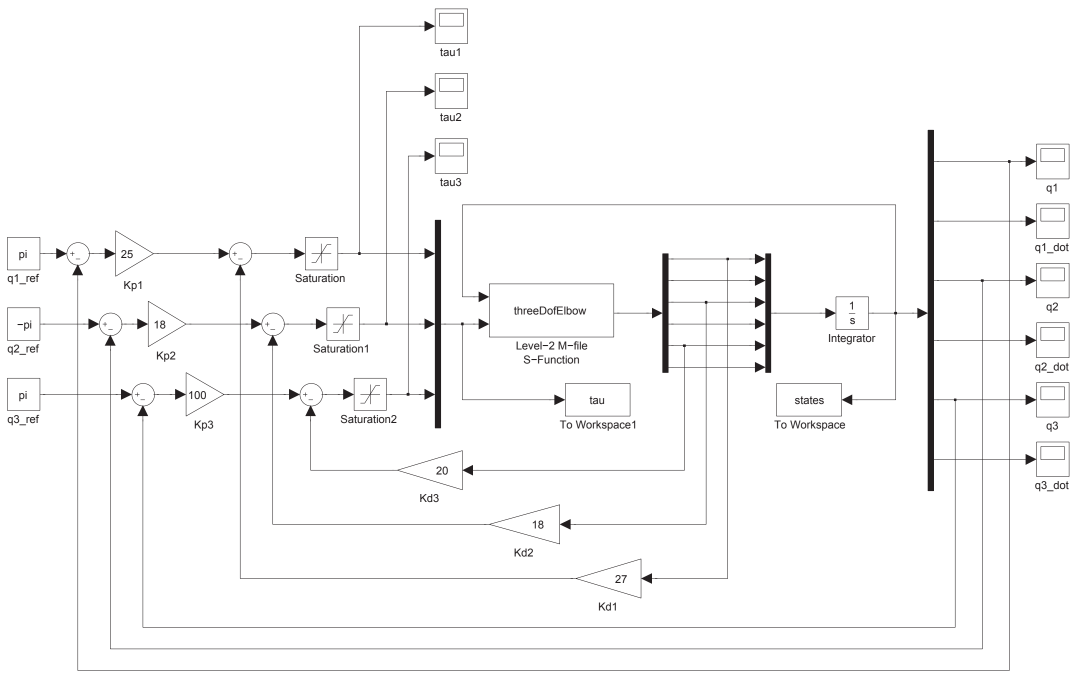

Figure 6.18: Simulink model with PD-controller and saturation

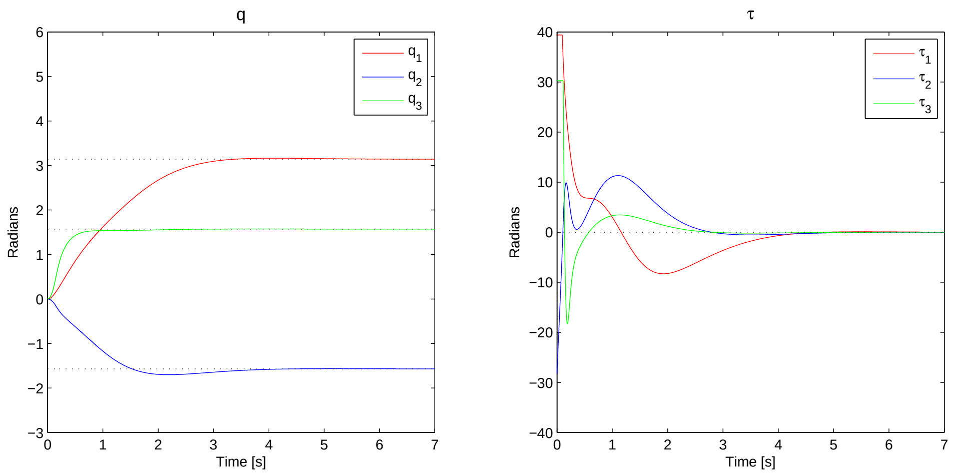

Figure 6.19: Response with PD-controller, simulation 1

error was never identified as a problem. It is a remarkable fact that the simple PD scheme for setpoint control works in the general case of a system model in the form of Equation (3.1). When $g(q)$ is zero, the PD control even achieves asymptotic

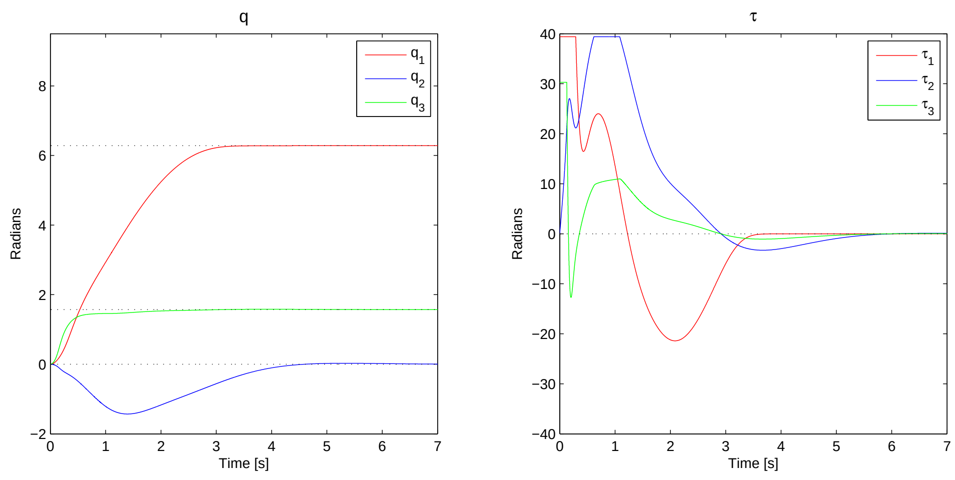

Figure 6.20: Response with PD-controller, simulation 2

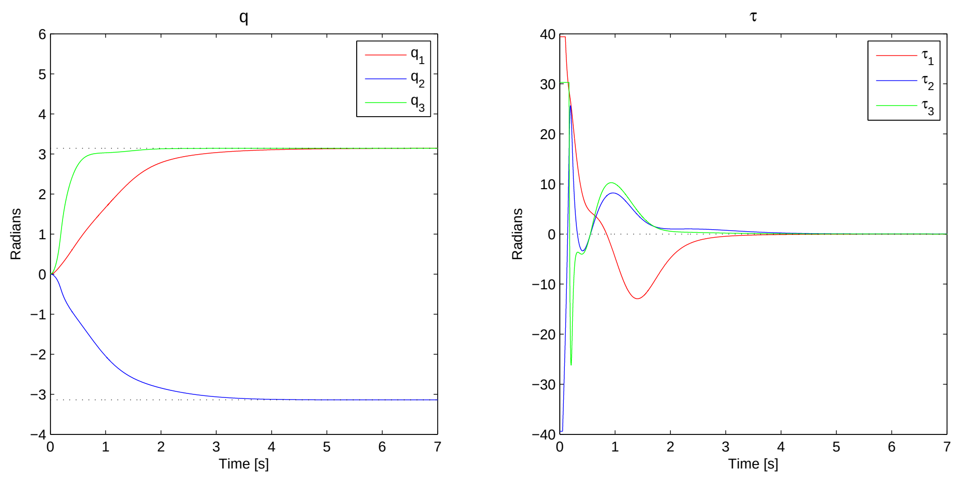

Figure 6.21: Response with PD-controller, simulation 3

tracking of the state variables. This can be proved in a Lyapunov stability analysis, as shown in [12] and [8]. This proof is of such importance and relevance to this report that it will be restated in this section.

Referring to [12], the general system model form in Equation (3.1) can be reformulated as

$$D(q)\ddot{q} + C(q,\dot{q}))\dot{q} + B\dot{q} + g(q) = u \tag{6.56}$$

where now $D(q) = M(q) + J$, where J is a diagonal matrix consisting of some actuator relations.

Consider the Lyapunov function candidate

$$V = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}\tilde{q}^T K_p \tilde{q} \tag{6.57}$$

which expresses the sum of the kinetic energy and the proportional feedback in the system. As mentioned in previous sections, $\tilde{q} = q - q_{ref}$. For a 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 points in q_{ref}. Thus, V is a positive function except in the equilibrium point $q = q_{ref}$ 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 point.

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

$$\dot{V} = \dot{q}^T M(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{M}(q)\dot{q} + \dot{q}^T K_p \tilde{q} \tag{6.58}$$

Solving for $M(q)\ddot{q}$ in Equation with $g(q) = 0$ and substituting into the equation above yields

$$\begin{aligned}
\dot{V} &= \dot{q}^T (u - C(q,\dot{q})\dot{q}) + \frac{1}{2}\dot{q}^T \dot{M}(q)\dot{q} + \dot{q}^T K_p \tilde{q} && (6.59)\\
&= \dot{q}^T (u + K_p \tilde{q}) + \frac{1}{2}\dot{q}^T (\dot{M}(q) - 2C(q,\dot{q}))\dot{q} && (6.60)\\
&= \dot{q}^T (u + K_p \tilde{q}) && (6.61)
\end{aligned}$$

where $\dot{M}(q) - 2C(q,\dot{q})$ is skew symmetric, and Equation (2.9) then gives $\dot{q}^T(\dot{M}(q) - 2C(q,\dot{q}))\dot{q} = 0$. Substituting for the control law, the above yields

$$\dot{V} = \dot{q}^T K_d \dot{q} \leq 0 \tag{6.62}$$

The above analysis shows that V is decreasing as long as $\dot{q}$ is not zero. To prove that the manipulator can not reach a position where $\dot{q} = 0$ outside q_d, La Salle's theorem can be used (see [12], page 291-292).

Chapter 7

The National Science Week in Norway

The National Science Week (norwegian: Forskningsdagene) was held in Norway in September 2010. On the website[1]the event is described like this:

> *The National Science Week in Norway (Forskningsdagene) is a nationwide event held every year to make science and research available to the public. Research and knowledge institutions throughout Norway have an opportunity to participate and provide the general public with new insight into what they do.*

The Department of Engineering Cybernetics at NTNU participated in the event in Trondheim with several science exhibits open to the public. One of them was a demonstration of the industrial robot manipulator IRB 140. I was asked if I wanted to lead this exhibit, since I already knew a lot about the robot through this project. I thought this could be a great opportunity to reach out to the public and show some of the things that cybernetics is all about, so I gladly accepted the invitation.

The target group of the event was mainly children and youth. Therefore, we wanted show something funny with the robot to attract people. We had a lot of balloons that we attached to the wall, and on the very tip of the robot arm we attached a needle. Then one could use the handheld control unit of the robot to actuate the joints, trying to hit the balloons. There was a big rush of people all day, and it was easy to see that the children loved it. Some people were also more interested and asked questions about the dynamics and the opportunities of such an industrial robot. I believe that this day contributed a lot to the public, and hopefully more people will see that technological solutions are fun and important in the future.

[1]http://www.forskningsdagene.no

Chapter 8

Future Work

There are several things which can be improved in this project. First, it is not to get away from that one has to make simplifications when modeling robot manipulators. In this project, all links have been modeled as rectangular prisms with uniform mass density, and thus the mass center in the geometric center of the link. This is without a doubt a big simplification to the inertia tensors. Looking at the IRB 140 manipulator for example, where the first three links have very irregular shapes, one understands that computing accurate inertia tensors will be a real challenge. A task for future work can be to estimate the mass centers and inertia tensors of the IRB 140. In addition, the manipulator has a spherical wrist at the end of link 3 that is not modeled in this project. Calculating the complete 6-DOF dynamic model with good estimates of unknown parameters sounds like a good challenge.

In addition to the dynamic model, it is a great task to compute the kinematics as well. When both the dynamics and the inverse kinematics are known, the manipulator can not only be controlled to desired joint variables, but it is possible to control the end effector to desired positions. That is a good start for studying trajectory planning and collision avoidance.

Most likely, the PD-controllers in this report can be tuned with better performance, and more challenging experiments can be simulated.

Chapter 9

Conclusion

This project has been about dynamic modeling of robot manipulators. It has been shown that the Newton-Euler formulation is a recursive approach, which has its great advantages. For the IRB 140 manipulator, the frames were defined in a specific manner resulting in mathematical simplifications. Simulations has showed that PD-controllers achieves asymptotic tracking of the joint variables, and the proof for this can be proved by a Lyapunov stability analysis.

Energy simulations of the IRB 140 manipulator showed that when the system was simulated without friction, the energy was not conserved as it should be. Matlab solves these differential equations numerically, and it is believed that the numerical error can be the reason for dissipation in energy. At least the setpoint tracking worked great in every simulation, such that the model seems to be correct. Moreover, the energy equations are quite big and may influence the accuracy of the solver.

Chapter 10

Lab exercise

Kybernetikk Introduksjon - Robotlab

Herman Høifødt

December 21, 2010

1 Sikkerhet

Roboten er i stand til å gjøre hurtige og uventede bevegelser med stor kraft. Derfor er det viktig at dere tar hensyn til noen enkle sikkerhetsregler:

- Ingen skal være innenfor robotens arbeidsområde når systemet er skrudd på. Hvis noen må bevege seg innenfor skal nødstopp være aktivert. Det er to nødstopp-knapper; en på styrekontrollen, og en på styreskapet.
- Pass fingrene når dere holder på med griperedskapet til roboten. Det er fort gjort å klemme fingrene dersom griperen aktiveres mens noen fikler med den.

Alt om sikkerhet står i kapittel 1 i FlexPendant manualen.

2 Introduksjon

I denne laben skal vi jobbe med en industrirobot som er produsert av ABB og heter IRB 140-6/0.8. I industrien brukes en slik robot for eksempel til lakkering og sveising, og kan utføre slike jobber langt raskere og mer nøyaktig enn et menneske. Roboten er veldig fleksibel og kan festes både til gulv, vegg og tak.

Denne laboppgaven begynner med en teoridel der dere blir kjent med en del viktige begreper innenfor robotikken. Det er ikke ment at dere skal bruke mye tid på denne delen, men ved å bruke noen minutter på å lese gjennom vil dere forstå mer om hva som faktisk skjer i de praktiske oppgavene. Det er også ment som en motivasjon for senere årskurs. Etter teoridelen kommer en systembeskrivelse, og derretter selve oppgavene som avsluttes med en konkurranseoppgave der labgruppen som løser oppgaven på kortest tid vinner.

De praktiske oppgavene går ut på å programmere roboten, men dere skal ikke implementere noe reguleringssystem. Reguleringen ligger inne fra før av, og parameterene er låst av ABB i henhold til bedriftshemmeligheter.

3 Teoridel

En robotmanipulator består av lenker (eng: links) og ledd (eng: joints), der det hovedsaklig er to typer ledd som brukes. Rotasjonsledd (eng: revolute joints) er som en hengsel som fører til en rotasjon mellom to lenker, mens prismatiske ledd (eng: prismatic joints) er som et teleskop som fører til en lineær bevegelse mellom to lenker. Når vi skal tegne en robotmanipulator bruker vi enkle symbolske respresentasjoner for ledd, slik som vist i figur 1.

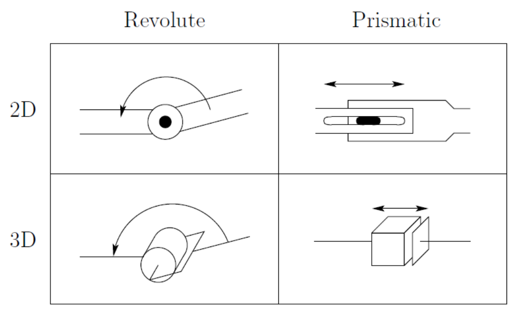

Figure 1: Symbolsk representasjon av ledd

En konfigurasjon er en fullstendig oversikt over posisjonen til ethvert punkt i roboten, eller med andre ord en beskrivelse av robotens nøyaktige stilling. Summen av alle mulige konfigurasjoner kalles for robotens konfigurasjonsområde, som da vil begrenses blant annet av robotens fysiske form og dens omgivelser. Vi ønsker at enhver mulig konfigurasjon av roboten skal kunne beskrives med minst mulig antall variabler. Først må vi bestemme oss for en bestemt konfigurasjon som skal være referansekonfigurasjonen. La så θ beskrive vinkelen i et rotasjonsledd og d beskrive forskyvningen i et prismatisk ledd, der disse størrelsene er i forhold til referansen. Så samler vi disse variablene for alle leddene og putter dem i en vektor q. Da kan vi si at roboten er i konfigurasjon q, der $q_i = \theta_i$ for rotasjonsledd og $q_i = d_i$ for prismatiske ledd, der $i = 1, ..., n$ og n er antall ledd. Vi sier også at robotmanipulatoren har n frihetsgrader, slik at altså frihetsgradene rett og slett bestemmes av antall ledd.

