System analysis with Matlab

1

M.Sc. Laboratory Advanced Control (WS 22/23)

Prof. Dr.–Ing. habil. Thomas Meurer, Chair of Automation and Control

This experiment aims to familiarize yourself with the use of the computer algebra system of Matlab for **mathematical modeling** and **system analysis**. While Matlab is an proprietary desktop environment predominantly used for numeric computing, optional toolboxes extend the range of application. To derive equations of motion it is necessary to perform symbolic calculations. For this purpose Matlab provides the Symbolic Math Toolbox. For the processing of the preparation tasks, it is necessary to install the program and the necessary toolbox on your own computer.

When submitting the solution of the preparatory tasks, please **ensure that the same parameter variables defined in the respective task are used**. Please also make sure that all tasks are well structured and summarized in a single file. For this purpose, divide the tasks into subtasks and use individual code blocks with headings for each task.

If you have any questions or suggestions regarding this experiment, please contact:

Sönke Bartels - sba@tf.uni-kiel.de

1.1 Basic Matlab commands

Next, some basic commands of Matlab in combination with the Matlab Symbolic Toolbox are described. If you are already familiar with Matlab Symbolic Math Toolbox, you can continue with Section 1.2. In addition to the program help of Matlab, further documentation can be found on the Matlab-website.

While Matlab files are generally marked by the extension `*.m`, performing symbolic calculations and printouts it is recommended to use the Live-Script capability (*.mlx). In the file `symbolic_basics.mlx` some basic commands and the usage of Matlab are introduced.

> **Exercise 1.1.** *Open the file* `symbolic_basics.mlx` *with Matlab and work through all commands. To execute a command, place the cursor in the respective input cell, then press the ctrl key together with the Enter key. The key F5 executes a script completely. Try to understand all commands and, if necessary, use the help function and the above mentioned references.*

1.2 Simple calculations with Maxima

Create a new Matlab-Livescript (*.mlx) and work on the following tasks (don't forget to add meaningful headings within your script):

Exercise 1.2.

(i) *Solve the system of linear equations*

$$\begin{aligned} 5x_1 + 3x_2 \qquad &= 10 \\ 2x_1 + 4x_2 + \ \ x_3 &= 6 \\ 2x_1 \qquad + 4x_3 &= 2\,. \end{aligned} \tag{1.1}$$

by rearranging the equation and using the command `linsolve()`. *Analyze if and how many solutions exist.*

(ii) *Represent the system as a set of linear symbolic equations by creating the symbols x_1, x_2, x_3 and solve the system using* `solve()`. *Compare the solutions.*

Exercise 1.3.

(i) *How many solutions has the system of linear equations*

$$\begin{aligned} x_2 + 3x_3 &= 0 \\ 2x_1 + 4x_2 \qquad &= 0 \\ -2x_2 - 6x_3 &= 0\,. \end{aligned} \tag{1.2}$$

Write a short explanation.

(ii) *Try to calculate the solution using* `linsolve()`.

Exercise 1.4.

(i) *Solve the equation*

$$\begin{bmatrix} 4 & 1 & 2 \\ -2 & 2 & 1 \\ 4 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -3 \\ 2 \\ 2 \end{bmatrix}. \tag{1.3}$$

(ii) *Transfer the matrix equation into a system of linear equations and solve it by the use of its inverse. Matlab offers the functionality of the backslash-operator "\". Look up, what this operator does and use it to solve the equation. Write a short comment on that in your file.*

1.3 Ordinary differential equations

Consider the ordinary differential equation

$$\frac{1}{2}\ddot{y} + 2\alpha\dot{y} + 4y = \gamma \tag{1.4}$$

with the arbitrary but constant inhomogeneity γ and constant parameter $\alpha \in (0,1)$.

Exercise 1.5.

