ESS101
Modelling and Simulation, 2025

Lecturer and examiner: Yasemin Bekiroğlu
Course assistant: Ahmet Tekden

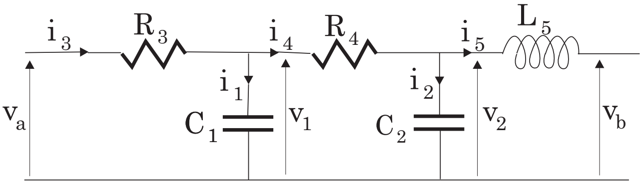
Systems & Control Division
Department of Electrical Engineering
Chalmers University of Technology

September, 2025

Lecture 3 – Physical modelling

The process of going from characterizing a system from its physical properties to determining a useful state space model.

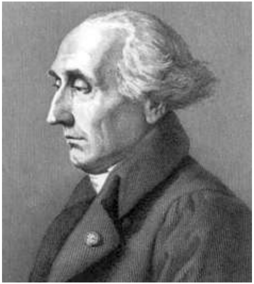
$$\dot{x}(t) = f(x(t), u(t))$$
$$y(t) = h(x(t), u(t))$$



introducing $\boldsymbol{x} = (v_1, v_2, i_5)$ and $\boldsymbol{u} = (v_a, v_b)$:

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} -\frac{1}{R_3C_1} - \frac{1}{R_4C_1} & \frac{1}{R_4C_1} & 0 \\ \frac{1}{R_4C_2} & -\frac{1}{R_4C_2} & -\frac{1}{C_2} \\ 0 & \frac{1}{L_5} & 0 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} \frac{1}{R_3C_1} & 0 \\ 0 & 0 \\ 0 & -\frac{1}{L_5} \end{bmatrix} \boldsymbol{u}(t)$$

Lagrange modelling



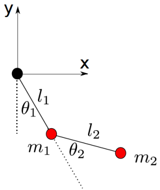
- Generalized coordinates
- Kinetic and potential energy
- Lagrange function
- Euler-Lagrange's equation and examples

Learning objective:

- Use methods and tools to **develop mathematical models of dynamical systems by using basic physical laws**. The emphasis will be on complex mechanical systems.

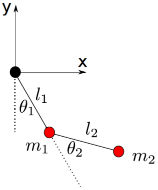
Lagrange Mechanics

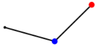
- Physical modelling complemented with **domain specific knowledge**.
- We will look into how domain specific techniques can facilitate physical modelling process, based on ***Lagrange Mechanics***.
- Lagrange Mechanics is a tool to **build mathematical models for mechanical systems**.
  - allows to describe arbitrarily complex mechanical systems, e.g. relative moving parts, accelerated frames.
  - allows to build simple models.



Lagrange Mechanics

- Physical modelling complemented with domain specific knowledge.
- We will look into how domain specific techniques can facilitate physical modelling process, based on ***Lagrange Mechanics***.
- Lagrange Mechanics is a tool to build mathematical models for mechanical systems.
  - allows to describe arbitrarily complex mechanical systems, e.g. relative moving parts, accelerated frames.
  - allows to build simple models.





Generalized coordinates of a mechanical system

Definition (Generalized coordinates)

A vector $\mathbf{q}(t) \in \mathrm{R}^{n_q}$ that describes the " configuration" (position of all parts) of a mechanical system is called *generalized coordinates*.

- does not tell how the system will evolve, but can tell **in what configuration the system is at a given time**.
- If n_q is equal to the number of degrees-of-freedom (DOF) of the system, the generalized coordinates are independent or *free*.
- In practice, the generalized coordinates are **usually positions, lengths or angles**.

Generalized coordinates of a mechanical system

- Represent **the state of a system in a configuration space**. These parameters must uniquely define the configuration of the system relative to a reference state.
- An example: the position of a pendulum using **the angle of the pendulum relative to vertical.**
- There may be many possible choices for generalized coordinates for a physical system, they are generally **selected to simplify calculations**.

Lagrange Mechanics

- Lagrange mechanics is based on a description of the mechanical system in terms of **energy**.
- To build models using Lagrange equations, we need to compute **Kinetic** and **Potential energy** functions of the system, denoted as T and V.
- Leading to **dynamical description, equation of motion** - describe the **behavior of a physical system** as a set of mathematical functions in terms of dynamic variables.

Kinetic energy

Consider a mechanical system with N particles, having masses $\{m_i\}$ and positions $\{\mathbf{p}_i\} \in \mathrm{R}^D$ with $D = 1, 2$ or 3.

