Assignment 1
ESS101

Axel Klang
Jonathan Tunell

September 16, 2025

1 Question 1

1.1 Part a)

To stay consistent through the task we defined a coordinate system based on the helicopter(m_1). This made it easier to define the coordinates for the hanging mass as a function of q as that contains the position of the helicopter.

Using the MATLAB sym package we defined all possible variables and created our discrete coordinate system as such:

$$\mathbf{p_1} = \begin{bmatrix} p_{1x} \\ p_{1y} \\ p_{1z} \end{bmatrix}, \dot{\mathbf{p_1}} = \begin{bmatrix} \dot{p_{1x}} \\ \dot{p_{1y}} \\ \dot{p_{1z}} \end{bmatrix}, \ddot{\mathbf{p_1}} = \begin{bmatrix} \ddot{p_{1x}} \\ \ddot{p_{1y}} \\ \ddot{p_{1z}} \end{bmatrix}, \mathbf{q} = \begin{bmatrix} \mathbf{p_1} \\ \theta \\ \phi \end{bmatrix}, \dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{p_1}} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}, \ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\mathbf{p_1}} \\ \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} \in \mathbb{R}^5$$

$$\mathbf{p_2} = \mathbf{p_1} + L * \begin{bmatrix} \sin(\phi) * \cos(\theta) \\ \sin(\phi) * \sin(\theta) \\ \cos(\phi) \end{bmatrix}, \dot{\mathbf{p}}_2 = \frac{\partial \mathbf{p_2}}{\partial \mathbf{q}} \dot{\mathbf{q}} \in \mathbb{R}^5$$

In order to calculate the lagrange function the formulas for the hanging mass from the task were used. The same formulas were then used for the kinetic and potential energies of the helicopter. Finally the Lagrange function was calculated:

$$Lag = (T_1 + T_2) - (V_1 + V_2) \tag{1}$$

Given the Lagrange function, it was now possible to calculate the Euler-Lagrange equation, giving it the notation E for simpler use:

$$E = \frac{d}{dt}\left(\frac{\partial Lag}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial Lag}{\partial \mathbf{q}} - \mathbf{Q} = 0 \tag{2}$$

This is applied in the code at line 34-38.
The model should then be rewritten on the format:

$$\dot{\mathbf{q}} = \mathbf{v} \tag{3a}$$
$$\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbf{b}\big(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}\big) \tag{3b}$$

Considering this and putting $E = 0$, $M(q)$ and $b\big(q, \dot{q}, u\big)$ can be "extracted" from the Euler-Lagrange equation as such:

$$M(q) = \frac{\partial E}{\partial \ddot{q}} \tag{4a}$$
$$b\big(q, \dot{q}, u\big) = -(E - M(q)\dot{v}) \tag{4b}$$

The resulting expressions can be found by running the code.

1.2 Part b)

When rewriting our model equations using the constrained Lagrange approach we first redefined our coordinate system since $\mathbf{p}_2$ is no longer a function of $\mathbf{q}$ and is instead defined explicitly, just like $\mathbf{p}_1$ in 1a). In this part we then use $\mathbf{q} = [\mathbf{p}_1; \mathbf{p}_2] \in \mathbb{R}^6$ as generalized coordinates.

The holonomic constraint is defined in the code as:

$$C = \frac{1}{2}\big(e^\top e - L^2\big), \qquad e = \mathbf{p}_1 - \mathbf{p}_2$$

The kinetic and potential energies are calculated in the same way as in (1) but expressed in the new state variables:

$$T = \frac{1}{2}m_1\, \dot{\mathbf{p}}_1^\top \dot{\mathbf{p}}_1 + \frac{1}{2}m_2\, \dot{\mathbf{p}}_2^\top \dot{\mathbf{p}}_2, \qquad V = g\big(m_1\, p_{1z} + m_2\, p_{2z}\big). \tag{5}$$

The Lagrangian used in the code is then:

$$\mathrm{Lag} = T - V - Z * C, \tag{6}$$

where Z is the Lagrange multiplier for the holonomic constraint.

From this the equations of motion are computed and defined as the Euler-Lagrange equation and the constraint equation:

$$E = \frac{d}{dt}\left(\frac{\partial \mathrm{Lag}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathrm{Lag}}{\partial \mathbf{q}} - \mathbf{Q} = 0 \tag{7}$$

$$C(q) = \frac{1}{2}(e^\top e - L^2) = 0 \tag{8}$$

with the external forces defined as:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{u} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{9}$$

$$\mathbf{u} \in \mathbb{R}^3. \tag{10}$$

From this the matrix M and the generalized force vector $\mathbf{b}$ are then calculated on lines 80 and 81 like in task 1a according to equation 4.

In task 1a, $\mathbf{p}_2$ was expressed as a function of $\mathbf{q}$ containing $\mathbf{p}_1$, which made $M(\mathbf{q})$ and $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ very complex with many trigonometric terms. In task 1b $\mathbf{p}_1$ and $\mathbf{p}_2$ are treated separately and linked by the constraint $C(\mathbf{q})$, which shifts most of this complexity into the constraint. As a result $M(\mathbf{q})$ becomes almost block–diagonal and $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ simpler. Therefore the constrained formulation reduces the complexity of M and $\mathbf{b}$ at the cost of an additional constraint equation and the multiplier Z.