(i) *Construct the differential equation* (1.4) *as a symbolic equation.*

(ii) *Solve the differential equation* (1.4) *by using* `dsolve()`. *Comment on the stability of the system based on the zeros of the characteristic polynomial.*

(iii) *Introduce suitable state variables and transfer the differential equation (1.4) into state space representation with input γ and output $y(t)$. Using the eigenvalues of the system matrix, analyze the stability of the system. Calculate the general solution of the system, e.g., using the solution formula for linear, time-invariant systems (see script for the lecture Control Systems [1], Theorem 3.1).*

(iv) *Determine the numerical solution for the parameter values $\alpha = 4/7$, $\gamma = -0.8$ and the initial conditions $y(0) = 1$, $\dot{y}(0) = 1$. Plot the solution trajectory $t \mapsto [y(t), \dot{y}(t), t]$ for times $t \in [0, 25]$.*

Self-check: At $t = 10$ the solution should take the value $y(10) \approx -0.2$.

Symbol	Maxima variable
α	`alpha`
γ	`gamma`
$y(0)$	`y0`
$\dot{y}(0)$	`dot_y0`

Table 1.1: Matlab nomenclature for Exercise 1.5.

1.4 Laplace transformation

With the help of the Laplace transformation, it is possible to assign functions of time t to functions in complex variables $s = \alpha + i\omega$. This transformation and the corresponding inverse transformation are frequently used in control engineering. Matlab already offers corresponding commands for these operations. The command `laplace(fkt)` provides the Laplace transformation of the function `fkt` from the time domain to the complex-valued image domain. The inverse transformation is generated with `ilaplace(fkt)`.

Exercise 1.6.

(i) *Transform the time function*

$$f(t) = \cos(3\omega t) + t^3 e^{-at} \tag{1.5}$$

to the s-domain of the Laplace–Transformation.

(ii) *Transform the function*

$$\hat{f}(s) = \frac{\omega}{(s+a)^2 + \omega^4} + \frac{7}{s} \tag{1.6}$$

to the time domain.

1.5 Modeling of a mechanical system

In the following, the Lagrange formalism for the mathematical modeling of rigid body systems is introduced using the example of a tower crane according to [2]. Figure 1.1 shows a simplified representation of the considered tower crane. All calculations shown in this section are executed in the Matlab file `lagrange_ex_tower_crane.mlx`, so that the results of the calculations can be traced in the Matlab-file and are only briefly if at all specified in the following section.

An advantage of the Lagrange formalism is that a complex problem is split into less complex subproblems. In the case of the tower crane, the crane is divided into three units, which are analyzed step by step. Subsequently, the subsystems are identified by the indices TA for the tower-boom unit, W for the car unit and M for the mass unit. The crane has five degrees of freedom. The position of the carriage l_W on the jib and the position of the mass m can be varied with the help of two translatory drives using the rope length l_S. In addition, the tower of the crane can be rotated by a rotary drive by the angle φ_{z_i} around the z_i-axis. The mass can be deflected by the angles φ_{x_t} and φ_{y_t} around the x_t- and y_t-axis, respectively.

A rotational movement, as it occurs for the tower and the deflection of the mass, is described by the rotational deflection of two coordinate systems to each other. In the case of the rotating tower crane, the left graphic of Figure 1.1 shows that three coordinate systems are required. For the sake of clarity, one coordinate system is not shown in the right-hand diagram.

First a fixed reference coordinate system, in the following called inertial system, is defined. The origin of this system lies in the center of the crane foundation (see Fig 1.1 x_i, y_i, z_i). The alignment of the axes does not change, which is why this system is the most suitable reference system in most cases.

In the following, two body-fixed coordinate systems are defined, which are fixed to the geometry of the corresponding body so that their axes follow the movements of the body with respect to the inertial system.