The *kinetic energy* T of the system is defined as

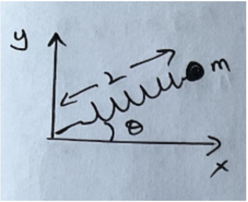
$$T = \frac{1}{2} \sum_{i=1}^{N} m_i \dot{\mathbf{p}}_i^T \dot{\mathbf{p}}_i, \quad \mathbf{p}_i = \mathbf{p}_i(t)$$

Using generalized coordinates $\mathbf{q}$,

$$\mathbf{p}(t) = \mathbf{p}(\mathbf{q}(t)) \quad \Rightarrow \quad \dot{\mathbf{p}} = \frac{\partial \mathbf{p}}{\partial \mathbf{q}} \dot{\mathbf{q}},$$

$$T = \frac{1}{2} \dot{\mathbf{q}}^T \Big(\sum_{i=1}^{N} m_i \frac{\partial \mathbf{p}_i}{\partial \mathbf{q}}^T \frac{\partial \mathbf{p}_i}{\partial \mathbf{q}} \Big) \dot{\mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T W(\mathbf{q}) \dot{\mathbf{q}}$$

Kinetic Energy - example



$$p = \begin{bmatrix} x \\ y \end{bmatrix} = L \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \qquad q = \begin{bmatrix} \theta \\ L \end{bmatrix}$$

$$T = \frac{1}{2} m \dot{p}^T \dot{p} = \frac{1}{2} m \begin{bmatrix} \dot{\theta} \\ \dot{L} \end{bmatrix}^T \begin{bmatrix} -L\sin\theta & L\cos\theta \\ \cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} -L\sin\theta & \cos\theta \\ L\cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{L} \end{bmatrix}$$

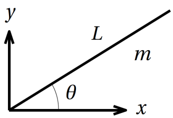
$$= \frac{1}{2} m \begin{bmatrix} \dot{\theta} \\ \dot{L} \end{bmatrix}^T \begin{bmatrix} L^2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{L} \end{bmatrix}$$

$$= \frac{1}{2} \dot{q}^T \underbrace{\begin{bmatrix} mL^2 & 0 \\ 0 & m \end{bmatrix}}_{W(q)} \dot{q}$$

Potential energy

Example:

- The potential energy **due to gravity** in most mechanical applications: $V = mgz$ where z is the height of the mass.
- The mass m is concentrated at the end of a rigid rod, the vertical position is given by: $p_z = Lsin\theta$, its potential energy is given by $V = mgLsin\theta$.



Euler-Lagrange's equation – summary

Kinetic, potential energies and the Lagrangian, expressed in generalized coordinates $\mathbf{q}$:

$$T = T(\mathbf{q}, \dot{\mathbf{q}}), \quad V = V(\mathbf{q}), \quad \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = T(\mathbf{q}, \dot{\mathbf{q}}) - V(\mathbf{q})$$

The Euler-Lagrange equation:

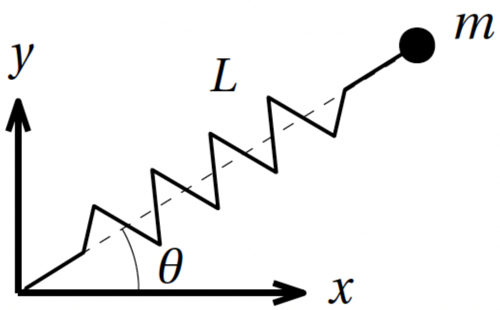
$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = 0 \qquad \text{or} \qquad \frac{d}{dt}\nabla_{\dot{\mathbf{q}}}\mathcal{L} - \nabla_{\mathbf{q}}\mathcal{L} = 0,$$

$$\nabla_{\mathbf{q}}\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{q}}\right)^T, \quad \nabla_{\dot{\mathbf{q}}}\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right)^T, \quad T = \frac{1}{2}\dot{\mathbf{q}}^T W(\mathbf{q})\dot{\mathbf{q}},$$

the Euler-Lagrange equation reads

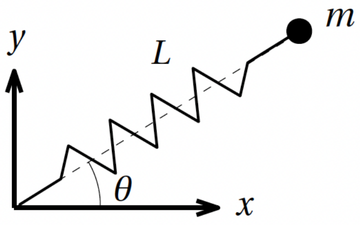
$$W(\mathbf{q})\ddot{\mathbf{q}} + \frac{\partial}{\partial \mathbf{q}}\left(W(\mathbf{q})\dot{\mathbf{q}}\right)\dot{\mathbf{q}} - \nabla_{\mathbf{q}} T + \nabla_{\mathbf{q}} V = 0$$

Example



- Derive E-L equations using the $[q = \theta, L]$:
- Derive E-L using $[q = x, y]$:

Example:



- Resulting E-L equations using the $[q = \theta, L]$:

$$0 = \begin{bmatrix} mL^2 & 0 \\ 0 & m \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} 2mL\dot{L}\dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} mgLcos\theta \\ mgsin\theta + K(L - L_0) - mL\dot{\theta}^2 \end{bmatrix}$$

- E-L using $[q = x, y]$:

$$0 = m\ddot{\mathbf{q}} + mg \begin{bmatrix} 0 \\ 1 \end{bmatrix} + K\left(1 - \frac{L_0}{\|q\|}\right) q$$

Some important points:

- Even for a simple system, modelling can be complicated, you can use symbolic computations in Matlab (e.g. differentiation)

Some important points:

- Even for a simple system, modelling can be complicated, you can use symbolic computations in Matlab (e.g. differentiation)
- Modelling, setting up the basic expressions, kinetic, potential energy, E-L equation, is very straightforward.