2 Question 2

2.1 Part a)

In order to get the model on the form specified in the task, the left hand side matrix was constructed by first computing:

$$\mathbf{a}(\mathbf{q}) = \nabla_q C \tag{11}$$

together with the matrix M from task 1b which gives the left hand side matrix:

$$LHS_{\text{mat}} = \begin{bmatrix} \mathbf{M} & \mathbf{a}(\mathbf{q}) \\ \mathbf{a}(\mathbf{q})^\top & 0 \end{bmatrix} \tag{12}$$

At this point $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ can be defined as the non-$\ddot{q}$ and Z terms of the E and $\ddot{C}$ functions. Therefore the $\ddot{q}$ and Z terms in each function is substituted with 0 in order to eliminate those parts. At that point, both $\mathbf{a}(\mathbf{q})$ and $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ can be calculated by running the code.

2.2 Part b)

In this part we solved the implicit model from part 2b for the unknowns $\ddot{\mathbf{q}}$ and Z. In MATLAB this was implemented with the matrix left division operator, \. Meaning it symbolically computes

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ Z \end{bmatrix} = \begin{bmatrix} M(\mathbf{q}) & a(\mathbf{q}) \\ a(\mathbf{q})^\top & 0 \end{bmatrix}^{-1} c(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}).$$

This gives explicit expressions for the accelerations and the Lagrange multiplier for a given state.

Compared to part 2a, where the model was written in compact implicit form, the resulting expressions for $\ddot{\mathbf{q}}$ and Z are much more complex because the matrix inversion expands all terms. The implicit form (28) is short and keeps M, a and c separate but requires the program to solve the system everytime it runs, while the explicit form (29) can be used directly but produces more complex symbolic expressions.

3 Optional simulation for fun

We performed a small simulation, with the help of ChatGPT on how to apply the model , for fun. We set some initial values and apply a force Q based on time and the model is used to calculate the trajectory of both the helicopter m1 and the hanging mass m2.

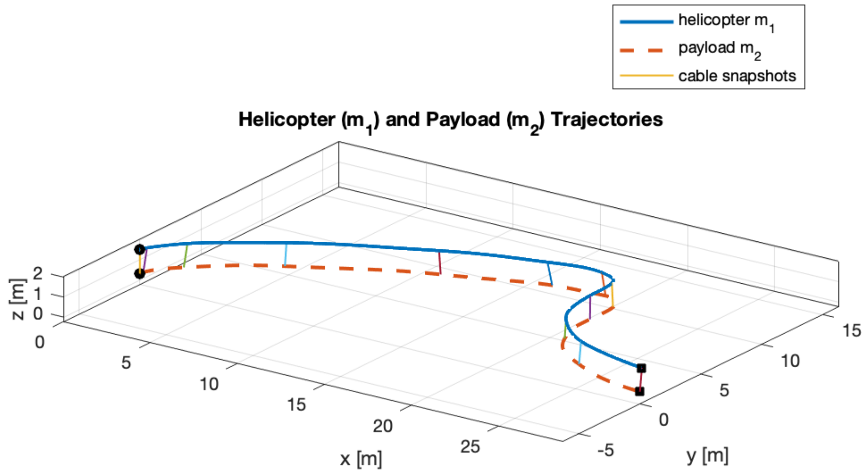

Figure 1: 3D-view of both trajectories with snapshots of the constraining rigid cable

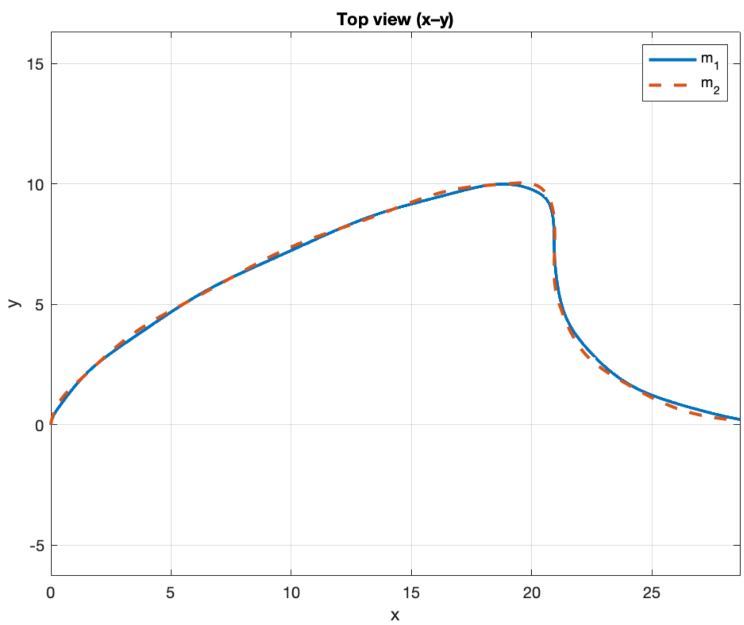

Figure 2: Top view of both trajectories