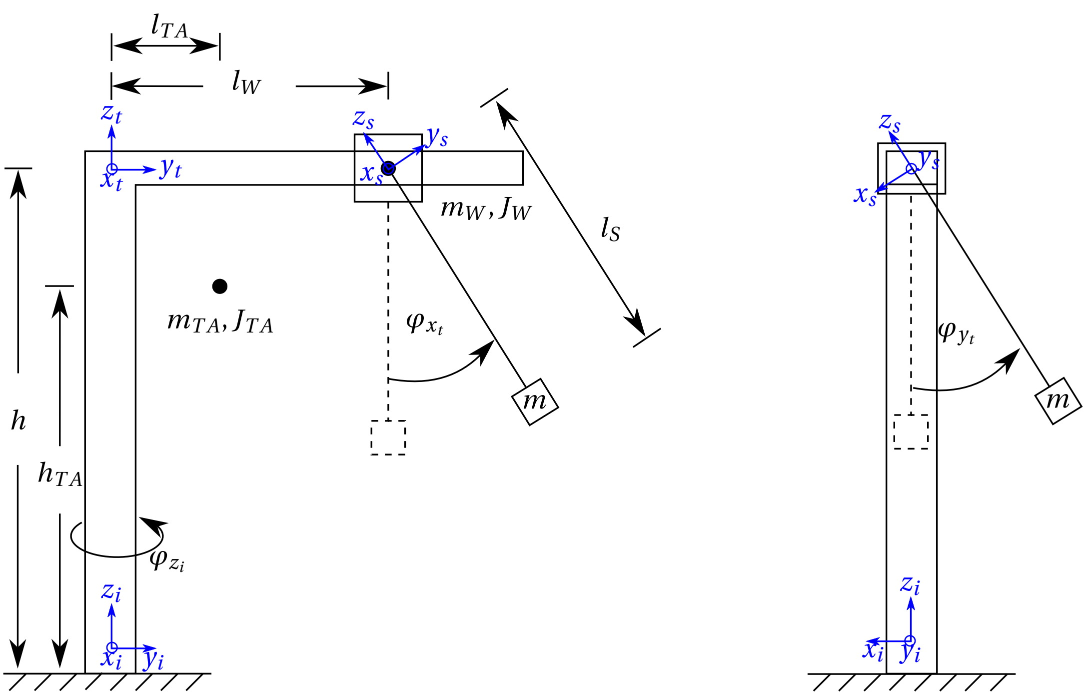

Figure 1.1: Simplified representation of a tower crane.

The first subsystem is the tower system, which is shifted with respect to the inertial system by the height h along the z_i-axis. The alignment of the z_t-axis is identical to the z_i-axis, while the y_t-axis is aligned along the longitudinal axis of the jib. The corresponding x_t-axis is oriented right-handed to the $y_t z_t$-plane. This system is firmly connected to the tower and the jib of the crane, i.e. it moves along with the rotation of the tower by the φ_{z_i} angle. The second system is called the rope system. Its origin is in the center of the carriage, where the winch of the rope is located. The rope is assumed to be massless. If the system is at rest (dashed position of the mass), the tower and rope system have identical orientation.

To determine the equations of motion, any motion must be expressed in the same coordinate system, here the inertial system. To transfer the position of a point related to a certain coordinate system (e.g. the tower system) into the coordinates of another coordinate system (e.g. the inertial system), the distance of the coordinate origins to each other and the rotation between the coordinate systems must be considered.

Let the two coordinate systems 1 and 2 be aligned identically (no rotation) and assume that the origins of the coordinate systems 1 and 2 are displaced by the vector d_1, i.e., only translations are conisdered. Now let

$$\boldsymbol{p}_1 = \boldsymbol{d}_1 + \boldsymbol{p}_2 \quad , \tag{1.7}$$

then $\boldsymbol{p}_k$ represents the position vector of a point in the coordinate system $k \in \{1, 2\}$. If the coordinate systems have the same origin but are rotated to each other, the following applies

$$\boldsymbol{p}_1 = R_1^2 \boldsymbol{p}_2 \quad . \tag{1.8}$$

The matrix R_1^2 is called the rotation matrix. It immediately becomes apparent that joint translation and rotation are represented by

$$\boldsymbol{p}_1 = \boldsymbol{d}_1 + R_1^2 \boldsymbol{p}_2 \quad . \tag{1.9}$$