Some important points:

- Even for a simple system, modelling can be complicated, you can use symbolic computations in Matlab (e.g. differentiation)
- Modelling, setting up the basic expressions, kinetic, potential energy, E-L equation, is very straightforward.
- Complexity of the equations changes based on how you choose the generalized coordinates.

Some important points:

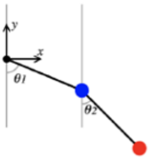
- Even for a simple system, modelling can be complicated, you can use symbolic computations in Matlab (e.g. differentiation)
- Modelling, setting up the basic expressions, kinetic, potential energy, E-L equation, is very straightforward.
- Complexity of the equations changes based on how you choose the generalized coordinates.
- We can include other forces (not just gravity) that externally affect the system.

Some important points:

- Even for a simple system, modelling can be complicated, you can use symbolic computations in Matlab (e.g. differentiation)
- Modelling, setting up the basic expressions, kinetic, potential energy, E-L equation, is very straightforward.
- Complexity of the equations changes based on how you choose the generalized coordinates.
- We can include other forces (not just gravity) that externally affect the system.
- Once the generalized forces $\mathbf{Q}$ are known, they can be readily included in the Lagrange formalism using:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{Q}^T \qquad \text{or} \qquad \frac{d}{dt}\nabla_{\dot{\mathbf{q}}}\mathcal{L} - \nabla_{\mathbf{q}}\mathcal{L} = \mathbf{Q},$$

Double pendulum example



$$
\begin{aligned}
x_1 &= l_1 \sin\theta_1 & \dot{x}_1 &= l_1\dot{\theta}_1 \cos\theta_1 \\
y_1 &= -l_1 \cos\theta_1 & \dot{y}_1 &= l_1\dot{\theta}_1 \sin\theta_1 \\
x_2 &= l_1 \sin\theta_1 + l_2 \sin\theta_2 & \dot{x}_2 &= l_1\dot{\theta}_1 \cos\theta_1 + l_2\dot{\theta}_2 \cos\theta_2 \\
y_2 &= -l_1 \cos\theta_1 - l_2 \cos\theta_2 & \dot{y}_2 &= l_1\dot{\theta}_1 \sin\theta_1 + l_2\dot{\theta}_2 \sin\theta_2
\end{aligned}
$$

$V =$

$T =$

$\mathcal{L} =$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = 0 \quad \text{for } q_i = \theta_1, \theta_2.$$

let $z_1 \equiv \dot{\theta}_1 \Rightarrow \ddot{\theta}_1 = \dot{z}_1$ and $z_2 \equiv \dot{\theta}_2 \Rightarrow \ddot{\theta}_2 = \dot{z}_2$.

$\dot{z}_1 =$

$\dot{z}_2 =$

Double pendulum example

```python
def deriv(y, t, L1, L2, m1, m2):
    """Return the first derivatives of y = theta1, z1, theta2, z2."""
    theta1, z1, theta2, z2 = y

    c, s = np.cos(theta1-theta2), np.sin(theta1-theta2)

    theta1dot = z1
    z1dot = (m2*g*np.sin(theta2)*c - m2*s*(L1*z1**2*c + L2*z2**2) -
             (m1+m2)*g*np.sin(theta1)) / L1 / (m1 + m2*s**2)
    theta2dot = z2
    z2dot = ((m1+m2)*(L1*z1**2*s - g*np.sin(theta2) + g*np.sin(theta1)*c) +
             m2*L2*z2**2*s*c) / L2 / (m1 + m2*s**2)
    return theta1dot, z1dot, theta2dot, z2dot

def calc_E(y):
    """Return the total energy of the system."""

    th1, th1d, th2, th2d = y.T
    V = -(m1+m2)*L1*g*np.cos(th1) - m2*L2*g*np.cos(th2)
    T = 0.5*m1*(L1*th1d)**2 + 0.5*m2*((L1*th1d)**2 + (L2*th2d)**2 +
            2*L1*L2*th1d*th2d*np.cos(th1-th2))
    return T + V

# Maximum time, time point spacings and the time grid (all in s).
tmax, dt = 30, 0.01
t = np.arange(0, tmax+dt, dt)
# Initial conditions: theta1, dtheta1/dt, theta2, dtheta2/dt.
y0 = np.array([3*np.pi/7, 0, 3*np.pi/4, 0])

# Do the numerical integration of the equations of motion
y = odeint(deriv, y0, t, args=(L1, L2, m1, m2))
```