De fleste robotmanipulatorer som brukes i industrien har et verktøy i enden av ytterste lenke (f.eks. en griper) som gjør en form for arbeid. Når roboten arbeider vil den bevege seg slik at q endres hele tiden. Et av problemene vi ønsker å se på er hvordan man kan beskrive disse bevegelsene matematisk uten å ta hensyn til kreftene og momentene som roboten påvirkes av. Dette kalles for kinematikkproblemet og kan videre deles inn i foroverkinematikk og inverskinematikk.

Foroverkinematikk vil si å beregne posisjonen og orienteringen til enden av ytterste ledd gitt q vektoren, og vi bruker da verdiene i q til å regne oss utover lenke for lenke helt til vi har kommet til enden.

La oss se på et enkelt lite eksempel på hvordan dette fungerer. Figur 2 viser en albumanipulator i planet (eng: planar elbow manipulator) med to lenker og to rotasjonsledd. Lengden av første lenke er a_1 og lengden av andre lenke er a_2, mens θ_1 og θ_2 er vinklene i de to leddene. Koodinatsystem 0 har origo i basen til roboten der vi har første ledd, og er et fast plassert koordinatsystem uavhengig av robotens bevegelse. Koordinatsystemene 1 og 2 derimot har origo henholdsvis i enden av hver sin lenke, og de er festet til denne lenken slik at de beveger seg deretter. z-aksen til alle koordinatsystemene går vinkelrett ut av papiret. Vi ser nå at referansekonfigurasjonen er $\theta_1 = \theta_2 = 0$, altså når hele robotarmen ligger langs x_0-aksen, og legg merke til at vinkel θ_i er definert i forhold til koordinatsystem $i - 1$. Ved hjelp av grunnleggende trigonometri kan vi nå sette opp likningene for posisjon mellom koordinatsystemene, og for koordinatsystem 2 relativt koordinatsystem 0 får vi

$$x = a_1 cos(\theta_1) + a_2 cos(\theta_1 + \theta_2) \tag{3.1}$$

$$y = a_1 sin(\theta_1) + a_2 sin(\theta_1 + \theta_2) \tag{3.2}$$

$$z = 0 \tag{3.3}$$

Nå det gjelder orienteringen ser vi ved hjelp av høyrehåndsregelen at vi må rotere $\theta_1 + \theta_2$ radianer rundt z-aksen. Dermed har vi løst foroverkinematikken for en albumanipulator i planet!

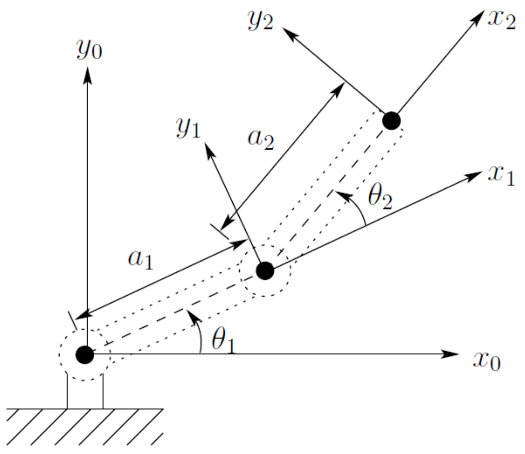

Figure 2: Albumanipulator i planet

Inverskinematikk vil si det motsatte av foroverkinematikk, altså å beregne q-vektoren gitt posisjonen til enden av ytterste lenke. Dette er vanligvis mer utfordrende enn å beregne foroverkinematikken fordi det kan finnes flere løsninger, eller kanskje ingen løsning i det hele tatt. For de ekstra interesserte kan dere prøve å regne ut løsningen på problemet for eksemplet med albumanipulatoren i planet, men dette er ikke en del av denne laben. Løsningen er

$$\theta_2 = tan^{-1}\frac{\pm\sqrt{1-D^2}}{D} \tag{3.4}$$

$$\theta_1 = tan^{-1}\left(\frac{y}{x}\right) - tan^{-1}\left(\frac{a_2 sin\theta_2}{a_1 + a_2 cos\theta_2}\right) \tag{3.5}$$

hvor D kommer fra cosinussetningen og er

$$D = cos\theta_2 = \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2} \tag{3.6}$$

Legg merke til at θ_2 ikke er avhengig av θ_1, men bare av posisjonen (x, y) som er gitt, mens θ_1 er avhengig av både θ_2 og posisjonen. Dette er pågrunn av at vi nå regner oss bakover i robotarmen.

Dette er en lite komplisert manipulator, men for en robotmanipulator med mange lenker som jobber i tre dimensjoner blir de matematiske likningene store. Dere skal derfor ikke forsøke å sette opp disse likningene for IRB 140-6/0.8, men hvordan dette gjøres vil dere lære i faget *Modellering og regulering av roboter* som er et valgfag i 4.klasse. Der lærer dere at det finnes en standard innenfor robotikken som kalles Denavit-Hartenberg konvensjonen (DH-konvensjonen), som innebærer regler for hvordan koordinatsystemer skal defineres og relateres til hverandre for å gjøre utregningene enklest mulig. Rotasjonsmatriser benyttes for å uttrykke et punkt eller en vektor i forskjellige koordinatsystemer. Vi skal ikke gå nærmere inn på dette her, da det inngår matriseregning som dere først lærer i 2.klasse i *Matematikk 3*.

Som nevnt i begynnelsen tar man ikke hensyn til kreftene og momentene som roboten påvirkes av i kinematikkproblemet, og da finnes det heller ingen pådrag som kan reguleres. Som kybernetikere ønsker vi derfor å lage en dynamisk modell av robotmanipulatoren som beskriver forholdet mellom kraft og bevegelse. Man kan for eksempel ta utgangspunkt i Newton's 2.lov og energilikninger og ender opp med en modell av differensiallikninger på formen

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \tag{3.7}$$

der de forskjellige symbolene beskriver

q	:	Vektor som beskriver leddvariablene
u	:	Vektor som beskriver pådragene
M	:	Matrise som beskriver robotens treghetsmatrise
C	:	Matrise som beskriver Coriolis -og sentrifugaleffekter
g	:	Vektor som beskriver gravitasjonskreftene

En slik modell er viktig ved design av roboter fordi vi nå kan inkludere en regulator i u, og vi får muligheten til å utføre simuleringer av systemet.

4 Systembeskrivelse

Det er tre enheter dere trenger å vite om for å gjennomføre laboppgaven, og de er

1. Robotmanipulatoren
2. Styreskapet
3. FlexPendant

4.1 Robotmanipulatoren

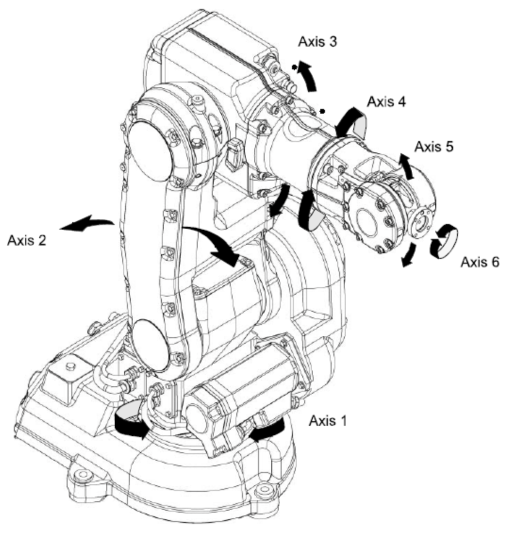

Figure 3: Robotmanipulatoren og de seks leddene

Roboten har seks rotasjonsledd og er i referansekonfigurasjon slik som i figur 3. Leddene styres av AC-motorer, og bevegelsene gires ned til valgt hastighet. På enden av ytterste lenke er det festet et griperedskap, og festepunktet for griperen er også dens rotasjonspunkt og kalles for verktøysenteret (eng: tool center point - TCP). Tabell 1 viser noen karakteristiske data for roboten.

Masse	98 kg
Maksimal last (masse til verktøy)	6 kg
Posisjonsnøyaktighet i TCP: Gjennomsnittlig avvik Maksimalt avvik	 0,35 mm 0,75 mm
Arbeidsområde for leddene: Ledd 1 Ledd 2 Ledd 3 Ledd 4 Ledd 5 Ledd 6	 +180° to -180° +110° to -90° +50° to -230° +200° to -200° +115° to -115° +400° to -400°

Table 1: Karakteristiske data ved roboten

4.2 Styreskapet

Figur 4 viser styreskapet, og her finnes all elektronikken som må til for å styre roboten. Dere trenger bare å bry dere om følgende:

A: Hovedbryter for å slå av og på systemet

B: Nødstopp-knapp

C: Lys som indikerer at robotmotorene er på

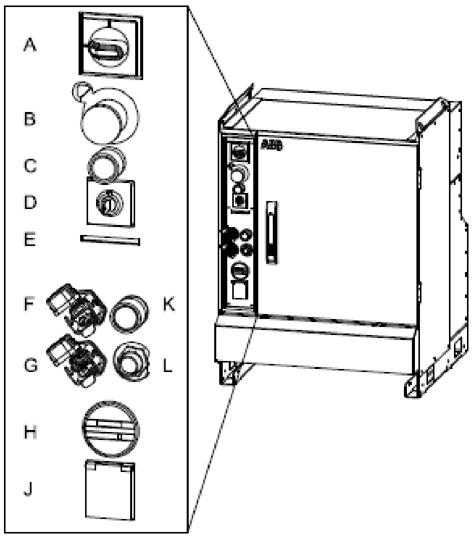

Figure 4: Styreskap

4.3 FlexPendant

FlexPendant er navnet på den håndholdte kontrollen som brukes blant annet til å lage robotprogrammer og styre roboten. Figur 5 viser en oversikt over hoveddelene, og figur 6 viser hvordan den bør holdes (med venstre hånd). På baksiden av FlexPendant finnes en plastikkpenn til touch-screenen.

A: Tilkobling til styreskap

B: Touch-screen

C: Nødstopp-knapp

D: Dødmannsknapp

E: Joystick

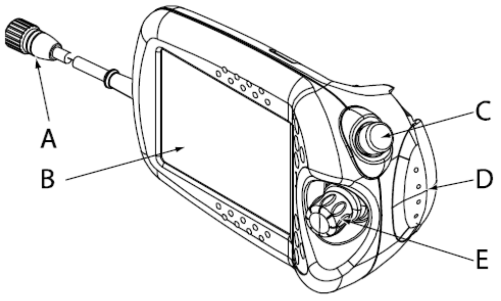

Figure 5: FlexPendant

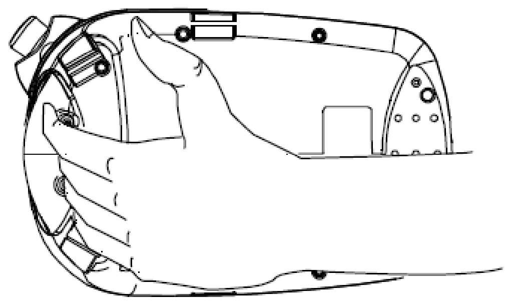

Figure 6: Slik holdes FlexPendant

5 Oppgaver

5.1 Oppstart og viktig informasjon

- Slå på hovedbryteren på styreskapet og vent til dere ser hovedvinduet med teksten "Welcome to ABB" på FlexPendant.
- Pass på at nøkkelen i styreskapet står i loddrett posisjon. Det skal da stå "Manual" til venstre i det grå statusfeltet øverst på FlexPendant.

- Hvis det står "Emergency Stop" i midten i statusfeltet er minst en nødstoppbryter aktivert. Da må dere vri bryterene med pilens retning, og etterpå trykke på knappen rett over nøkkelen på styreskapet. Det samme må dere gjøre for å deaktivere nødstoppen om dere selv får bruk for den. Teksten i statusfeltet skal da endres til "Guard Stop"

Før dere går videre skal det altså stå "Manual" og "Guard Stop" i statusfeltet.

- For å kjøre roboten må dødmannsknappen holdes inne slik at motorene slåes på (den skal bare trykkes halvveis inn). Dere gjør det riktig når "Guard Stop" endres til "Motors On". Bruk så joysticken til å kjøre roboten.
- Dere kan komme til å støte på feilmeldinger som popper opp på FlexPendant, blant annet hvis roboten kjøres ut av dens arbeidsområde. Les i feilmeldingen hva som er galt og hva som må gjøres, og trykk "Acknowledge".

5.2 Oppgave 1: Manuell kjøring med roboten

- Trykk på ROB_1 for å åpne hurtigmenyen.
- Trykk på for å velge koordinatsystem og akser.
- Trykk på << Show Details
- Velg et av fire koordinatsystemer (vi skal bare bruke to av de):
  - Trykk på for basekoordinatsystemet som er et fast montert koordinatsystem med origo i basen, og med akser som vist i figur 7.

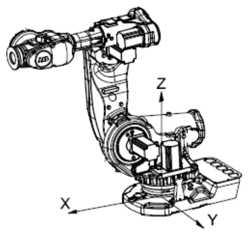

Figure 7: Basekoordinatsystemet

  - Trykk på for verktøykoordinatsystemet som er festet til valgt verktøy og flyttes og roteres deretter. Verktøyet "tool0" (se på FlexPendant) er forhåndsdefinert og har origo i TCP, altså der et verktøyet festes.
- Velg en av fire bevegelsesmoduser:

- Trykk på for å styre ledd 1, 2 og 3. Velg ledd med joysticken slik: 2 1 3
- Trykk på for å styre ledd 4, 5 og 6. Velg ledd med joysticken slik: 5 4 6
- Trykk på for lineær bevegelse, det vil at TCP beveger seg lineært i forhold til valgt koordinatsystem. Velg akse med joysticken slik: X Y Z
- Trykk på for rotasjon, det vil si at TCP roterer om aksene i valgt koordinatsystem. Velg akse med joysticken slik: X Y Z

• Studer hva som skjer i disse tilfellene, og prøv å forstå at bevegelsene stemmer i forhold til koordinatsysteme i figuren:

  - Bevegelse av aksene hver for seg (Any + /)
  - Lineær bevegelse i forhold til basekoordinatsystem (+)
  - Lineær bevegelse i forhold til verktøykoordinatsystem (+)
  - Rotasjon i forhold til basekoordinatsystem (+)
  - Rotasjon i forhold til verktøykoordinatsystem (+)

5.3 Oppgave 2: Kjør et demo-program

Dere skal nå kjøre et forhåndslaget program hvor roboten skal plukke opp en penn og tegne på et ark. Kjør først programmet uten tegnebrett for å se hvor brettet må plasseres. Så teiper dere fast et ark på tegnebrettet, plasserer det på riktig sted, og kjører programmet en gang til.

• Trykk på for å åpne hovedmenyen.

• Trykk på **Program Editor**. Her ser dere programkoden til programmet som er åpent for øyeblikket.

• Trykk på **Tasks and Programs**.

• Trykk på **File** og **Load Program**. Hvis dere får opp en melding angående lagring velger dere **Don't Save**.

- Finn programmet DEMOPROGRAM i mappen ..../HOME. Gå inn i mappen og åpne DEMOPRGRAM.pgf
- Trykk på **Debug**.
- Trykk på **PP to Main** for å sette programpekeren til første programlinje.
- Hold dødmannsknappen inne og trykk på ▶ som er nedenfor joysticken.
- Når dere er ferdige kan dere avslutte ved å trykke på ☒ øverst i høyre hjørnet.

5.4 Oppgave 3: Lag et program

I denne oppgaven skal dere lage deres eget tegne-program der roboten plukker opp pennen, tegner en figur, og setter pennen tilbake i holderen. Hvordan dette programmeres forklares nøye steg for steg slik, at dere skal være i stand til å klare den siste konkurranse-oppgaven på egen hånd.