The rotation matrix can be formed by combining elementary rotation matrices that describe a rotation around an individual axis. A uniform rotation sequence should be maintained, since matrix multiplication is not commutative. For example, if the sequence $x \to y \to z$ is selected, the rotation matrix R_1^2 is calculated as follows:

$$R_1^2 = R_z R_y R_x \quad , \tag{1.10}$$

where

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_x) & -\sin(\varphi_x) \\ 0 & \sin(\varphi_x) & \cos(\varphi_x) \end{bmatrix}, \; R_y = \begin{bmatrix} \cos(\varphi_y) & 0 & -\sin(\varphi_y) \\ 0 & 1 & 0 \\ \sin(\varphi_y) & 0 & \cos(\varphi_y) \end{bmatrix}, \tag{1.11}$$

$$R_z = \begin{bmatrix} \cos(\varphi_z) & \sin(\varphi_z) & 0 \\ -\sin(\varphi_z) & \cos(\varphi_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The R_x matrix represents the rotation around the x-axis of the original coordinate system through the angle φ_x. The resulting new "auxiliary" coordinate system is rotated around the y-axis, which is represented by the matrix R_y at angle φ_y. The sequence is completed by the rotation around the

z-axis of the previously created "auxiliary" coordinate system, in terms of the matrix R_z and the angle φ_z. The matrix R_2^1 describes the rotation from coordinate 1 to 2. If the coordinate representation of a vector in system 2 is to be transformed to system 1 as in (1.9), then the inverse of the rotation matrix is needed. The inverse can be determined by exploiting the orthogonality of the rotation matrix, which yields

$$R_1^2 = \left(R_2^1\right)^{-1} = \left(R_2^1\right)^T \quad . \tag{1.12}$$

With these equations it is now possible to transform a vector into different coordinate systems (also called frames), which is necessary for the Lagrange formalism.

1.5.1 Lagrange equations of 2nd kind

In the following, the derivation of the equations of motion using the Lagrange formalism is briefly explained and is illustrated for the example of the tower crane. The Lagrange formalism leads to the Euler-Lagrange equations and is particularly suitable for the mathematical modeling of rigid body systems. First n generalized coordinates are defined

$$\boldsymbol{q} = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}^T \quad ,$$

that determine the degrees of freedom of the system under consideration together with the associated n generalized forces

$$\boldsymbol{Q} = \begin{bmatrix} Q_1 & \cdots & Q_n \end{bmatrix}^T \quad .$$

Next the Lagrange function

$$L = W_{\text{kin}} - W_{\text{pot}} \tag{1.13}$$

is introduced, which describes the difference between the kinetic energy stored in the system and the potential energy. The dynamics of a non-dissipative system then follows from the Euler-Lagrange equation according to

$$[illegible] - \frac{\partial L}{\partial q_i} [illegible] \quad i = 1, \cdots, n \quad . \tag{1.14}$$

If [illegible] system is considered, then this equation must be extended to consider the dissipative effect of the damping elements. In the case of pure velocity proportional friction or damping terms, the so-called Rayleigh dissipation function R is formed, i.e.

$$R = \frac{1}{2} \sum_{j=1}^{n} k_j \dot{q}_j^2, \quad k_j > 0 \quad . \tag{1.15}$$

In this case the Euler-Lagrange equations read

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} [illegible] , \qquad i = 1, \cdots, n \quad . \tag{1.16}$$

Example: Tower crane
In the following the Lagrange formalism is applied to the example of the tower crane shown in Figure 1.1. In the wxMatlab file `Lagrange_ex_tower_crane.mlx` the individual calculations can be traced in detail. In the first step, the centers of gravity of the individual units (tower and jib, car, mass) are transferred to the inertial system. This is necessary to calculate the kinetic and potential energy of

the whole system. The energies must be calculated in a uniform coordinate system depending on the generalized coordinates $\boldsymbol{q} = [l_W \; l_S \; \varphi_{zi} \; \varphi_{xt} \; \varphi_{yt}]^T$ must be expressed. Following the brief introduction above, for the tower crane we obtain It holds:

$$\begin{aligned} \boldsymbol{s}_{TAi}(\boldsymbol{q}) &= \boldsymbol{d}_i^t(\boldsymbol{q}) + R_i^t(\boldsymbol{q})\boldsymbol{s}_{TAt}(\boldsymbol{q}), \\ \boldsymbol{s}_{Wi}(\boldsymbol{q}) &= \boldsymbol{d}_i^t(\boldsymbol{q}) + R_i^t(\boldsymbol{q})\boldsymbol{s}_{Wt}(\boldsymbol{q}), \\ \boldsymbol{s}_{Mi}(\boldsymbol{q}) &= \boldsymbol{d}_i^t(\boldsymbol{q}) + R_i^t(\boldsymbol{q})\left(\boldsymbol{d}_t^s(\boldsymbol{q}) + R_t^s(\boldsymbol{q})\boldsymbol{s}_{Ms}(\boldsymbol{q})\right) \quad , \end{aligned} \tag{1.17}$$

where $\boldsymbol{s}_{Xy}$ represents the position vector of the center of gravity of unit $X \in \{TA, W, M\}$ in the coordinates of the system $y \in \{i, t, s\}$. It should be noted that the center of gravity of the car depends largely on the car position l_w. For the determination of the energies, the translational and rotational velocities must be calculated in dependence of the generalized coordinates $\boldsymbol{q}$. These can be determined easily and in a systematic manner using the centers of gravity and the rotation matrices according to [2]. The following applies to the translational velocity:

$$\begin{aligned} \boldsymbol{v}_{TAi}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \frac{d}{dt}\boldsymbol{s}_{TAi}(\boldsymbol{q}) = \frac{\partial}{\partial \boldsymbol{q}}\boldsymbol{s}_{TAi}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{v}_{Wi}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \frac{d}{dt}\boldsymbol{s}_{Wi}(\boldsymbol{q}) = \frac{\partial}{\partial \boldsymbol{q}}\boldsymbol{s}_{Wi}(\boldsymbol{q})\dot{\boldsymbol{q}} \\ \boldsymbol{v}_{Mi}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \frac{d}{dt}\boldsymbol{s}_{Mi}(\boldsymbol{q}) = \frac{\partial}{\partial \boldsymbol{q}}\boldsymbol{s}_{Mi}(\boldsymbol{q})\dot{\boldsymbol{q}} \quad . \end{aligned} \tag{1.18}$$

For the calculation of the vector $\boldsymbol{\omega}_{kj}$ of the angular velocity between the coordinate systems k and j, the skew-symmetric matrix $S_{kj}(\boldsymbol{\omega}_{kj})$ is calculated so that the entries of the vector can be read directly. This procedure is necessary because the time derivative of the angles due to the couplings of the individual rotations does not correspond to the angular velocities ($\dot{\boldsymbol{\varphi}} \neq \boldsymbol{\omega}$). This holds true only in the case that an object is rotated around a single axis. For the angular velocity of the tower-jib-unit as well as for the carriage we obtain

$$\begin{aligned} S_{ti}(\boldsymbol{\omega}_{ti}) &= \dot{R}_i^t(\boldsymbol{q})(R_i^t)^T(\boldsymbol{q}) = \sum_{j=1}^{n}\left(\left(\frac{\partial}{\partial q_j}R_i^t(\boldsymbol{q})\right)(R_i^t)^T(\boldsymbol{q})\dot{q}_j\right) = \begin{bmatrix} 0 & -\omega_{z_{ti}} & \omega_{y_{ti}} \\ \omega_{z_{ti}} & 0 & -\omega_{x_{ti}} \\ -\omega_{y_{ti}} & \omega_{x_{ti}} & 0 \end{bmatrix} \\ \Rightarrow \boldsymbol{\omega}_{ti} &= \begin{bmatrix} \omega_{x_{ti}} \\ \omega_{y_{ti}} \\ \omega_{z_{ti}} \end{bmatrix} \quad . \end{aligned} \tag{1.19}$$

This expression also reveals the definition of the skey-symmetric matrix S_{kj}. The angular velocity $\boldsymbol{\omega}_{si}$ of the mass unit can be determined with the matrix R_{si}. As also $\boldsymbol{\omega}_{st}$ is needed, $\boldsymbol{\omega}_{si}$ is determined in two steps:

$$\begin{aligned} S_{st}(\boldsymbol{\omega}_{st}) &= \dot{R}_t^s(\boldsymbol{q})(R_t^s)^T(\boldsymbol{q}) = \sum_{j=1}^{n}\left(\left(\frac{\partial}{\partial q_j}R_t^s(\boldsymbol{q})\right)(R_t^s)^T(\boldsymbol{q})\dot{q}_j\right) = \begin{bmatrix} 0 & -\omega_{z_{st}} & \omega_{y_{st}} \\ \omega_{z_{st}} & 0 & -\omega_{x_{st}} \\ -\omega_{y_{st}} & \omega_{x_{st}} & 0 \end{bmatrix} \\ \Rightarrow \boldsymbol{\omega}_{st} &= \begin{bmatrix} \omega_{x_{st}} \\ \omega_{y_{st}} \\ \omega_{z_{st}} \end{bmatrix} \quad , \\ \boldsymbol{\omega}_{si} &= \boldsymbol{\omega}_{ti} + R_i^t\boldsymbol{\omega}_{st} \quad . \end{aligned} \tag{1.20}$$

For the determination of the kinetic energy, the inertia tensors or inertia matrix, respectively, of the individual units are required. For simplicity, it is assumed that the inertia tensor of the mass can be neglected. If the inertia tensor is defined in the body-fixed coordinate system k of a unit X, it is constant, so that the following applies

$$J_{X_k} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} = \text{const} \quad . \tag{1.21}$$

This procedure simplifies the setup of the inertial tensor, but the tensor has to be transfered to the inertial system i, which is achieved by making use of the respective rotation matrices so that:

$$J_{X_i} = R_i^k J_{X_k} (R_i^k)^T \quad . \tag{1.22}$$

The kinetic energy of the individual units is now calculated:

$$W_{\text{kin},X_i}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \underbrace{\frac{1}{2}\boldsymbol{\omega}_{ki}^T J_{X_i} \boldsymbol{\omega}_{ki}}_{W_{\text{kin,rot},X_i}} + \underbrace{\frac{1}{2} m_X \boldsymbol{v}_{X_i}^T \boldsymbol{v}_{X_i}}_{W_{\text{kin,trans},X_i}} \quad . \tag{1.23}$$

The potential energy reads:

$$W_{\text{pot},X_i}(\boldsymbol{q},\dot{\boldsymbol{q}}) = m_X \boldsymbol{a}_{\text{grav}}^T \boldsymbol{s}_{X_i} \qquad \text{where} \quad \boldsymbol{a}_{\text{grav}} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad . \tag{1.24}$$

Here the assumption is made that gravity acts opposite to the z_i-axis.

Viscous friction is assumed in the bearings and in the running rail of the carriage. This is assumed proportional to the velocity and is summarized in the Rayleigh dissipation function as shown in 1.15). To properly set up the friction terms, note that the friction only acts between two subsystems. Thus to define R, the respective rotatory or translatory velocity vector must be considered in the correct coordinate system. Here the individual friction terms are given in the form

$$R_X = \frac{1}{2} k_X \boldsymbol{\omega}_{X_k}^T \boldsymbol{\omega}_{X_k} \quad . \tag{1.25}$$

Summing up over all individual terms yields the Rayleigh dissipation function.

For the calculation of the generalized force, which is generated by a force described in the coordinate system j, the generating force must be transformed into the coordinates of the inertial system i. Let $\boldsymbol{F}_{X_i}$ denote the force vector in the inertial system, then the respective generalized force reads

$$\boldsymbol{Q}_{X_i} = \frac{\partial \boldsymbol{v}_{X_i}}{\partial \dot{\boldsymbol{q}}} \boldsymbol{F}_{X_i} \quad , \tag{1.26}$$

in terms of the generalized coordinates. The derivation of a generalized torque is performed analogously, whereby the following relationship applies

$$\boldsymbol{Q}_{X_i} = \frac{\partial \boldsymbol{\omega}_{X_{ki}}}{\partial \dot{\boldsymbol{q}}} \boldsymbol{M}_{X_i} \quad , \tag{1.27}$$

with M_{X_i} respresenting the torque of each unit X in the inertial system i. The vector of the generalized forces and moments $\boldsymbol{Q}$ is calculated by summing up over the individual single terms $\boldsymbol{Q}_{X_i}$. It should be noted that this procedure may be simplified as in many cases. The transformation of forces and moments into the generalized forces and moments can be read directly.