- Trykk på ABB
- Trykk på **Program Editor**.
- Trykk på **Tasks and Programs**.
- Trykk på **File** og **New Program**. Hvis dere får opp en melding angående lagring velger dere **Don't Save**.
- Velg et navn på programmet og trykk **OK**.
- Trykk på **Tasks and Programs**.
- Sjekk under **Program Name** at det er deres program som ligger i minnet, og trykk på **Open**.

Hvis dette er gjort riktig skal den tomme programkoden bestå av fire linjer. Inne i main-metoden skal dere nå legge inn instruksjoner for hvordan roboten skal bevege seg. Instruksjonene dere skal bruke vil ha formen

$$[\text{Bevegelsesmåte}], [\text{Målkonfigurasjon}], [\text{Hastighet}], [\text{Nøyaktighet}], [\text{Verktøy}] \tag{5.1}$$

der de forskjellige parameterene er

[Bevegelsesmåte]: MoveL (lineær), MoveC (sirkulær), MoveJ (ulineært, raskeste vei)

[Målkonfigurasjon]: Konfigurasjonen som roboten skal ende opp i (markeres alltid med *).

[Hastighet]: Bevegelseshastighet som kan endres ved å dobbelklikke på den. vX gir en hastighet der TCP beveger seg med X millimeter per sekund.

[Nøyaktighet]: Lengden på svingbuen mellom denne og neste instruksjon. FINE gir ingen lengde (altså spisse hjørner), mens zX angir hvor lang den skal være der X er svingbuens lengde i millimeter (altså avrundede hjørner).

[Verktøy]: Vi skal bare holde oss til tool0 som har origo i TCP.

Mens dere er i programmeringsvinduet kan dere når som helst styre roboten manuelt ved å åpne og lukke hurtigmenyen slik som i oppgave 2. Da begynner vi å programmere:

1. Åpne griperen.
2. Styr roboten manuelt til en konfigurasjon slik at blyanten er mellom gripeklypene. Merk linja <SMT> i programmeringsvinduet.
3. Velg Add Instruction og MoveL slik at en instruksjonslinje legges til.
4. Lukk griperen.
5. Styr roboten manuelt rett opp til et punkt over penn-holderen. Dere må da styre lineært i basekoordinatsystemet langs z-aksen (vri joysticken mot klokka for å gå oppover).
6. Velg Add Instruction og MoveL.
7. Styr roboten manuelt til blyanten er noen cm rett over punkt A på figuren. Griperens orientering må være slik at blyanten står vertikalt og for å få til en nøyaktig vertikal stilling må dere endre styrehastighet i manuell modus. Trykk på SPEED og velg SLOW. Husk at dere når som helst kan endre styrehastigheten!
8. Velg Add Instruction og MoveL.
9. Styr roboten PÅ LAV STYREHASTIGHET ned til punkt A, slik at blyanten såvidt trykker på papiret. Dere må bruke lineær bevegelse i basekoordinatsystemet langs z-aksen.
10. Velg Add Instruction og MoveL.
11. Styr roboten manuelt til punkt B. Pass på styrehastigheten!
12. Velg Add Instruction og MoveL.
13. Repeter for punkt C og D.
14. Nå skal vi tilbake til punkt A, og for at nye punkt A skal være det samme som gamle punkt A skal vi kopiere målkonfigurasjonen fra instruksjonslinje X. Trykk først på Add Instruction og MoveL. Trykk så på målkonfigurasjonen (*) i linje X, og på EDIT og COPY. Derretter trykker dere på målkonfigurasjonen (*) i siste instruksjonen som dere nettopp la til og trykker på EDIT og PASTE.
15. Nå skal vi tilbake til punkt D i en sirkulær bevegelse innom punkt E. Trykk først Add Instruction og MoveC. Legg merke til at MoveC har to stjerner, der den første er passeringskonfigurasjonen og den andre er målkonfigurasjonen. Styr roboten manuelt til punkt E. Merk så den første stjerna og velg MODIFY POSITION. Kopier så målkonfigurasjonen (*) i linje X og lim inn denne på den siste stjerna.
16. Til slutt skal pennen tilbake i holderen. Legg til nødvendige instruksjoner, og avslutt med å åpne griperen.

5.5 Oppgave 4: Konkurranse - Stable klosser

I denne oppgaven skal dere programmere roboten til å stable tre klosser oppå hverandre. Siste side i oppgaveheftet viser posisjonen klossene skal ligge i før start. Riv av arket, teip det fast i arbeidsbordet og legg klossene i riktig posisjoner. Bruk det dere har lært og erfart til nå til å programmere roboten. Når dere har fullført programmeringen vil studass ta tida dere bruker på å stable klossene. Tiden gjelder fra første kloss gripes til siste kloss slippes. Dette blir en konkurranse der labgruppen med raskest tid vinner en premie!

Regler:

- Ved start må klossene ligge i gitte posisjoner, men dere kan gripe gripe klossene i ønsket rekkefølge og den endelige stabelen kan stå hvor som helst.
- Tiden gjelder fra første kloss gripes til siste kloss slippes.
- Dere har kun lov til å bruke MoveL og MoveC instruksjoner, men hastigheten og nøyaktigheten til instruksjonene velger dere selv (vmax gir maksimal lovlig hastighet til en instruksjon).

Appendix A

Errata for chapter 7.6 Newton Euler Formulation in [12]

Chapter 7.6 Newton-Euler formulation in the book [12] has some misleading definitions and errors. Most of the errors are based on index definitions that is not in accordance to later use, but there are also some other errors. The following errors have been discovered:

- Page 275, definition of R^i_{i+1} should be:
  R^i_{i+1} : 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:

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

- Page 277, Equation (7.147) should be:

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

- Page 278, Equation (7.153) should be:

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

- Page 278, Equation (7.154) should be:

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

- 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:

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

- Page 279, Equation (7.158) should be:

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

- 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:

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

- Page 279, Equation (7.162) for ω_2 should be:

$$\omega_2 = (\dot{q}_1 + \dot{q}_2)k \tag{A.8}$$

- Page 280, Equation (7.163) should be:

$$r_{0,c1} = l_{c1} i, \qquad r_{1,c1} = (l_{c1} - l_1) i, \qquad r_{0,1} = l_1 i \tag{A.9}$$

- Page 280, Equation (7.164) should be:

$$r_{1,c2} = l_{c2} i, \qquad r_{2,c2} = (l_{c2} - l_2) i, \qquad r_{1,2} = l_2 i \tag{A.10}$$

- Page 280, Equation (7.166) should be:

$$g_1 = (R_1^0)^T gj = -g \begin{bmatrix} sin(q_1) \\ cos(q_1) \\ 0 \end{bmatrix} \tag{A.11}$$

- Page 280, Equation (7.168) should be:

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

- Page 281, Equation (7.169) should be:

$$\begin{aligned} (R_2^1)^T a_{e,1} &= \begin{bmatrix} cos(q_2) & sin(q_2) \\ -sin(q_2) & cos(q_2) \end{bmatrix} \begin{bmatrix} -l_1\dot{q}_1^2 \\ l_1\ddot{q}_1 \end{bmatrix} && \text{(A.13)} \\ &= \begin{bmatrix} -l_1\dot{q}_1^2 cos(q_2) + l_1\ddot{q}_1 sin(q_2) \\ l_1\dot{q}_1^2 sin(q_2) + l_1\ddot{q}_1 cos(q_2) \end{bmatrix} && \text{(A.14)} \end{aligned}$$

- Page 281, Equation (7.170) should be:

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

- Page 281, Equation (7.171) should be:

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

- Page 281, Equation (7.174) should be:

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

- Page 282, Equation (7.175) should be:

$$f_1 = R_2^1 f_2 + m_1 a_{c,1} - m_1 g_1 \tag{A.18}$$

- Page 282, Equation (7.176) should be:

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

- 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:

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

Appendix B

Maple code

Dynamic Model: Planar elbow manipulator

```
> restart:
> with(LinearAlgebra):
>
> #Defining joint variables
> q := Vector([[q1], [q2]]):
> Dq := Vector([[Dq1], [Dq2]]):
> DDq := Vector([[DDq1], [DDq2]]):
>
> #Defining link length vectors
> r0c1 := Vector([[lc1], [0], [0]]):
> r1c2 := Vector([[lc2], [0], [0]]):
> r01 := Vector([[l1], [0], [0]]):
> r12 := Vector([[l2], [0], [0]]):
> r1c1 := r0c1 - r01:
> r2c2 := r1c2 - r12:
>
> #Rotation matrices
```

$$> R01 := \begin{bmatrix} \cos(q1) & -\sin(q1) & 0 \\ \sin(q1) & \cos(q1) & 0 \\ 0 & 0 & 1 \end{bmatrix}:$$

$$> R12 := \begin{bmatrix} \cos(q2) & -\sin(q2) & 0 \\ \sin(q2) & \cos(q2) & 0 \\ 0 & 0 & 1 \end{bmatrix}:$$

```
> R02 := combine(MatrixMatrixMultiply(R01, R12), trig):
>
> #Gravity vector in inertial frame
> g0 := Vector([[0], [-g], [0]]):
>
> #Angular velocity and angular acceleration
> ω1 := Vector([[0], [0], [Dq1]]):
> α1 := Vector([[0], [0], [DDq1]]):
> ω2 := Vector([[0], [0], [Dq1+Dq2]]):
> α2 := Vector([[0], [0], [DDq1+DDq2]]):
>
> #Forward recursion: Link 1
> ac1 := CrossProduct(α1, r0c1) + CrossProduct(ω1, CrossProduct(ω1, r0c1))
```

$$ac1 := \begin{bmatrix} -Dq1^2\, lc1 \\ DDq1\, lc1 \\ 0 \end{bmatrix} \tag{1}$$

```
> ae1 := CrossProduct(α1, r01) + CrossProduct(ω1, CrossProduct(ω1, r01))
```

$$ae1 := \begin{bmatrix} -Dq1^2\, l1 \\ DDq1\, l1 \\ 0 \end{bmatrix} \tag{2}$$

```
> g1 := MatrixVectorMultiply(Transpose(R01), g0)
```

$$g1 := \begin{bmatrix} -\sin(q1)\, g \\ -\cos(q1)\, g \\ 0 \end{bmatrix} \tag{3}$$

```
>
> #Forward recursion: Link 2
> ac2 := MatrixVectorMultiply(Transpose(R12), ae1) + CrossProduct(α2, r1c2) + CrossProduct(ω2, CrossProduct(ω2, r1c2))
```

$$ac2 := \begin{bmatrix} -\cos(q2)\, Dq1^2\, l1 + \sin(q2)\, DDq1\, l1 - (Dq1 + Dq2)^2\, lc2 \\ \sin(q2)\, Dq1^2\, l1 + \cos(q2)\, DDq1\, l1 + (DDq1 + DDq2)\, lc2 \\ 0 \end{bmatrix} \tag{4}$$

```
> g2 := MatrixVectorMultiply(Transpose(R02), g0)
```

$$g2 := \begin{bmatrix} -\sin(q1 + q2)\, g \\ -\cos(q1 + q2)\, g \\ 0 \end{bmatrix} \tag{5}$$

```
>
```

> **#Backward recursion: Link 2**

> $f2 := m2 \cdot ac2 - m2 \cdot g2$:

> $f2 := collect(combine(f2, trig), \{DDq1, DDq2, Dq1, Dq2\})$:

> $\tau 2 := CrossProduct(m2 \cdot g2 - m2 \cdot ac2, r1c2) + MatrixVectorMultiply(I2, \alpha 2)$:

> $\tau 2z := collect(combine(\tau 2[3], trig), \{DDq1, DDq2, Dq1, Dq2\})$

$$\tau 2z := lc2\, m2 \sin(q2)\, Dq1^2\, l1 + (lc2\, m2 \cos(q2)\, l1 + m2\, lc2^2 + I2)\, DDq1 + (I2 + m2\, lc2^2)\, DDq2 + lc2\, m2 \cos(q1 + q2)\, g \tag{6}$$

> **#Backward recursion: Link 1**

> $f1 := MatrixScalarMultiply(R12, f2) + m1 \cdot ac1 - m1 \cdot g1$:

> $f1 := collect(combine(f1, trig), \{DDq1, DDq2, Dq1, Dq2\})$:

> $\tau 1 := \tau 2 + CrossProduct(m1 \cdot g1, r0c1) - CrossProduct(m1 \cdot ac1, r0c1) - CrossProduct(MatrixScalarMultiply(R12, f2), r01) + MatrixVectorMultiply(I1, \alpha 1)$:

> $\tau 1z := collect(combine(\tau 1[3], trig), \{DDq1, DDq2, Dq1, Dq2\})$

$$\tau 1z := -2\, l1 \sin(q2)\, m2\, lc2\, Dq1\, Dq2 - l1 \sin(q2)\, m2\, lc2\, Dq2^2 + (2\, lc2\, m2 \cos(q2)\, l1 + I2 + m2\, lc2^2 + m2\, l1^2 + I1 + m1\, lc1^2)\, DDq1 + (lc2\, m2 \cos(q2)\, l1 + m2\, lc2^2 + I2)\, DDq2 + m2\, l1\, g \cos(q1) + lc2\, m2 \cos(q1 + q2)\, g + m1 \cos(q1)\, g\, lc1 \tag{7}$$

> **#Setting up the matrix elements**

> $m11 := (2\, lc2\, m2 \cos(q2)\, l1 + I2 + m2\, lc2^2 + m2\, l1^2 + I1 + m1\, lc1^2)$:

> $m12 := (lc2\, m2 \cos(q2)\, l1 + m2\, lc2^2 + I2)$:

> $m21 := (lc2\, m2 \cos(q2)\, l1 + m2\, lc2^2 + I2)$:

> $m22 := (I2 + m2\, lc2^2)$:

> $c11 := -2\, l1 \sin(q2)\, m2\, lc2\, Dq2$:

> $c12 := -l1 \sin(q2)\, m2\, lc2\, Dq2$:

> $c21 := lc2\, m2 \sin(q2)\, Dq1 l1$:

> $c22 := 0$:

> $g1 := m2\, l1\, g \cos(q1) + lc2\, m2 \cos(q1 + q2)\, g + m1 \cos(q1)\, g\, lc1$:

> $g2 := lc2\, m2 \cos(q1 + q2)\, g$:

>

> **#Setting up the dynamic system**

> $M0 := Matrix([[m11, m12], [m21, m22]])$

$$M0 := \begin{bmatrix} 2\, lc2\, m2 \cos(q2)\, l1 + I2 + m2\, lc2^2 + m2\, l1^2 + I1 + m1\, lc1^2 & lc2\, m2 \cos(q2)\, l1 + m2\, lc2^2 + I2 \\ lc2\, m2 \cos(q2)\, l1 + m2\, lc2^2 + I2 & I2 + m2\, lc2^2 \end{bmatrix} \tag{8}$$

> $C0 := Matrix([[c11, c12], [c21, c22]])$

$$C0 := \begin{bmatrix} -2\, l1 \sin(q2)\, m2\, lc2\, Dq2 & -l1 \sin(q2)\, m2\, lc2\, Dq2 \\ lc2\, m2 \sin(q2)\, Dq1\, l1 & 0 \end{bmatrix} \tag{9}$$

> $G0 := Vector([[g1], [g2]])$

$$G0 := \begin{bmatrix} m2\, l1\, g \cos(q1) + lc2\, m2 \cos(q1 + q2)\, g + m1 \cos(q1)\, g\, lc1 \\ lc2\, m2 \cos(q1 + q2)\, g \end{bmatrix} \tag{10}$$

> $M0inv := MatrixInverse(M0)$:

> $DynSysWithGrav := combine(simplify(MatrixVectorMultiply(M0inv, (-MatrixVectorMultiply(C0, Dq) - G0 + Vector([[u1], [u2]])))), trig)$:

> $DynSysWithoutGrav := combine(simplify(MatrixVectorMultiply(M0inv, (-MatrixVectorMultiply(C0, Dq) + Vector([[u1], [u2]])))), trig)$

$$\begin{aligned} DynSysWithoutGrav := \Big[\Big[& \big(4\, I2\, l1 \sin(q2)\, m2\, lc2\, Dq1\, Dq2 + 2\, I2\, l1 \sin(q2)\, m2\, lc2\, Dq2^2 + 2\, I2\, u1 + 4\, m2^2\, lc2^3\, l1 \sin(q2)\, Dq1\, Dq2 \\ & + 2\, m2^2\, lc2^3\, l1 \sin(q2)\, Dq2^2 + 2\, m2\, lc2^2\, u1 + lc2^2\, m2^2\, l1^2\, Dq1^2 \sin(2\, q2) - 2\, lc2\, m2 \cos(q2)\, l1\, u2 + 2\, m2^2\, lc2^3 \sin(q2)\, Dq1^2\, l1 \\ & - 2\, m2\, lc2^2\, u2 + 2\, I2\, lc2\, m2 \sin(q2)\, Dq1^2\, l1 - 2\, I2\, u2\big) \Big/ \big(2\, m2\, l1^2\, I2 + m2^2\, l1^2\, lc2^2 + 2\, I1\, I2 + 2\, I1\, m2\, lc2^2 + 2\, m1\, lc1^2\, I2 \\ & + 2\, m1\, lc1^2\, m2\, lc2^2 - m2^2\, l1^2\, lc2^2 \cos(2\, q2)\big)\Big], \\ & \Big[\big(-2\, lc2^2\, m2^2\, l1^2\, Dq1\, Dq2 \sin(2\, q2) - lc2^2\, m2^2\, l1^2\, Dq2^2 \sin(2\, q2) - 2\, lc2\, m2 \cos(q2)\, l1\, u1 - 4\, m2^2\, lc2^3\, l1 \sin(q2)\, Dq1\, Dq2 \\ & - 2\, m2^2\, lc2^3\, l1 \sin(q2)\, Dq2^2 - 2\, m2\, lc2^2\, u1 - 4\, I2\, l1 \sin(q2)\, m2\, lc2\, Dq1\, Dq2 - 2\, I2\, l1 \sin(q2)\, m2\, lc2\, Dq2^2 - 2\, I2\, u1 \\ & - 2\, lc2^2\, m2^2\, l1^2\, Dq1^2 \sin(2\, q2) + 4\, lc2\, m2 \cos(q2)\, l1\, u2 - 2\, I2\, lc2\, m2 \sin(q2)\, Dq1^2\, l1 + 2\, I2\, u2 - 2\, m2^2\, lc2^3 \sin(q2)\, Dq1^2\, l1 + 2\, m2\, lc2^2\, u2 \\ & - 2\, m2^2\, l1^3\, lc2 \sin(q2)\, Dq1^2 + 2\, m2\, l1^2\, u2 - 2\, I1\, lc2\, m2 \sin(q2)\, Dq1^2\, l1 + 2\, I1\, u2 - 2\, m1\, lc1^2\, lc2\, m2 \sin(q2)\, Dq1^2\, l1 + 2\, m1\, lc1^2\, u2\big) \Big/ \\ & \big(2\, m2\, l1^2\, I2 + m2^2\, l1^2\, lc2^2 + 2\, I1\, I2 + 2\, I1\, m2\, lc2^2 + 2\, m1\, lc1^2\, I2 + 2\, m1\, lc1^2\, m2\, lc2^2 - m2^2\, l1^2\, lc2^2 \cos(2\, q2)\big)\Big]\Big] \end{aligned} \tag{11}$$

>

> **#Computing the inertias**

> $Inertia := int\left((x^2 + y^2) \cdot \rho, \left[x = -\frac{l}{2} .. \frac{l}{2}, y = -\frac{b}{2} .. \frac{b}{2}, z = -\frac{b}{2} .. \frac{b}{2}\right]\right)$:

>

> **#Rewriting equations in Matlab code to simplify Matlab input**

> $with(CodeGeneration)$:

> $Matlab(DynSysWithGrav(1), resultname = \text{"dx2"})$

```
> Matlab(DynSysWithGrav(2), resultname = "dx4")
> Matlab(DynSysWithoutGrav(1), resultname = "dx2")
> Matlab(DynSysWithoutGrav(2), resultname = "dx4")
> Epot := m1 · g · lc1 · sin(q1) + m2 · g ·(l1 · sin(q1) + lc2 · sin(q1 + q2))
```

$$Epot := m1\, g\, lc1 \sin(q1) + m2\, g\, (l1 \sin(q1) + lc2 \sin(q1 + q2)) \tag{12}$$

```
> Ekin := 1/2 ·VectorMatrixMultiply(Transpose(Dq), MatrixVectorMultiply(M0, Dq))
```

$$Ekin := \frac{((2\, lc2\, m2 \cos(q2)\, l1 + I2 + m2\, lc2^2 + m2\, l1^2 + I1 + m1\, lc1^2)\, Dq1 + (lc2\, m2 \cos(q2)\, l1 + m2\, lc2^2 + I2)\, Dq2)\, Dq1}{2} + \frac{((lc2\, m2 \cos(q2)\, l1 + m2\, lc2^2 + I2)\, Dq1 + (I2 + m2\, lc2^2)\, Dq2)\, Dq2}{2} \tag{13}$$

```
>
```

Dynamic Model: First 3-DOF of the 6-DOF IRB 140 manipulator

```
>
> restart :
> with(LinearAlgebra) :
>
> #Defining joint variables
> q := Vector([[q1(t)], [q2(t)], [q3(t)]]) :
> Dq := map(diff, q, t) :
> DDq := map(diff, Dq, t) :
>
> #Defining link length vectors
> r0c1 := Vector([[lc1x], [-lc1y], [0]]) :
> r1c2 := Vector([[0], [lc2], [0]]) :
> r2c3 := Vector([[lc3], [0], [0]]) :
> r01 := Vector([[l1x], [-l1y], [0]]) :
> r12 := Vector([[0], [l2], [0]]) :
> r23 := Vector([[l3], [0], [0]]) :
> r1c1 := r0c1 - r01 :
> r2c2 := r1c2 - r12 :
> r3c3 := r2c3 - r23 :
>
> #Rotation matrices
```

$$> R01 := MatrixMatrixMultiply\left(MatrixMatrixMultiply\left(\begin{bmatrix} \cos(q[1]) & -\sin(q[1]) & 0 \\ \sin(q[1]) & \cos(q[1]) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}\right), \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

$$R01 := \begin{bmatrix} 0 & -\cos(q1(t)) & \sin(q1(t)) \\ 0 & -\sin(q1(t)) & -\cos(q1(t)) \\ 1 & 0 & 0 \end{bmatrix} \tag{1}$$

$$> R12 := MatrixMatrixMultiply\left(\begin{bmatrix} \cos\left(\left(q\left[2\right]\right) - \frac{pi}{2}\right) & -\sin\left(q\left[2\right] - \frac{pi}{2}\right) & 0 \\ \sin\left(q\left[2\right] - \frac{pi}{2}\right) & \cos\left(q\left[2\right] - \frac{pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

$$R12 := \begin{bmatrix} \cos\left(-q2(t) + \frac{\pi}{2}\right) & \sin\left(-q2(t) + \frac{\pi}{2}\right) & 0 \\ -\sin\left(-q2(t) + \frac{\pi}{2}\right) & \cos\left(-q2(t) + \frac{\pi}{2}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$> R23 := MatrixMatrixMultiply\left(\begin{bmatrix} \cos(q[3]) & -\sin(q[3]) & 0 \\ \sin(q[3]) & \cos(q[3]) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

$$R23 := \begin{bmatrix} \cos(q3(t)) & -\sin(q3(t)) & 0 \\ \sin(q3(t)) & \cos(q3(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

```
> R02 := MatrixMatrixMultiply(R01, R12) :
> R03 := MatrixMatrixMultiply(R02, R23) :
>
> #Axis of rotation of joint i expressed in frame i
> z0 := Vector([[0], [0], [1]]) :
> b1 := MatrixMatrixMultiply(Transpose(R01), z0)
```

$$b1 := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{4}$$

```
> b2 := combine(MatrixMatrixMultiply(Transpose(R02), MatrixVectorMultiply(R01, z0)), trig)
```

$$b2 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{5}$$

```
> b3 := combine(MatrixMatrixMultiply(Transpose(R03), MatrixVectorMultiply(R02, z0)), trig)
```

(6)

$$b3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{6}$$

```
>
> #Gravity vector in inertial frame
```
> $g0 := Vector([[0], [0], [-g]])$:

```
>
> #Inertia matrices
```
> $I1 := \begin{bmatrix} I1xx & 0 & 0 \\ 0 & I1yy & 0 \\ 0 & 0 & I1zz \end{bmatrix}$:

> $I2 := \begin{bmatrix} I2xx & 0 & 0 \\ 0 & I2yy & 0 \\ 0 & 0 & I2zz \end{bmatrix}$:

> $I3 := \begin{bmatrix} I3xx & 0 & 0 \\ 0 & I3yy & 0 \\ 0 & 0 & I3zz \end{bmatrix}$:

```
>
> #Forward recursion: Link 1
```
> $\omega 1 := b1 \cdot Dq[1]$

$$\omega 1 := \begin{bmatrix} \dot{q1}(t) \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

> $\alpha 1 := b1 \cdot DDq[1] + CrossProduct(\omega 1, b1 \cdot Dq[1])$

$$\alpha 1 := \begin{bmatrix} \ddot{q1}(t) \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

> $D\omega 1 := map(diff, \omega 1, t)$:

> $ae1 := CrossProduct(D\omega 1, r01) + CrossProduct(\omega 1, CrossProduct(\omega 1, r01))$

$$ae1 := \begin{bmatrix} 0 \\ \dot{q1}(t)^2\, l1y \\ -\ddot{q1}(t)\, l1y \end{bmatrix} \tag{9}$$

> $ac1 := CrossProduct(D\omega 1, r0c1) + CrossProduct(\omega 1, CrossProduct(\omega 1, r0c1))$

$$ac1 := \begin{bmatrix} 0 \\ \dot{q1}(t)^2\, lc1y \\ -\ddot{q1}(t)\, lc1y \end{bmatrix} \tag{10}$$

> $g1 := MatrixVectorMultiply(Transpose(R01), g0)$

$$g1 := \begin{bmatrix} -g \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

```
>
> #Forward recursion: Link 2
```
> $\omega 2 := MatrixVectorMultiply(Transpose(R12), \omega 1) + b2 \cdot Dq[2]$:

> $\omega 2 := combine(\omega 2, trig)$

$$\omega 2 := \begin{bmatrix} \cos\left(-q2(t) + \frac{\pi}{2}\right) \dot{q1}(t) \\ \sin\left(-q2(t) + \frac{\pi}{2}\right) \dot{q1}(t) \\ \dot{q2}(t) \end{bmatrix} \tag{12}$$

> $\alpha 2 := MatrixVectorMultiply(Transpose(R12), \alpha 1) + b2 \cdot DDq[2] + CrossProduct(\omega 2, b2 \cdot Dq[2])$:

> $\alpha 2 := combine(\alpha 2, trig)$

(13)

$$\alpha2 := \begin{bmatrix} \cos\left(-q2(t)+\frac{\pi}{2}\right)\ddot{q1}(t)+\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)\,\dot{q2}(t) \\ \sin\left(-q2(t)+\frac{\pi}{2}\right)\ddot{q1}(t)-\cos\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)\,\dot{q2}(t) \\ \ddot{q2}(t) \end{bmatrix} \tag{13}$$

```
> Dω2 := map(diff, ω2, t) :
> ae2 := MatrixVectorMultiply(Transpose(R12), ae1) + CrossProduct(Dω2, r12) + CrossProduct(ω2, CrossProduct(ω2, r12)) :
> ae2 := combine(ae2, trig)
```

$$ae2 := \begin{bmatrix} -\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)^2\, l1y-\ddot{q2}(t)\, l2+\frac{\dot{q1}(t)^2\, l2\sin(-2\,q2(t)+\pi)}{2} \\ \cos\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)^2\, l1y-\dot{q2}(t)^2\, l2-\frac{\dot{q1}(t)^2\, l2\cos(-2\,q2(t)+\pi)}{2}-\frac{\dot{q1}(t)^2\, l2}{2} \\ -\ddot{q1}(t)\, l1y+l2\cos\left(-q2(t)+\frac{\pi}{2}\right)\ddot{q1}(t)+2\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)\,\dot{q2}(t)\, l2 \end{bmatrix} \tag{14}$$

```
> ac2 := MatrixVectorMultiply(Transpose(R12), ae1) + CrossProduct(Dω2, r1c2) + CrossProduct(ω2, CrossProduct(ω2, r1c2)) :
> ac2 := combine(ac2, trig)
```

$$ac2 := \begin{bmatrix} -\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)^2\, l1y-\ddot{q2}(t)\, lc2+\frac{\dot{q1}(t)^2\, lc2\sin(-2\,q2(t)+\pi)}{2} \\ \cos\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)^2\, l1y-\dot{q2}(t)^2\, lc2-\frac{\dot{q1}(t)^2\, lc2\cos(-2\,q2(t)+\pi)}{2}-\frac{\dot{q1}(t)^2\, lc2}{2} \\ -\ddot{q1}(t)\, l1y+lc2\cos\left(-q2(t)+\frac{\pi}{2}\right)\ddot{q1}(t)+2\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)\,\dot{q2}(t)\, lc2 \end{bmatrix} \tag{15}$$

```
> g2 := MatrixVectorMultiply(Transpose(R02), g0) :
> g2 := combine(g2, trig)
```

$$g2 := \begin{bmatrix} -\cos\left(-q2(t)+\frac{\pi}{2}\right)g \\ -\sin\left(-q2(t)+\frac{\pi}{2}\right)g \\ 0 \end{bmatrix} \tag{16}$$

```
>
> #Forward recursion: Link 3
> ω3 := MatrixVectorMultiply(Transpose(R23), ω2) + b3·Dq[3] :
> ω3 := combine(ω3, trig)
```

$$\omega3 := \begin{bmatrix} \dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right) \\ \dot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right) \\ \dot{q2}(t)+\dot{q3}(t) \end{bmatrix} \tag{17}$$

```
> α3 := MatrixVectorMultiply(Transpose(R23), α2) + b3·DDq[3] + CrossProduct(ω3, b3·Dq[3]) :
> α3 := combine(α3, trig)
```

$$\alpha3 := \begin{bmatrix} \ddot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+\dot{q1}(t)\,\dot{q2}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+\dot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q3}(t) \\ \ddot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)-\dot{q1}(t)\,\dot{q2}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)-\dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q3}(t) \\ \ddot{q2}(t)+\ddot{q3}(t) \end{bmatrix} \tag{18}$$

```
> Dω3 := map(diff, ω3, t) :
> ae3 := MatrixVectorMultiply(Transpose(R23), ae2) + CrossProduct(Dω3, r23) + CrossProduct(ω3, CrossProduct(ω3, r23)) :
> ae3 := combine(ae3, trig)
```

$$ae3 := \Bigg[\Bigg[-\dot{q1}(t)^2\, l1y\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)-\cos(q3(t))\,\ddot{q2}(t)\, l2+\frac{\dot{q1}(t)^2\, l2\sin(-q3(t)-2\,q2(t)+\pi)}{2}-\sin(q3(t))\,\dot{q2}(t)^2\, l2$$
$$-\frac{\sin(q3(t))\,\dot{q1}(t)^2\, l2}{2}-\frac{\dot{q1}(t)^2\, l3}{2}+\frac{\dot{q1}(t)^2\, l3\cos(-2\,q3(t)-2\,q2(t)+\pi)}{2}-l3\,\dot{q2}(t)^2-2\, l3\,\dot{q2}(t)\,\dot{q3}(t)-l3\,\dot{q3}(t)^2\Bigg],$$
$$\Bigg[\dot{q1}(t)^2\, l1y\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+\sin(q3(t))\,\ddot{q2}(t)\, l2-\frac{\dot{q1}(t)^2\, l2\cos(-q3(t)-2\,q2(t)+\pi)}{2}-\cos(q3(t))\,\dot{q2}(t)^2\, l2$$
$$-\frac{\cos(q3(t))\,\dot{q1}(t)^2\, l2}{2}+l3\,\ddot{q2}(t)+l3\,\ddot{q3}(t)+\frac{\dot{q1}(t)^2\, l3\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}\Bigg], \tag{19}$$

$$\left[-\ddot{q1}(t)\,l1y+l2\cos\left(-q2(t)+\frac{\pi}{2}\right)\ddot{q1}(t)+2\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)\,\dot{q2}(t)\,l2-l3\,\ddot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+2\,l3\,\dot{q1}(t)\,\dot{q2}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+2\,l3\,\dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q3}(t)\right]\right]$$