The equations of motion can now be determined using the Lagrange formalism according to (1.16). The resulting system is summarized in the Matlab file `Lagrange_ex_tower_crane.mlx`. To represent the resulting coupled system of second order differential equations as a system of first order differential equations it is necessary to introduce a suitable state vector, e.g.,

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q}^T & \dot{\boldsymbol{q}}^T \end{bmatrix}^T \quad . \tag{1.28}$$

The input vector reads $\boldsymbol{u} = \begin{bmatrix} F_W & F_M & M_T \end{bmatrix}^T$ and the system outputs are given by the carriage position l_W, the rope length l_S and angles of the tower φ_{z_i}.

1.5.2 Linearization

The resulting nonlinear state space representation is now linearized around the equilibrium position $\boldsymbol{x}_R = \begin{bmatrix} l_{Wr} & l_{Sr} & \varphi_{zi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ for the stationary input $\boldsymbol{u}_R = \begin{bmatrix} 0 & F_{Mr} & 0 \end{bmatrix}^T$. For details the reader is referred to Section 2.3 of the lecture Control Systems [1]. The resulting linear system representation in the state $\Delta\boldsymbol{x}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_R$, the input $\Delta\boldsymbol{u}(t) = \boldsymbol{u}(t) - \boldsymbol{u_R}$ and the output $\Delta\boldsymbol{y}(t) = \boldsymbol{y}(t) - \boldsymbol{y_R}$ describes the system dynamics in the vincinity of the equilibrium. The result of the linearization is also shown in the Matlab file `Lagrange_ex_tower_crane.mlx`.

Exercise 1.7. *The file* `Lagrange_ex_tower_crane.mlx` *contains a Matlab-script in which the above calculations for the example of the tower crane are performed. Work through this script step by step and reproduce it, if necessary using the program help.*

1.6 Double pendulum

In the following, the double pendulum shown in Figure 1.2 is analyzed. The pendulum consists of two pendulum arms, which are connected to each other in a freely revolving manner (around the x-axis). In addition, the lower pendulum arm is revolving coupled to a carriage. The acceleration of this carriage $\ddot{y}$, which is effected by a belt drive, serves as the control input for the system. From Figure 1.2 it can be seen that the position of the pendulum arms can be described by the angles $\varphi_1(t)$, $\varphi_2(t)$. The angles of the pendulum arms are defined in counterclockwise direction around the corresponding z-axis (z_{G1}, z_{G2}) of the body-fixed coordinate systems with respect to the z-axis of the inertial system (z_i). The inertial system is positioned on the left outer edge of the track. The body-fixed coordinate systems are located at the center of the pendulum joints and the orientation of their z-axes corresponds to the longitudinal axis of the respective pendulum arm, as shown in Figure 1.2. The distance of the center of gravity to the pivot point of a pendulum arm $k \in \{1,2\}$ is denoted by l_{Sk}, the total length of an arm is given by the parameter l_k. The bearings of the pendulum joints exhibit viscous friction, which can be described as:

$$R_{Gk} = \frac{1}{2} k_{Gk} \boldsymbol{\omega_k}^T \boldsymbol{\omega_k} \quad . \tag{1.29}$$

The parameter k_{Gk} is defined as a proportional friction constant. Physically, friction occurs between the carriage and track when the carriage is moved by the belt drive. However, this friction is compensated by an underlying control circuit of the carriage drive, so that it can be ignored subsequently. The geometrical and physical data of the pendulum can be found in Table 1.2.

Exercise 1.8.

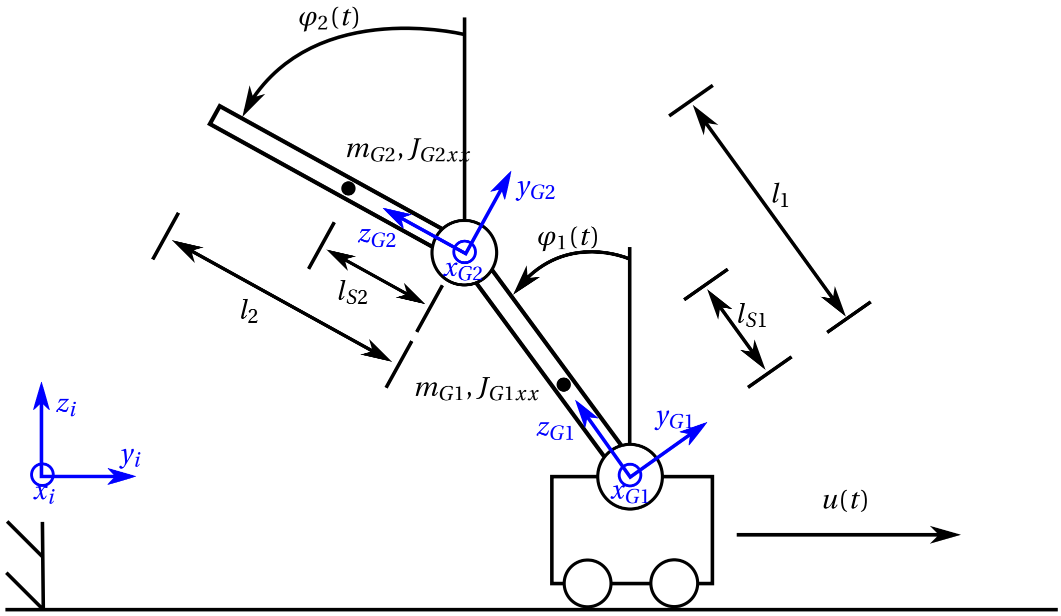

Figure 1.2: Schematic representation of the double pendulum.

Description	Parameter	Maxima variable	Value	
		Pendulum arm 1		
Length	l_1	`l1`	0.3230	m
Distance pivot to center of gravity	l_{S1}	`l_S1`	0.1870	m
Mass	m_{G1}	`m_G1`	0.8810	kg
Moment of inertia	J_{G1xx}	`J_G1xx`	0.0134	Nms^2
Friction constant	k_{G1}	`k_G1`	0.0032	Nms
		Pendulum arm 2		
Length	l_2	`l2`	0.4800	m
Distance pivot to center of gravity	l_{S2}	`l_S2`	0.1940	m
Mass	m_{G2}	`m_G2`	0.5510	kg
Moment of inertia	J_{G2xx}	`J_G2xx`	0.0208	Nms^2
Friction constant	k_{G2}	`k_G2`	0.0012	Nms

Table 1.2: Parameters of the double pendulum system and Maxima nomenclature.

(i) *Determine suitable generalized coordinates* $\boldsymbol{q}$ *for the double pendulum.*

(ii) *Set up the equations of motion of the double pendulum using the Lagrange formalism. (Concentrate here on the pendulum arms as the equation of motion of the carriage can easily be added later).*

To verify your intermediate results, the kinetic and potential energy of the system are given below:

$$W_{kin} = \frac{1}{2}\Big(\big({\dot\varphi_2}^2 l_{S2}^2 + 2 l_1 \dot\varphi_1 \dot\varphi_2 \cos(\varphi_2 - \varphi_1) l_{S2} + l_1^2 \dot\varphi_1^2\big) m_{G2} + \dot\varphi_1^2 l_{S1}^2 m_{G1} + J_{G1xx}\dot\varphi_1^2 + J_{G2xx}\dot\varphi_2^2\Big) \quad ,$$

$$W_{pot} = g\big(\cos(\varphi_2) l_{S2} + l_1\cos(\varphi_1)\big) m_{G2} + g\cos(\varphi_1) l_{S1} m_{G1} \quad .$$

(iii) *Convert the equations of motion so that they are in standard form, i.e., two equations each containing the second derivative of an angle on the left-hand side.*

References

[1] T. Meurer. „Control Systems - Lecture notes“. In: *http://www.control.tf.uni-kiel.de/en/teaching/winter-term* (2019) (cit. on pp. 3, 10).

[2] T. Meurer. „Rigid Body Dynamics and Robotics - Lecture notes“. In: *http://www.control.tf.uni-kiel.de/en/teaching/winter-term* (2019) (cit. on pp. 5, 8).