> *ac3* := *MatrixVectorMultiply*(*Transpose*(*R23*), *ae2*) + *CrossProduct*(*D*ω*3*, *r2c3*) + *CrossProduct*(ω*3*, *CrossProduct*(ω*3*, *r2c3*)) :

> *ac3* := *combine*(*ac3*, *trig*)

$$ac3 := \left[\left[-\dot{q1}(t)^2\,l1y\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)-\cos(q3(t))\,\ddot{q2}(t)\,l2+\frac{\dot{q1}(t)^2\,l2\sin(-q3(t)-2\,q2(t)+\pi)}{2}-\sin(q3(t))\,\dot{q2}(t)^2\,l2-\frac{\sin(q3(t))\,\dot{q1}(t)^2\,l2}{2}-\frac{\dot{q1}(t)^2\,lc3}{2}+\frac{\dot{q1}(t)^2\,lc3\cos(-2\,q3(t)-2\,q2(t)+\pi)}{2}-lc3\,\dot{q2}(t)^2-2\,lc3\,\dot{q2}(t)\,\dot{q3}(t)-lc3\,\dot{q3}(t)^2\right],\right.$$
$$\left[\dot{q1}(t)^2\,l1y\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+\sin(q3(t))\,\ddot{q2}(t)\,l2-\frac{\dot{q1}(t)^2\,l2\cos(-q3(t)-2\,q2(t)+\pi)}{2}-\cos(q3(t))\,\dot{q2}(t)^2\,l2-\frac{\cos(q3(t))\,\dot{q1}(t)^2\,l2}{2}+lc3\,\ddot{q2}(t)+lc3\,\ddot{q3}(t)+\frac{\dot{q1}(t)^2\,lc3\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}\right],$$
$$\left.\left[-\ddot{q1}(t)\,l1y+l2\cos\left(-q2(t)+\frac{\pi}{2}\right)\ddot{q1}(t)+2\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)\,\dot{q2}(t)\,l2-lc3\,\ddot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+2\,lc3\,\dot{q1}(t)\,\dot{q2}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+2\,lc3\,\dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q3}(t)\right]\right] \tag{20}$$

> *g3* := *MatrixVectorMultiply*(*Transpose*(*R03*), *g0*) :

> *g3* := *combine*(*g3*, *trig*)

$$g3 := \begin{bmatrix} -\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)g \\ -\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)g \\ 0 \end{bmatrix} \tag{21}$$

>

> **#Backward recursion: Link 3**

> *f3* := *m3*·*ac3* − *m3*·*g3* :

> τ*3* := −*CrossProduct*(*f3*, *r2c3*) + *MatrixVectorMultiply*(*I3*, α*3*) + *CrossProduct*(ω*3*, *MatrixVectorMultiply*(*I3*, ω*3*)) :

> τ*3* := *combine*(τ*3*, *trig*)

$$\tau3 := \left[\left[I3xx\,\ddot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+I3xx\,\dot{q1}(t)\,\dot{q2}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+I3xx\,\dot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q3}(t)+\dot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)I3zz\,\dot{q2}(t)+\dot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)I3zz\,\dot{q3}(t)-I3yy\,\dot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q2}(t)-I3yy\,\dot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q3}(t)\right],\right.$$
$$\left[m3\,lc3\,\ddot{q1}(t)\,l1y-m3\,lc3\,l2\cos\left(-q2(t)+\frac{\pi}{2}\right)\ddot{q1}(t)-2\,m3\,lc3\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q1}(t)\,\dot{q2}(t)\,l2+m3\,lc3^2\,\ddot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)-2\,m3\,lc3^2\,\dot{q1}(t)\,\dot{q2}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)-2\,m3\,lc3^2\,\dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q3}(t)+I3yy\,\ddot{q1}(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)-I3yy\,\dot{q1}(t)\,\dot{q2}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)-I3yy\,\dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q3}(t)+I3xx\,\dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q2}(t)+I3xx\,\dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q3}(t)-\dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)I3zz\,\dot{q2}(t)-\dot{q1}(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)I3zz\,\dot{q3}(t)\right],$$
$$\left.\left[lc3\,m3\,\dot{q1}(t)^2\,l1y\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)+lc3\,m3\sin(q3(t))\,\ddot{q2}(t)\,l2-\frac{lc3\,m3\,\dot{q1}(t)^2\,l2\cos(-q3(t)-2\,q2(t)+\pi)}{2}-lc3\,m3\cos(q3(t))\,\dot{q2}(t)^2\,l2-\frac{lc3\,m3\cos(q3(t))\,\dot{q1}(t)^2\,l2}{2}+m3\,lc3^2\,\ddot{q2}(t)+m3\,lc3^2\,\ddot{q3}(t)+\frac{m3\,\dot{q1}(t)^2\,lc3^2\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}+lc3\,m3\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)g+I3zz\,\ddot{q2}(t)+I3zz\,\ddot{q3}(t)+\frac{\dot{q1}(t)^2\,I3yy\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}-\frac{\dot{q1}(t)^2\,I3xx\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}\right]\right] \tag{22}$$

> τ*3z* := *collect*(*combine*(τ*3*[3], *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* :

> τ*2* := *MatrixVectorMultiply*(*R23*, τ*3*) − *CrossProduct*(*f2*, *r1c2*) + *CrossProduct*(*MatrixVectorMultiply*(*R23*, *f3*), *r2c2*) + *MatrixVectorMultiply*(*I2*, α*2*) + *CrossProduct*(ω*2*, *MatrixVectorMultiply*(*I2*, ω*2*)) :

> τ*2* := *combine*(τ*2*, *trig*)

$$
\begin{aligned}
&\tau2 := \Big[\Big[I2xx \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\,\ddot{q1}(t) - \sin(q3(t))\, m3\, lc3\, \ddot{q1}(t)\, l1y + 2\, m2 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t)\,\dot{q2}(t)\, lc2^2 + 2\, m3 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t)\\
&\dot{q2}(t)\, l2^2 - \frac{3\, m3\, lc3\, \ddot{q1}(t)\sin\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l2}{2} - lc2\, m2\, \ddot{q1}(t)\, l1y + m2\, lc2^2 \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\,\ddot{q1}(t) - m3\, \ddot{q1}(t)\, l1y\, l2\\
&+ m3\, l2^2 \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\,\ddot{q1}(t) + I2xx \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t)\,\dot{q2}(t) + I3xx\, \dot{q1}(t)\,\dot{q2}(t) \sin\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big) + I3xx\, \dot{q1}(t)\\
&\dot{q3}(t) \sin\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big) + \dot{q1}(t)\, I3zz\, \dot{q2}(t) \sin\Big(-q2(t)+\frac{\pi}{2}\Big) + \dot{q1}(t)\, I3zz\, \dot{q3}(t) \sin\Big(-q2(t)+\frac{\pi}{2}\Big) - I3yy\, \dot{q1}(t)\,\dot{q2}(t) \sin\Big(-2\, q3(t)\\
&-q2(t)+\frac{\pi}{2}\Big) - I3yy\, \dot{q1}(t)\,\dot{q3}(t) \sin\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big) + 3\, m3\, lc3\, \dot{q1}(t)\,\dot{q2}(t) \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l2 + 2\, m3\, lc3\, \dot{q1}(t) \cos\Big(\\
&-q3(t)-q2(t)+\frac{\pi}{2}\Big)\,\dot{q3}(t)\, l2 - m3\, lc3\, \dot{q1}(t)\,\dot{q2}(t)\, l2 \cos\Big(-q2(t)+\frac{\pi}{2}+q3(t)\Big) + \frac{I3xx\, \ddot{q1}(t) \cos\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big)}{2}\\
&+ \frac{I3xx\, \ddot{q1}(t) \cos\Big(-q2(t)+\frac{\pi}{2}\Big)}{2} + \frac{I3yy\, \ddot{q1}(t) \cos\Big(-q2(t)+\frac{\pi}{2}\Big)}{2} + \frac{m3\, lc3\, l2\, \ddot{q1}(t) \sin\Big(-q2(t)+\frac{\pi}{2}+q3(t)\Big)}{2} + m3\, lc3^2\, \dot{q1}(t)\,\dot{q2}(t) \sin\Big(\\
&-q2(t)+\frac{\pi}{2}\Big) - m3\, lc3^2\, \dot{q1}(t)\,\dot{q2}(t) \sin\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big) + m3\, lc3^2\, \dot{q1}(t)\,\dot{q3}(t) \sin\Big(-q2(t)+\frac{\pi}{2}\Big) - m3\, lc3^2\, \dot{q1}(t)\,\dot{q3}(t) \sin\Big(-2\, q3(t)\\
&-q2(t)+\frac{\pi}{2}\Big) - \frac{m3\, lc3^2\, \ddot{q1}(t) \cos\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big)}{2} + \frac{m3\, lc3^2\, \ddot{q1}(t) \cos\Big(-q2(t)+\frac{\pi}{2}\Big)}{2} + \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t)\, I2zz\, \dot{q2}(t) -\\
&\dot{q2}(t)\, I2yy \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t) - \frac{I3yy\, \ddot{q1}(t) \cos\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big)}{2}\Big],\\
&\Big[\frac{I3yy\, \ddot{q1}(t) \sin\Big(-q2(t)+\frac{\pi}{2}\Big)}{2} - m3\, lc3\, \dot{q1}(t)\,\dot{q2}(t)\, l2 \sin\Big(-q2(t)+\frac{\pi}{2}+q3(t)\Big) - m3\, lc3\, \dot{q1}(t)\,\dot{q2}(t)\, l2 \sin\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\\
&+ I2yy \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\,\ddot{q1}(t) + \frac{I3xx\, \ddot{q1}(t) \sin\Big(-q2(t)+\frac{\pi}{2}\Big)}{2} - \frac{I3xx\, \ddot{q1}(t) \sin\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big)}{2} - I2yy \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t)\\
&\dot{q2}(t) + I3xx\, \dot{q1}(t)\,\dot{q2}(t) \cos\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big) + I3xx\, \dot{q1}(t)\,\dot{q3}(t) \cos\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big) - \dot{q1}(t)\, I3zz\, \dot{q2}(t) \cos\Big(-q2(t)+\frac{\pi}{2}\Big) -\\
&\dot{q1}(t)\, I3zz\, \dot{q3}(t) \cos\Big(-q2(t)+\frac{\pi}{2}\Big) - I3yy\, \dot{q1}(t)\,\dot{q2}(t) \cos\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big) - m3\, lc3^2\, \dot{q1}(t)\,\dot{q2}(t) \cos\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big)\\
&- m3\, lc3^2\, \dot{q1}(t)\,\dot{q2}(t) \cos\Big(-q2(t)+\frac{\pi}{2}\Big) + \cos(q3(t))\, m3\, lc3\, \ddot{q1}(t)\, l1y - \frac{m3\, lc3\, l2\, \ddot{q1}(t) \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)}{2}\\
&- \frac{m3\, lc3\, l2\, \ddot{q1}(t) \cos\Big(-q2(t)+\frac{\pi}{2}+q3(t)\Big)}{2} - I3yy\, \dot{q1}(t)\,\dot{q3}(t) \cos\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big) + \frac{m3\, lc3^2\, \ddot{q1}(t) \sin\Big(-q2(t)+\frac{\pi}{2}\Big)}{2}\\
&+ \frac{m3\, lc3^2\, \ddot{q1}(t) \sin\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big)}{2} + \frac{I3yy\, \ddot{q1}(t) \sin\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big)}{2} - m3\, lc3^2\, \dot{q1}(t)\,\dot{q3}(t) \cos\Big(-2\, q3(t)-q2(t)+\frac{\pi}{2}\Big)\\
&- m3\, lc3^2\, \dot{q1}(t)\,\dot{q3}(t) \cos\Big(-q2(t)+\frac{\pi}{2}\Big) + \dot{q2}(t)\, I2xx \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t) - \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t)\, I2zz\, \dot{q2}(t)\Big],\\
&\Big[-\frac{\dot{q1}(t)^2\, I2xx \sin(-2\, q2(t)+\pi)}{2} + I2zz\, \ddot{q2}(t) + lc2\, m2 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t)^2\, l1y + \cos(q3(t))\, m3\, lc3\, \dot{q3}(t)^2\, l2 + \sin(q3(t))\, m3\, lc3\\
&\ddot{q3}(t)\, l2 - \frac{m2\, \dot{q1}(t)^2\, lc2^2 \sin(-2\, q2(t)+\pi)}{2} - lc2\, m2 \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, g + lc3\, m3\, \dot{q1}(t)^2\, l1y \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\\
&+ 2\, lc3\, m3 \sin(q3(t))\, \ddot{q2}(t)\, l2 - lc3\, m3\, \dot{q1}(t)^2\, l2 \cos(-q3(t)-2\, q2(t)+\pi) + m3 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\,\dot{q1}(t)^2\, l1y\, l2
\end{aligned}
\tag{23}
$$

$$+\frac{\dot{q}1(t)^2\, I2yy \sin(-2\,q2(t)+\pi)}{2}-\frac{m3\,\dot{q}1(t)^2\, l2^2 \sin(-2\,q2(t)+\pi)}{2}-m3\cos\left(-q2(t)+\frac{\pi}{2}\right) g\, l2$$

$$-\frac{\dot{q}1(t)^2\, I3xx \sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}+m3\,lc3^2\,\ddot{q}3(t)+m3\,lc3^2\,\ddot{q}2(t)+m2\,\ddot{q}2(t)\,lc2^2+2\cos(q3(t))\,m3\,lc3\,\dot{q}2(t)\,\dot{q}3(t)\,l2$$

$$+\frac{m3\,\dot{q}1(t)^2\,lc3^2\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}+lc3\,m3\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)g+I3zz\,\ddot{q}2(t)+I3zz\,\ddot{q}3(t)$$

$$+\frac{\dot{q}1(t)^2\,I3yy\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}+m3\,\ddot{q}2(t)\,l2^2\Big]\Big]$$

> $\tau 2z := collect(combine(\tau 2[3], trig), \{DDq[1], DDq[2], DDq[3], Dq[1], Dq[2], Dq[3]\})$:

>

> **#Backward recursion: Link 1**

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

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

> $\tau 1 := combine(\tau 1, trig)$

$$\tau 1 := \Big[\Big[I3xx\,\dot{q}1(t)\,\dot{q}3(t)\sin(-2\,q3(t)-2\,q2(t)+\pi)+I3xx\,\dot{q}1(t)\,\dot{q}2(t)\sin(-2\,q3(t)-2\,q2(t)+\pi)-I3yy\,\dot{q}1(t)\,\dot{q}2(t)\sin(-2\,q3(t)-2\,q2(t)+\pi) \qquad (24)$$

$$-2\cos\left(-q2(t)+\frac{\pi}{2}\right)lc2\,m2\,\ddot{q}1(t)\,l1y-2\cos\left(-q2(t)+\frac{\pi}{2}\right)m3\,\ddot{q}1(t)\,l1y\,l2+2\,m3\,lc3\,\ddot{q}1(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)l1y$$

$$-\frac{m3\,lc3^2\,\ddot{q}1(t)\cos(-2\,q3(t)-2\,q2(t)+\pi)}{2}-\dot{q}2(t)\,I2yy\,\dot{q}1(t)\sin(-2\,q2(t)+\pi)-I3yy\,\dot{q}1(t)\,\dot{q}3(t)\sin(-2\,q3(t)-2\,q2(t)+\pi)$$

$$-2\,m3\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q}1(t)\,\dot{q}2(t)\,l2\,l1y-2\,m3\,lc3\,\dot{q}1(t)\,\dot{q}2(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)l1y-2\,m3\,lc3\,\dot{q}1(t)\cos\Big(-q3(t)-q2(t)$$

$$+\frac{\pi}{2}\Big)\dot{q}3(t)\,l1y-2\,m2\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q}1(t)\,\dot{q}2(t)\,lc2\,l1y+m2\,\dot{q}1(t)\,\dot{q}2(t)\,lc2^2\sin(-2\,q2(t)+\pi)+m3\,\dot{q}1(t)\,\dot{q}2(t)\,l2^2\sin(-2\,q2(t)+\pi)$$

$$+2\,m3\,lc3\,\dot{q}1(t)\,\dot{q}2(t)\,l2\cos(-q3(t)-2\,q2(t)+\pi)+m3\,lc3\,\dot{q}1(t)\,\dot{q}3(t)\,l2\cos(q3(t))+m3\,lc3\,\dot{q}1(t)\,\dot{q}3(t)\,l2\cos(-q3(t)-2\,q2(t)+\pi)$$

$$+I2xx\,\dot{q}1(t)\,\dot{q}2(t)\sin(-2\,q2(t)+\pi)+\frac{I2xx\,\ddot{q}1(t)\cos(-2\,q2(t)+\pi)}{2}+\frac{I3xx\,\ddot{q}1(t)\cos(-2\,q3(t)-2\,q2(t)+\pi)}{2}$$

$$-\frac{I3yy\,\ddot{q}1(t)\cos(-2\,q3(t)-2\,q2(t)+\pi)}{2}+\frac{m3\,l2^2\,\ddot{q}1(t)}{2}+\frac{m2\,lc2^2\,\ddot{q}1(t)}{2}+\frac{m3\,lc3^2\,\ddot{q}1(t)}{2}+m1\,\ddot{q}1(t)\,lc1y^2+m3\,\ddot{q}1(t)\,l1y^2+m2$$

$$\ddot{q}1(t)\,l1y^2+\frac{m3\,l2^2\,\ddot{q}1(t)\cos(-2\,q2(t)+\pi)}{2}+\frac{m2\,lc2^2\,\ddot{q}1(t)\cos(-2\,q2(t)+\pi)}{2}-m3\,lc3\,\ddot{q}1(t)\,l2\sin(-q3(t)-2\,q2(t)+\pi)+m3\,lc3$$

$$\ddot{q}1(t)\,l2\sin(q3(t))-m3\,lc3^2\,\dot{q}1(t)\,\dot{q}2(t)\sin(-2\,q3(t)-2\,q2(t)+\pi)-m3\,lc3^2\,\dot{q}1(t)\,\dot{q}3(t)\sin(-2\,q3(t)-2\,q2(t)+\pi)$$

$$-\frac{I2yy\,\ddot{q}1(t)\cos(-2\,q2(t)+\pi)}{2}+\frac{I2yy\,\ddot{q}1(t)}{2}+\frac{I2xx\,\ddot{q}1(t)}{2}+\frac{I3xx\,\ddot{q}1(t)}{2}+\frac{I3yy\,\ddot{q}1(t)}{2}+I1xx\,\ddot{q}1(t)\Big],$$

$$\Big[m3\,lc3\,\ddot{q}1(t)\,l1y\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)-m3\,lc3\,\ddot{q}1(t)\,l2\cos(-q3(t)-2\,q2(t)+\pi)-m3\,lc3^2\,\dot{q}1(t)\,\dot{q}2(t)\cos(-2\,q3(t)-2\,q2(t)+\pi)$$

$$-m3\,lc3^2\,\dot{q}1(t)\,\dot{q}3(t)\cos(-2\,q3(t)-2\,q2(t)+\pi)+m2\,\dot{q}1(t)\,\dot{q}2(t)\,lc2^2\cos(-2\,q2(t)+\pi)+m3\,\dot{q}1(t)\,\dot{q}2(t)\,l2^2\cos(-2\,q2(t)+\pi)$$

$$-\frac{I2xx\,\ddot{q}1(t)\sin(-2\,q2(t)+\pi)}{2}-\frac{I3xx\,\ddot{q}1(t)\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}+\frac{I3yy\,\ddot{q}1(t)\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}-\dot{q}1(t)\,I3zz\,\dot{q}3(t)-$$

$$\dot{q}1(t)\,I3zz\,\dot{q}2(t)-\dot{q}1(t)\,I2zz\,\dot{q}2(t)-2\,m3\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q}1(t)\,\dot{q}2(t)\,l2\,l1x-2\,m3\,lc3\,\dot{q}1(t)\,\dot{q}2(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)l1x$$

$$-2\,m3\,lc3\,\dot{q}1(t)\cos\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)\dot{q}3(t)\,l1x-2\,m2\sin\left(-q2(t)+\frac{\pi}{2}\right)\dot{q}1(t)\,\dot{q}2(t)\,lc2\,l1x+I2xx\,\dot{q}1(t)\,\dot{q}2(t)\cos(-2\,q2(t)+\pi)-$$

$$\dot{q}2(t)\,I2yy\,\dot{q}1(t)\cos(-2\,q2(t)+\pi)+\sin\left(-q2(t)+\frac{\pi}{2}\right)lc2\,m2\,\ddot{q}1(t)\,l1y+\sin\left(-q2(t)+\frac{\pi}{2}\right)m3\,\ddot{q}1(t)\,l1y\,l2-m3\,l2\cos\left(-q2(t)+\frac{\pi}{2}\right)$$

$$\ddot{q}1(t)\,l1x+m3\,lc3\,\ddot{q}1(t)\sin\left(-q3(t)-q2(t)+\frac{\pi}{2}\right)l1x-m2\,lc2\cos\left(-q2(t)+\frac{\pi}{2}\right)\ddot{q}1(t)\,l1x+\frac{I2yy\,\ddot{q}1(t)\sin(-2\,q2(t)+\pi)}{2}-m3\,lc3^2\,\dot{q}1(t)$$

$$\dot{q}2(t)-m3\,lc3^2\,\dot{q}1(t)\,\dot{q}3(t)-m2\,\dot{q}1(t)\,\dot{q}2(t)\,lc2^2-m3\,\dot{q}1(t)\,\dot{q}2(t)\,l2^2+I3xx\,\dot{q}1(t)\,\dot{q}3(t)\cos(-2\,q3(t)-2\,q2(t)+\pi)+I3xx\,\dot{q}1(t)\,\dot{q}2(t)\cos($$

$$-2\,q3(t)-2\,q2(t)+\pi)-\frac{m3\,l2^2\,\ddot{q}1(t)\sin(-2\,q2(t)+\pi)}{2}-\frac{m2\,lc2^2\,\ddot{q}1(t)\sin(-2\,q2(t)+\pi)}{2}-I3yy\,\dot{q}1(t)\,\dot{q}2(t)\cos(-2\,q3(t)-2\,q2(t)$$

$$+\pi)-I3yy\,\dot{q}1(t)\,\dot{q}3(t)\cos(-2\,q3(t)-2\,q2(t)+\pi)+\frac{m3\,lc3^2\,\ddot{q}1(t)\sin(-2\,q3(t)-2\,q2(t)+\pi)}{2}-2\,m3\,lc3\,\dot{q}1(t)\,\dot{q}2(t)\,l2\sin(-q3(t)$$

$$-2\,q2(t)+\pi)-2\,m3\,lc3\,\dot{q}1(t)\,\dot{q}2(t)\,l2\sin(q3(t))-m3\,lc3\,\dot{q}1(t)\,\dot{q}3(t)\,l2\sin(-q3(t)-2\,q2(t)+\pi)-m3\,lc3\,\dot{q}1(t)\,\dot{q}3(t)\,l2\sin(q3(t))$$

$$+lc1x\, m1\, \ddot{q1}(t)\, lc1y + m3\, \ddot{q1}(t)\, l1y\, l1x + m2\, \ddot{q1}(t)\, l1y\, l1x\Big],$$

$$\Big[-\frac{\dot{q1}(t)^2\, I2xx \sin(-2\, q2(t)+\pi)}{2} + I2zz\, \ddot{q2}(t) + lc2\, m2 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, \dot{q1}(t)^2\, l1y + \cos(q3(t))\, m3\, lc3\, \dot{q3}(t)^2\, l2 + \sin(q3(t))\, m3\, lc3$$

$$\ddot{q3}(t)\, l2 - \frac{m2\, \dot{q1}(t)^2\, lc2^2 \sin(-2\, q2(t)+\pi)}{2} - lc2\, m2 \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, g + lc3\, m3\, \dot{q1}(t)^2\, l1y \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)$$

$$+2\, lc3\, m3 \sin(q3(t))\, \ddot{q2}(t)\, l2 - lc3\, m3\, \dot{q1}(t)^2\, l2 \cos(-q3(t)-2\, q2(t)+\pi) + m3 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, \dot{q1}(t)^2\, l1y\, l2$$

$$+\frac{\dot{q1}(t)^2\, I2yy \sin(-2\, q2(t)+\pi)}{2} - \frac{m3\, \dot{q1}(t)^2\, l2^2 \sin(-2\, q2(t)+\pi)}{2} - m3 \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, g\, l2 - 2\, m3\, lc3\, \dot{q2}(t)\, \dot{q3}(t) \cos\Big(-q3(t)$$

$$-q2(t)+\frac{\pi}{2}\Big)\, l1y + lc1y\, m1\, g - \frac{\dot{q1}(t)^2\, I3xx \sin(-2\, q3(t)-2\, q2(t)+\pi)}{2} + m3\, lc3^2\, \ddot{q3}(t) + m3\, lc3^2\, \ddot{q2}(t) + 2\, m3\, lc3\, \dot{q2}(t)\, \dot{q3}(t) \sin\Big(-q3(t)$$

$$-q2(t)+\frac{\pi}{2}\Big)\, l1x + m2\, \ddot{q2}(t)\, lc2^2 - m3\, \dot{q1}(t)^2\, l2 \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, l1x - \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, m3\, \dot{q2}(t)^2\, l2\, l1x + 2 \cos(q3(t))\, m3\, lc3\, \dot{q2}(t)$$

$$\dot{q3}(t)\, l2 + \frac{m3\, \dot{q1}(t)^2\, lc3^2 \sin(-2\, q3(t)-2\, q2(t)+\pi)}{2} + lc3\, m3 \sin\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, g - \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, m3\, \ddot{q2}(t)\, l2\, l1y + m3\, lc3$$

$$\ddot{q2}(t) \sin\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l1y + m3\, lc3\, \ddot{q3}(t) \sin\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l1y - m3\, lc3\, \dot{q2}(t)^2 \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l1y$$

$$-m3\, lc3\, \dot{q3}(t)^2 \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l1y - \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, m3\, \dot{q2}(t)^2\, l2\, l1y + I3zz\, \ddot{q2}(t) + I3zz\, \ddot{q3}(t)$$

$$+\frac{\dot{q1}(t)^2\, I3yy \sin(-2\, q3(t)-2\, q2(t)+\pi)}{2} + m3\, \ddot{q2}(t)\, l2^2 + \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, m2\, \ddot{q2}(t)\, lc2\, l1x - \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, m2\, \dot{q1}(t)^2\, lc2\, l1x$$

$$-\cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, m2\, \dot{q2}(t)^2\, lc2\, l1x - \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, m2\, \ddot{q2}(t)\, lc2\, l1y - \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, m2\, \dot{q2}(t)^2\, lc2\, l1y + lc1x\, m1\, \dot{q1}(t)^2\, lc1y$$

$$+m3\, \dot{q1}(t)^2\, l1y\, l1x + m2\, \dot{q1}(t)^2\, l1y\, l1x + m3\, g\, l1y + m2\, g\, l1y + \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, m3\, \ddot{q2}(t)\, l2\, l1x + m3\, lc3\, \ddot{q2}(t) \cos\Big(-q3(t)-q2(t)$$

$$+\frac{\pi}{2}\Big)\, l1x + m3\, lc3\, \ddot{q3}(t) \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l1x + m3\, \dot{q1}(t)^2\, lc3 \sin\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l1x + m3\, lc3\, \dot{q2}(t)^2 \sin\Big(-q3(t)-q2(t)$$

$$+\frac{\pi}{2}\Big)\, l1x + m3\, lc3\, \dot{q3}(t)^2 \sin\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l1x\Big]\Big]$$

> $\tau 1x := collect(combine(\tau 1[1], trig), \{DDq[1], DDq[2], DDq[3], Dq[1], Dq[2], Dq[3]\})$:

> **#Setting up the matrix elements**

> $$m11 := \Big(-\frac{I3yy \cos(-2\, q3(t)-2\, q2(t)+\pi)}{2} + \frac{I3xx \cos(-2\, q3(t)-2\, q2(t)+\pi)}{2} + \frac{I2xx}{2} + m2\, l1y^2 + \frac{m2\, lc2^2}{2} + \frac{m3\, l2^2}{2} + \frac{m3\, lc3^2}{2} + \frac{I2yy}{2}$$
$$-\frac{I2yy \cos(-2\, q2(t)+\pi)}{2} + I1xx + 2\, m3\, lc3 \sin\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\, l1y - m3\, lc3\, l2 \sin(-q3(t)-2\, q2(t)+\pi) + \frac{I2xx \cos(-2\, q2(t)+\pi)}{2}$$
$$+lc3\, m3 \sin(q3(t))\, l2 + m1\, lc1y^2 + \frac{I3xx}{2} + \frac{m3\, l2^2 \cos(-2\, q2(t)+\pi)}{2} + m3\, l1y^2 - 2 \cos\Big(-q2(t)+\frac{\pi}{2}\Big)\, lc2\, m2\, l1y - 2 \cos\Big(-q2(t)$$
$$+\frac{\pi}{2}\Big)\, m3\, l1y\, l2 + \frac{I3yy}{2} - \frac{m3\, lc3^2 \cos(-2\, q3(t)-2\, q2(t)+\pi)}{2} + \frac{m2\, lc2^2 \cos(-2\, q2(t)+\pi)}{2}\Big):$$

> $$c11 := \Big(\Big(\Big(m3\, l2^2 \sin(-2\, q2(t)+\pi) - I3yy \sin(-2\, q3(t)-2\, q2(t)+\pi) + I2xx \sin(-2\, q2(t)+\pi) + I3xx \sin(-2\, q3(t)-2\, q2(t)+\pi) - I2yy \sin($$
$$-2\, q2(t)+\pi) - lc3^2\, m3 \sin(-2\, q3(t)-2\, q2(t)+\pi) - 2\, m3 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, l1y\, l2 + 2\, lc3\, m3\, l2 \cos(-q3(t)-2\, q2(t)+\pi)$$
$$-2\, lc3\, m3\, l1y \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big) - 2\, lc2\, m2 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, l1y + m2\, lc2^2 \sin(-2\, q2(t)+\pi)\Big)\cdot Dq[2]\Big) + \Big(lc3\, m3 \cos(q3(t))\, l2$$
$$+lc3\, m3\, l2 \cos(-q3(t)-2\, q2(t)+\pi) - lc3^2\, m3 \sin(-2\, q3(t)-2\, q2(t)+\pi) - I3yy \sin(-2\, q3(t)-2\, q2(t)+\pi) + I3xx \sin(-2\, q3(t)-2\, q2(t)$$
$$+\pi) - 2\, lc3\, m3\, l1y \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big)\Big)\cdot Dq[3]\Big):$$

> $m22 := (m3\, lc3^2 + I3zz + I2zz + m3\, l2^2 + 2\, lc3\, m3 \sin(q3(t))\, l2 + m2\, lc2^2)$:

> $m23 := (lc3\, m3 \sin(q3(t))\, l2 + m3\, lc3^2 + I3zz)$:

> $$c21 := \Big(\Big(-lc3\, m3\, l2 \cos(-q3(t)-2\, q2(t)+\pi) + lc2\, m2 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, l1y + \frac{I3yy \sin(-2\, q3(t)-2\, q2(t)+\pi)}{2}$$
$$-\frac{I3xx \sin(-2\, q3(t)-2\, q2(t)+\pi)}{2} + lc3\, m3\, l1y \cos\Big(-q3(t)-q2(t)+\frac{\pi}{2}\Big) - \frac{m2\, lc2^2 \sin(-2\, q2(t)+\pi)}{2} - \frac{m3\, l2^2 \sin(-2\, q2(t)+\pi)}{2}$$
$$+m3 \sin\Big(-q2(t)+\frac{\pi}{2}\Big)\, l1y\, l2 + \frac{lc3^2\, m3 \sin(-2\, q3(t)-2\, q2(t)+\pi)}{2} + \frac{I2yy \sin(-2\, q2(t)+\pi)}{2} - \frac{I2xx \sin(-2\, q2(t)+\pi)}{2}\Big)\Big)\cdot Dq[1]:$$

> $c22 := (2 \cos(q3(t))\, m3\, lc3\, l2)\cdot Dq[3]$:

> $c23 := (\cos(q3(t))\, m3\, lc3\, l2)\cdot Dq[3]$:

> $g2 := lc3\, m3 \sin\left(-q3(t) - q2(t) + \frac{\pi}{2}\right) g - m3\, g \cos\left(-q2(t) + \frac{\pi}{2}\right) l2 - lc2\, m2 \cos\left(-q2(t) + \frac{\pi}{2}\right) g :$

> $m32 := (lc3\, m3 \sin(q3(t))\, l2 + m3\, lc3^2 + I3zz) :$

> $m33 := (m3\, lc3^2 + I3zz) :$

> $c31 := \left(lc3\, m3\, l1y \cos\left(-q3(t) - q2(t) + \frac{\pi}{2}\right) + \frac{lc3^2\, m3 \sin(-2\, q3(t) - 2\, q2(t) + \pi)}{2} - \frac{lc3\, m3 \cos(q3(t))\, l2}{2} - \frac{lc3\, m3\, l2 \cos(-q3(t) - 2\, q2(t) + \pi)}{2} + \frac{I3yy \sin(-2\, q3(t) - 2\, q2(t) + \pi)}{2} - \frac{I3xx \sin(-2\, q3(t) - 2\, q2(t) + \pi)}{2}\right) \cdot Dq[1] :$

> $c32 := -lc3\, m3 \cos(q3(t))\, l2 \cdot Dq[2] :$

> $g3 := lc3\, m3 \sin\left(-q3(t) - q2(t) + \frac{\pi}{2}\right) g :$

>

> #Setting up the dynamic system

> $M0 := Matrix([[m11, 0, 0], [0, m22, m23], [0, m32, m33]])$

$$M0 := \left[\left[\frac{I3yy \cos(2\, q3(t) + 2\, q2(t))}{2} - \frac{I3xx \cos(2\, q3(t) + 2\, q2(t))}{2} + \frac{I2xx}{2} + m2\, l1y^2 + \frac{m2\, lc2^2}{2} + \frac{m3\, l2^2}{2} + \frac{m3\, lc3^2}{2} + \frac{I2yy}{2} + \frac{I2yy \cos(2\, q2(t))}{2} + I1xx + 2\, m3\, lc3 \cos(q3(t) + q2(t))\, l1y - m3\, lc3\, l2 \sin(q3(t) + 2\, q2(t)) - \frac{I2xx \cos(2\, q2(t))}{2} + lc3\, m3 \sin(q3(t))\, l2 + m1\, lc1y^2 + \frac{I3xx}{2} - \frac{m3\, l2^2 \cos(2\, q2(t))}{2} + m3\, l1y^2 - 2 \sin(q2(t))\, lc2\, m2\, l1y - 2 \sin(q2(t))\, m3\, l1y\, l2 + \frac{I3yy}{2} + \frac{m3\, lc3^2 \cos(2\, q3(t) + 2\, q2(t))}{2} - \frac{m2\, lc2^2 \cos(2\, q2(t))}{2}, 0, 0\right], \left[0, m3\, l2^2 + I3zz + m2\, lc2^2 + 2\, lc3\, m3 \sin(q3(t))\, l2 + m3\, lc3^2 + I2zz, lc3\, m3 \sin(q3(t))\, l2 + m3\, lc3^2 + I3zz\right], \left[0, lc3\, m3 \sin(q3(t))\, l2 + m3\, lc3^2 + I3zz, m3\, lc3^2 + I3zz\right]\right] \qquad \textbf{(25)}$$

> $C0 := Matrix([[c11, 0, 0], [c21, c22, c23], [c31, c32, 0]])$

$$C0 := [[(m3\, l2^2 \sin(2\, q2(t)) - I3yy \sin(2\, q3(t) + 2\, q2(t)) + I2xx \sin(2\, q2(t)) + I3xx \sin(2\, q3(t) + 2\, q2(t)) - I2yy \sin(2\, q2(t)) - lc3^2\, m3 \sin(2\, q3(t) + 2\, q2(t)) - 2\, m3 \cos(q2(t))\, l1y\, l2 - 2\, lc3\, m3\, l2 \cos(q3(t) + 2\, q2(t)) - 2\, lc3\, m3\, l1y \sin(q3(t) + q2(t)) - 2\, lc2\, m2 \cos(q2(t))\, l1y + m2\, lc2^2 \sin(2\, q2(t)))\, \dot{q}2(t) + (lc3\, m3 \cos(q3(t))\, l2 - lc3\, m3\, l2 \cos(q3(t) + 2\, q2(t)) - lc3^2\, m3 \sin(2\, q3(t) + 2\, q2(t)) - I3yy \sin(2\, q3(t) + 2\, q2(t)) + I3xx \sin(2\, q3(t) + 2\, q2(t)) - 2\, lc3\, m3\, l1y \sin(q3(t) + q2(t)))\, \dot{q}3(t), 0, 0],$$
$$\left[\left(lc3\, m3\, l2 \cos(q3(t) + 2\, q2(t)) + lc2\, m2 \cos(q2(t))\, l1y + \frac{I3yy \sin(2\, q3(t) + 2\, q2(t))}{2} - \frac{I3xx \sin(2\, q3(t) + 2\, q2(t))}{2} + lc3\, m3\, l1y \sin(q3(t) + q2(t)) - \frac{m2\, lc2^2 \sin(2\, q2(t))}{2} - \frac{m3\, l2^2 \sin(2\, q2(t))}{2} + m3 \cos(q2(t))\, l1y\, l2 + \frac{lc3^2\, m3 \sin(2\, q3(t) + 2\, q2(t))}{2} + \frac{I2yy \sin(2\, q2(t))}{2} - \frac{I2xx \sin(2\, q2(t))}{2}\right) \dot{q}1(t), 2 \cos(q3(t))\, m3\, lc3\, l2\, \dot{q}3(t), \cos(q3(t))\, m3\, lc3\, l2\, \dot{q}3(t)\right],$$
$$\left[\left(lc3\, m3\, l1y \sin(q3(t) + q2(t)) + \frac{lc3^2\, m3 \sin(2\, q3(t) + 2\, q2(t))}{2} - \frac{lc3\, m3 \cos(q3(t))\, l2}{2} + \frac{lc3\, m3\, l2 \cos(q3(t) + 2\, q2(t))}{2} + \frac{I3yy \sin(2\, q3(t) + 2\, q2(t))}{2} - \frac{I3xx \sin(2\, q3(t) + 2\, q2(t))}{2}\right) \dot{q}1(t), -lc3\, m3 \cos(q3(t))\, l2\, \dot{q}2(t), 0\right]\right] \qquad \textbf{(26)}$$

> $G0 := Vector([[0], [g2], [g3]])$

$$G0 := \begin{bmatrix} 0 \\ lc3\, m3 \cos(q3(t) + q2(t))\, g - m3\, g \sin(q2(t))\, l2 - lc2\, m2 \sin(q2(t))\, g \\ lc3\, m3 \cos(q3(t) + q2(t))\, g \end{bmatrix} \qquad \textbf{(27)}$$

> $M0inv := MatrixInverse(M0) :$

> $DynSys := combine(simplify(MatrixVectorMultiply(M0inv, (-MatrixVectorMultiply(C0, Dq) - G0 + Vector([[u1], [u2], [u3]])))), trig) :$

> $DynSysWithoutGrav := combine(simplify(MatrixVectorMultiply(M0inv, (-MatrixVectorMultiply(C0, Dq) + Vector([[u1], [u2], [u3]])))), trig)$

$$DynSysWithoutGrav := \Big[\Big[\big(-2\, \dot{q}1(t)\, \dot{q}2(t)\, m3\, l2^2 \sin(2\, q2(t)) + 2\, \dot{q}1(t)\, \dot{q}2(t)\, I3yy \sin(2\, q3(t) + 2\, q2(t)) - 2\, \dot{q}1(t)\, \dot{q}2(t)\, I2xx \sin(2\, q2(t)) - 2\, \dot{q}1(t)\, \dot{q}2(t)\, I3xx \sin(2\, q3(t) + 2\, q2(t)) + 2\, \dot{q}1(t)\, \dot{q}2(t)\, I2yy \sin(2\, q2(t)) + 2\, \dot{q}1(t)\, \dot{q}2(t)\, lc3^2\, m3 \sin(2\, q3(t) + 2\, q2(t)) + 4\, \dot{q}1(t)\, \dot{q}2(t)\, m3 \cos(q2(t))\, l1y\, l2 + 4\, \dot{q}1(t)\, \dot{q}2(t)\, lc3\, m3\, l2 \cos(q3(t) + 2\, q2(t)) + 4\, \dot{q}1(t)\, \dot{q}2(t)\, lc3\, m3\, l1y \sin(q3(t) + q2(t)) + 4\, \dot{q}1(t)\, \dot{q}2(t)\, lc2\, m2 \cos(q2(t))\, l1y - 2\, \dot{q}1(t)\, \dot{q}2(t)\, m2\, lc2^2 \sin(2\, q2(t)) - 2\, m3\, lc3\, \dot{q}1(t)\, \dot{q}3(t)\, l2 \cos(q3(t)) + 2\, \dot{q}1(t)\, \dot{q}3(t)\, lc3\, m3\, l2 \cos(q3(t) + 2\, q2(t)) + 2\, \dot{q}1(t)\, \dot{q}3(t)\, lc3^2\, m3 \sin(2\, q3(t) + 2\, q2(t)) + 2\, \dot{q}1(t)\, \dot{q}3(t)\, I3yy \sin(2\, q3(t) + 2\, q2(t)) - 2\, \dot{q}1(t)\, \dot{q}3(t)\, I3xx \sin(2\, q3(t) + 2\, q2(t)) + 4\, \dot{q}1(t)\, \dot{q}3(t)\, lc3\, m3\, l1y \sin(q3(t) + q2(t)) + 2\, u1\big) \Big/ \big(I3yy \cos(2\, q3(t) + 2\, q2(t)) - I3xx \cos(2\, q3(t) + 2\, q2(t)) + I2xx + 2\, m2\, l1y^2 + m2\, lc2^2 + m3\, l2^2 + m3\, lc3^2 + I2yy + I2yy \cos(2\, q2(t)) + 2\, I1xx + 4\, m3\, lc3 \cos(q3(t) + q2(t))\, l1y - 2\, m3\, lc3\, l2 \sin(q3(t) + 2\, q2(t)) - I2xx \cos(2\, q2(t)) + 2\, lc3\, m3 \sin(q3(t))\, l2 + 2\, m1\, lc1y^2 + I3xx - m3\, l2^2 \cos(2\, q2(t)) + 2\, m3\, l1y^2 - 4 \sin(q2(t))\, lc2\, m2\, l1y - 4 \sin(q2(t))\, m3\, l1y\, l2 + I3yy + m3\, lc3^2 \cos(2\, q3(t) + 2\, q2(t)) - m2\, lc2^2 \cos(2\, q2(t))\big)\Big], \qquad \textbf{(28)}$$
$$\Big[\big(2\, m3\, lc3^2\, \dot{q}1(t)^2\, m2\, lc2^2 \sin(2\, q2(t)) - 2\, m3^2\, lc3^2\, \dot{q}1(t)^2 \cos(q2(t))\, l1y\, l2 - 8\, m3^2\, lc3^3 \cos(q3(t))\, \dot{q}2(t)\, \dot{q}3(t)\, l2$$

$$-2\, I3zz\, \dot{q1}(t)^2\, lc3\, m3\, l2 \cos(q3(t)+2\, q2(t)) - 4\, I3zz\, \dot{q1}(t)^2\, lc2\, m2 \cos(q2(t))\, l1y - 4\, I3zz\, \dot{q1}(t)^2\, m3 \cos(q2(t))\, l1y\, l2$$
$$-4\, I3zz \cos(q3(t))\, m3\, lc3\, \dot{q3}(t)^2\, l2 - 2\, I3zz\, \dot{q1}(t)^2\, lc3\, m3 \cos(q3(t))\, l2 - 4\, I3zz\, lc3\, m3 \cos(q3(t))\, \dot{q2}(t)^2\, l2$$
$$-4\, m3\, lc3^2\, \dot{q1}(t)^2\, lc2\, m2 \cos(q2(t))\, l1y - 8\, I3zz \cos(q3(t))\, m3\, lc3\, \dot{q2}(t)\, \dot{q3}(t)\, l2 + lc3\, m3\, l2\, \dot{q1}(t)^2\, I3yy \cos(q3(t)+2\, q2(t))$$
$$-lc3\, m3\, l2\, \dot{q1}(t)^2\, I3yy \cos(3\, q3(t)+2\, q2(t)) - lc3\, m3\, l2\, \dot{q1}(t)^2\, I3xx \cos(q3(t)+2\, q2(t)) + lc3\, m3\, l2\, \dot{q1}(t)^2\, I3xx \cos(3\, q3(t)+2\, q2(t))$$
$$-2\, lc3^2\, m3^2\, l2\, \dot{q1}(t)^2\, l1y \cos(2\, q3(t)+q2(t)) + 4\, I3zz\, u2 - 4\, I3zz\, u3 + 4\, m3\, lc3^2\, u2 - 4\, m3\, lc3^2\, u3 - m3^2\, lc3^3\, \dot{q1}(t)^2\, l2 \cos(q3(t)+2\, q2(t))$$
$$+m3^2\, lc3^2\, \dot{q1}(t)^2\, l2^2 \sin(2\, q2(t)) - 2\, m3\, lc3^2\, \dot{q1}(t)^2\, I2yy \sin(2\, q2(t)) + 2\, m3\, lc3^2\, \dot{q1}(t)^2\, I2xx \sin(2\, q2(t)) - 4\, m3^2\, lc3^3 \cos(q3(t))\, \dot{q3}(t)^2\, l2$$
$$+2\, I3zz\, \dot{q1}(t)^2\, m2\, lc2^2 \sin(2\, q2(t)) + 2\, I3zz\, \dot{q1}(t)^2\, m3\, l2^2 \sin(2\, q2(t)) + m3^2\, l2^2\, \dot{q1}(t)^2\, lc3^2 \sin(2\, q3(t)+2\, q2(t)) - 4\, lc3\, m3 \sin(q3(t))\, l2\, u3$$
$$-2\, m3^2\, lc3^3\, \dot{q1}(t)^2 \cos(q3(t))\, l2 - 4\, m3^2\, lc3^3 \cos(q3(t))\, \dot{q2}(t)^2\, l2 - lc3^3\, m3^2\, l2\, \dot{q1}(t)^2 \cos(3\, q3(t)+2\, q2(t)) - 2\, I3zz\, \dot{q1}(t)^2\, I2yy \sin(2\, q2(t))$$
$$+2\, I3zz\, \dot{q1}(t)^2\, I2xx \sin(2\, q2(t)) - lc3^2\, m3^2\, l2^2\, \dot{q1}(t)^2 \sin(2\, q3(t)) - 2\, lc3^2\, m3^2\, l2^2\, \dot{q2}(t)^2 \sin(2\, q3(t)) \Big) \Big/ \Big(2\, m3^2\, l2^2\, lc3^2 + 4\, m3\, l2^2\, I3zz$$
$$+4\, m2\, lc2^2\, m3\, lc3^2 + 4\, m2\, lc2^2\, I3zz + 4\, I2zz\, m3\, lc3^2 + 4\, I2zz\, I3zz + 2\, m3^2\, l2^2\, lc3^2 \cos(2\, q3(t)) \Big) \Big],$$
$$\Big[\Big(4\, lc3^2\, m3^2\, l2^2\, \dot{q2}(t)\, \dot{q3}(t) \sin(2\, q3(t)) - lc3\, m3\, l2\, \dot{q1}(t)^2\, m2\, lc2^2 \cos(-q3(t)+2\, q2(t)) - 2\, m3\, lc3^2\, \dot{q1}(t)^2\, m2\, lc2^2 \sin(2\, q2(t))$$
$$+2\, m3^2\, lc3^2\, \dot{q1}(t)^2 \cos(q2(t))\, l1y\, l2 + 8\, m3^2\, lc3^3 \cos(q3(t))\, \dot{q2}(t)\, \dot{q3}(t)\, l2 + 2\, I3zz\, \dot{q1}(t)^2\, lc3\, m3\, l2 \cos(q3(t)+2\, q2(t))$$
$$+4\, I3zz\, \dot{q1}(t)^2\, lc2\, m2 \cos(q2(t))\, l1y + 4\, I3zz\, \dot{q1}(t)^2\, m3 \cos(q2(t))\, l1y\, l2 + 4\, I3zz \cos(q3(t))\, m3\, lc3\, \dot{q3}(t)^2\, l2$$
$$-2\, m3^2\, l2^2\, \dot{q1}(t)^2\, lc3\, l1y \sin(q3(t)+q2(t)) + 2\, I3zz\, \dot{q1}(t)^2\, lc3\, m3 \cos(q3(t))\, l2 + 4\, I3zz\, lc3\, m3 \cos(q3(t))\, \dot{q2}(t)^2\, l2$$
$$-2\, m2\, lc2^2\, \dot{q1}(t)^2\, lc3^2\, m3 \sin(2\, q3(t)+2\, q2(t)) - 4\, I2zz\, \dot{q1}(t)^2\, lc3\, m3\, l1y \sin(q3(t)+q2(t)) + 2\, I2zz\, \dot{q1}(t)^2\, lc3\, m3 \cos(q3(t))\, l2$$
$$-2\, I2zz\, \dot{q1}(t)^2\, lc3\, m3\, l2 \cos(q3(t)+2\, q2(t)) + 4\, I2zz\, lc3\, m3 \cos(q3(t))\, \dot{q2}(t)^2\, l2 + 4\, m3\, lc3^2\, \dot{q1}(t)^2\, lc2\, m2 \cos(q2(t))\, l1y$$
$$+8\, I3zz \cos(q3(t))\, m3\, lc3\, \dot{q2}(t)\, \dot{q3}(t)\, l2 - 4\, m2\, lc2^2\, \dot{q1}(t)^2\, lc3\, m3\, l1y \sin(q3(t)+q2(t)) + 2\, m2\, lc2^2\, \dot{q1}(t)^2\, lc3\, m3 \cos(q3(t))\, l2$$
$$-m2\, lc2^2\, \dot{q1}(t)^2\, lc3\, m3\, l2 \cos(q3(t)+2\, q2(t)) + 4\, m2\, lc2^2\, lc3\, m3 \cos(q3(t))\, \dot{q2}(t)^2\, l2 + 2\, lc3^2\, m3^2\, l2^2\, \dot{q3}(t)^2 \sin(2\, q3(t))$$
$$-lc3\, m3\, l2\, \dot{q1}(t)^2\, I3yy \cos(q3(t)+2\, q2(t)) + lc3\, m3\, l2\, \dot{q1}(t)^2\, I3yy \cos(3\, q3(t)+2\, q2(t)) + lc3\, m3\, l2\, \dot{q1}(t)^2\, I3xx \cos(q3(t)+2\, q2(t))$$
$$-lc3\, m3\, l2\, \dot{q1}(t)^2\, I3xx \cos(3\, q3(t)+2\, q2(t)) + 2\, lc3^2\, m3^2\, l2\, \dot{q1}(t)^2\, l1y \cos(2\, q3(t)+q2(t)) - 2\, lc3\, m3^2\, l2^2\, \dot{q1}(t)^2\, l1y \sin(-q3(t)+q2(t))$$
$$+lc3\, m3\, l2\, \dot{q1}(t)^2\, I2yy \cos(-q3(t)+2\, q2(t)) - lc3\, m3\, l2\, \dot{q1}(t)^2\, I2yy \cos(q3(t)+2\, q2(t)) - lc3\, m3\, l2\, \dot{q1}(t)^2\, I2xx \cos(-q3(t)+2\, q2(t))$$
$$+lc3\, m3\, l2\, \dot{q1}(t)^2\, I2xx \cos(q3(t)+2\, q2(t)) - 4\, I3zz\, u2 + 4\, I3zz\, u3 + 4\, I2zz\, u3 - 4\, m3\, lc3^2\, u2 + 4\, m3\, l2^2\, u3 + 4\, m2\, lc2^2\, u3 + 4\, m3\, lc3^2\, u3$$
$$-4\, lc3\, m3 \sin(q3(t))\, l2\, u2 + m3^2\, lc3^3\, \dot{q1}(t)^2\, l2 \cos(q3(t)+2\, q2(t)) - 2\, m3^2\, lc3^2\, \dot{q1}(t)^2\, l2^2 \sin(2\, q2(t)) + 2\, m3\, lc3^2\, \dot{q1}(t)^2\, I2yy \sin(2\, q2(t))$$
$$-2\, m3\, lc3^2\, \dot{q1}(t)^2\, I2xx \sin(2\, q2(t)) + 4\, m3^2\, lc3^3 \cos(q3(t))\, \dot{q3}(t)^2\, l2 - 2\, I3zz\, \dot{q1}(t)^2\, m2\, lc2^2 \sin(2\, q2(t)) - 2\, I3zz\, \dot{q1}(t)^2\, m3\, l2^2 \sin(2\, q2(t))$$
$$-2\, m3^2\, l2^2\, \dot{q1}(t)^2\, lc3^2 \sin(2\, q3(t)+2\, q2(t)) + 2\, m3^2\, l2^3\, \dot{q1}(t)^2\, lc3 \cos(q3(t)) - m3^2\, l2^3\, \dot{q1}(t)^2\, lc3 \cos(q3(t)+2\, q2(t))$$
$$-2\, m3\, l2^2\, \dot{q1}(t)^2\, I3yy \sin(2\, q3(t)+2\, q2(t)) + 2\, m3\, l2^2\, \dot{q1}(t)^2\, I3xx \sin(2\, q3(t)+2\, q2(t)) + 4\, m3^2\, l2^3\, lc3 \cos(q3(t))\, \dot{q2}(t)^2$$
$$-2\, m2\, lc2^2\, \dot{q1}(t)^2\, I3yy \sin(2\, q3(t)+2\, q2(t)) + 2\, m2\, lc2^2\, \dot{q1}(t)^2\, I3xx \sin(2\, q3(t)+2\, q2(t)) + 8\, lc3\, m3 \sin(q3(t))\, l2\, u3$$
$$+2\, m3^2\, lc3^3\, \dot{q1}(t)^2 \cos(q3(t))\, l2 + 4\, m3^2\, lc3^3 \cos(q3(t))\, \dot{q2}(t)^2\, l2 - 2\, I2zz\, \dot{q1}(t)^2\, lc3^2\, m3 \sin(2\, q3(t)+2\, q2(t)) - lc3\, m3^2\, l2^3\, \dot{q1}(t)^2 \cos($$
$$-q3(t)+2\, q2(t)) + lc3^3\, m3^2\, l2\, \dot{q1}(t)^2 \cos(3\, q3(t)+2\, q2(t)) + 2\, I3zz\, \dot{q1}(t)^2\, I2yy \sin(2\, q2(t)) - 2\, I3zz\, \dot{q1}(t)^2\, I2xx \sin(2\, q2(t))$$
$$-2\, I2zz\, \dot{q1}(t)^2\, I3yy \sin(2\, q3(t)+2\, q2(t)) + 2\, I2zz\, \dot{q1}(t)^2\, I3xx \sin(2\, q3(t)+2\, q2(t)) + 2\, lc3^2\, m3^2\, l2^2\, \dot{q1}(t)^2 \sin(2\, q3(t))$$
$$+4\, lc3^2\, m3^2\, l2^2\, \dot{q2}(t)^2 \sin(2\, q3(t)) + 2\, lc3\, m3\, l2\, \dot{q1}(t)^2\, lc2\, m2\, l1y \sin(q3(t)+q2(t)) - 2\, lc3\, m3\, l2\, \dot{q1}(t)^2\, lc2\, m2\, l1y \sin(-q3(t)+q2(t)) \Big) \Big/$$
$$\Big(2\, m3^2\, l2^2\, lc3^2 + 4\, m3\, l2^2\, I3zz + 4\, m2\, lc2^2\, m3\, lc3^2 + 4\, m2\, lc2^2\, I3zz + 4\, I2zz\, m3\, lc3^2 + 4\, I2zz\, I3zz + 2\, m3^2\, l2^2\, lc3^2 \cos(2\, q3(t)) \Big) \Big] \Big]$$

```
>
> #Computing the inertias
```

$$> \; I1xxx := int\left((y^2+z^2) \cdot \rho 1, \left[x = -\frac{l1x}{2} .. \frac{l1x}{2}, y = -\frac{w1}{2} .. \frac{w1}{2}, z = -\frac{w1}{2} .. \frac{w1}{2} \right] \right)$$

$$I1xxx := \frac{\rho 1\, l1x\, w1^4}{6} \tag{29}$$

$$> \; I1yyy := int\left((x^2+z^2) \cdot \rho 1, \left[x = -\frac{l1x}{2} .. \frac{l1x}{2}, y = -\frac{w1}{2} .. \frac{w1}{2}, z = -\frac{w1}{2} .. \frac{w1}{2} \right] \right)$$

$$I1yyy := \rho 1\, w1 \left(\frac{1}{12}\, l1x^3\, w1 + \frac{1}{12}\, l1x\, w1^3 \right) \tag{30}$$

$$> \; I1zzz := int\left((y^2+x^2) \cdot \rho 1, \left[x = -\frac{l1x}{2} .. \frac{l1x}{2}, y = -\frac{w1}{2} .. \frac{w1}{2}, z = -\frac{w1}{2} .. \frac{w1}{2} \right] \right)$$

$$I1zzz := \rho 1\, w1 \left(\frac{1}{12}\, l1x^3\, w1 + \frac{1}{12}\, l1x\, w1^3 \right) \tag{31}$$

$$> \; I2xxx := int\left((y^2+z^2) \cdot \rho 2, \left[x = -\frac{w2}{2} .. \frac{w2}{2}, y = -\frac{l2}{2} .. \frac{l2}{2}, z = -\frac{w2}{2} .. \frac{w2}{2} \right] \right)$$

$$I2xxx := \rho 2\, w2 \left(\frac{1}{12}\, l2^3\, w2 + \frac{1}{12}\, l2\, w2^3 \right) \tag{32}$$

> $I2yyy := int\left((x^2 + z^2) \cdot \rho 2, \left[x = -\frac{w2}{2} .. \frac{w2}{2}, y = -\frac{l2}{2} .. \frac{l2}{2}, z = -\frac{w2}{2} .. \frac{w2}{2} \right] \right)$

$$I2yyy := \frac{\rho 2\, l2\, w2^4}{6} \tag{33}$$

> $I2zzz := int\left((y^2 + x^2) \cdot \rho 2, \left[x = -\frac{w2}{2} .. \frac{w2}{2}, y = -\frac{l2}{2} .. \frac{l2}{2}, z = -\frac{w2}{2} .. \frac{w2}{2} \right] \right)$

$$I2zzz := \rho 2\, w2 \left(\frac{1}{12}\, l2^3\, w2 + \frac{1}{12}\, l2\, w2^3 \right) \tag{34}$$

> $I3xxx := int\left((y^2 + z^2) \cdot \rho 3, \left[x = -\frac{l3}{2} .. \frac{l3}{2}, y = -\frac{w3}{2} .. \frac{w3}{2}, z = -\frac{w3}{2} .. \frac{w3}{2} \right] \right)$

$$I3xxx := \frac{\rho 3\, l3\, w3^4}{6} \tag{35}$$

> $I3yyy := int\left((x^2 + z^2) \cdot \rho 3, \left[x = -\frac{l3}{2} .. \frac{l3}{2}, y = -\frac{w3}{2} .. \frac{w3}{2}, z = -\frac{w3}{2} .. \frac{w3}{2} \right] \right)$

$$I3yyy := \rho 3\, w3 \left(\frac{1}{12}\, l3^3\, w3 + \frac{1}{12}\, l3\, w3^3 \right) \tag{36}$$

> $I3zzz := int\left((y^2 + x^2) \cdot \rho 3, \left[x = -\frac{l3}{2} .. \frac{l3}{2}, y = -\frac{w3}{2} .. \frac{w3}{2}, z = -\frac{w3}{2} .. \frac{w3}{2} \right] \right)$

$$I3zzz := \rho 3\, w3 \left(\frac{1}{12}\, l3^3\, w3 + \frac{1}{12}\, l3\, w3^3 \right) \tag{37}$$

```
>
> #Converting equations to Matlab code to simplify Matlab input
> with(CodeGeneration) :
> dx2 := Matlab(DynSys(1), resultname = "dx2") :
> dx4 := Matlab(DynSys(2), resultname = "dx4")
> dx6 := Matlab(DynSys(3), resultname = "dx6")
> dx2 := Matlab(DynSysWithoutGrav(1), resultname = "dx2") :
> dx4 := Matlab(DynSysWithoutGrav(2), resultname = "dx4") :
> dx6 := Matlab(DynSysWithoutGrav(3), resultname = "dx6") :
>
```

> $Ekin := \frac{1}{2} \cdot VectorMatrixMultiply(Transpose(Dq), MatrixVectorMultiply(M0, Dq))$

$$\begin{aligned} Ekin := \frac{1}{2} \Bigg(\Bigg(& \frac{I3yy \cos(2\, q3(t) + 2\, q2(t))}{2} - \frac{I3xx \cos(2\, q3(t) + 2\, q2(t))}{2} + \frac{I2xx}{2} + m2\, l1y^2 + \frac{m2\, lc2^2}{2} + \frac{m3\, l2^2}{2} + \frac{m3\, lc3^2}{2} + \frac{I2yy}{2} \\ & + \frac{I2yy \cos(2\, q2(t))}{2} + I1xx + 2\, m3\, lc3 \cos(q3(t) + q2(t))\, l1y - m3\, lc3\, l2 \sin(q3(t) + 2\, q2(t)) - \frac{I2xx \cos(2\, q2(t))}{2} + lc3\, m3 \sin(q3(t))\, l2 \\ & + m1\, lc1y^2 + \frac{I3xx}{2} - \frac{m3\, l2^2 \cos(2\, q2(t))}{2} + m3\, l1y^2 - 2 \sin(q2(t))\, lc2\, m2\, l1y - 2 \sin(q2(t))\, m3\, l1y\, l2 + \frac{I3yy}{2} \\ & + \frac{m3\, lc3^2 \cos(2\, q3(t) + 2\, q2(t))}{2} - \frac{m2\, lc2^2 \cos(2\, q2(t))}{2} \Bigg) \dot{q1}(t)^2 \Bigg) \\ & + \frac{\left((m3\, l2^2 + I3zz + m2\, lc2^2 + 2\, lc3\, m3 \sin(q3(t))\, l2 + m3\, lc3^2 + I2zz)\, \dot{q2}(t) + (lc3\, m3 \sin(q3(t))\, l2 + m3\, lc3^2 + I3zz)\, \dot{q3}(t) \right) \dot{q2}(t)}{2} \\ & + \frac{\left((lc3\, m3 \sin(q3(t))\, l2 + m3\, lc3^2 + I3zz)\, \dot{q2}(t) + (m3\, lc3^2 + I3zz)\, \dot{q3}(t) \right) \dot{q3}(t)}{2} \end{aligned} \tag{38}$$

```
>
```

Bibliography

[1] Horsepower and Torque. `http://craig.backfire.ca/pages/autos/horsepower`.

[2] What are AC motors? `http://www.wisegeek.com/what-are-ac-motors.htm`.

[3] Chaotic Pendulums. `http://www.chaoticpendulums.com/chaos-theory-a9.html`, 2007.

[4] ABB. IRB-140-6-0p8-data, 2004.

[5] J.G. Balchen, T. Andresen, and B.A. Foss. *Reguleringsteknikk.* Institutt for teknisk kybernetikk, 2004.

[6] 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.

[7] J.M. Hollerbach. A Recursive Lagrangian Formulation of Maniputator 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] L. Sciavicco and B. Siciliano. *Modelling and control of robot manipulators.* Springer Verlag, 2000.

[10] T. Shinbrot, C. Grebogi, J. Wisdom, and J.A. Yorke. Chaos in a double pendulum. *American Journal of Physics*, 60:491, 1992.

[11] W.M. Silver. On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators. *The International Journal of Robotics Research*, 1(2):60, 1982.

[12] M.W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control.* Wiley New Jersey, 2006.

[13] M.W. Spong and M. Vidyasagar. *Robot dynamics and control.* Wiley New Jersey, 1989.

[14] J.R. Taylor. *Classical mechanics.* Univ Science Books, 2005.