Technische Universität Berlin
Institute for Process and Chemical Engineering
Chair of Control Engineering

# IESS – Laboratory 1

## Modeling a self-balancing robot

Mechatronic systems are complex systems that integrate mechanical, electrical, and software components to achieve a desired functionality. Modeling is an essential tool in the design and analysis of mechatronic systems, as it gives insight into how the different components of the system interact with each other, and how they will behave under different operating conditions.

The mechatronic model can help with:

**System Design** Models are used to design and simulate mechatronic systems before they are built. Engineers can use these models to test different design configurations, optimize the system's performance, and identify potential problems or issues that may arise during operation.

**Control System** Mechatronic systems often require advanced control systems to achieve their desired functionality. Models are used to design and test these control systems before they are implemented, allowing engineers to optimize the system's performance, reduce development time, and ensure that the system operates safely and reliably.

**Prediction** Models can be used to predict how a system will behave in the future, based on its current state and the inputs it receives. This can help us make better decisions and plan for the future, whether it's in the context of a business, a government, or a scientific experiment.

**Optimization** Models can be used to optimize the performance of a system, by identifying the input values that will result in the best output. This can be especially useful in engineering and manufacturing, where small changes in input values can have a big impact on the quality and efficiency of a product.

**Performance Analysis** Models are used to analyze the performance of mechatronic systems under different operating conditions. Engineers can use these models to identify potential bottlenecks or limitations in the system, optimize its performance, and ensure that it meets its design requirements.

In this laboratory, we will use Euler-Lagrange modeling to derive a first principles model of a freestanding robot. The requirements for the laboratory include:

- Basic knowledge of robotics and control systems (mechatronic systems)
- Mathematical understanding of equations of motion and state-space models
- Basic knowledge in circuit analysis

**Defining a coordinate system and model parameters**

Defining a coordinate system is a critical step when modeling any physical system, as it allows us to describe the position, orientation, and motion of objects within the system. Figure 1 defines the coordinate system that we will use for a free-standing robot with two degrees of freedom. Table 1 defines the states and input of the system and Table 2 defines the parameters of the free-standing robot.

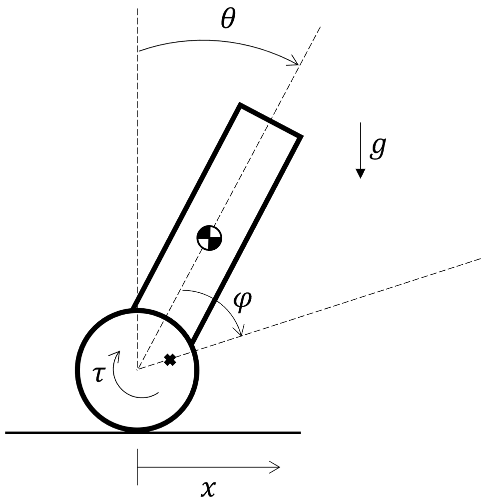


Figure 1: Schematic of a self-balancing robot.

Table 1: The states used to describe the motion of the self-balancing robot and the driving torque input. *Note:* all states are functions of time, $t$, though we have dropped this for brevity.

| State | [unit] | Definition |
|---|---|---|
| $\varphi$ | [radians] | Angle between chassis and wheel initial position |
| $\theta$ | [radians] | Clockwise angle of the chassis from the vertical position |
| $x$ | [m] | Linear displacement of the robot |
| $\dot{\varphi}$ | [radians·s$^{-1}$] | Angular velocity of the wheel |
| $\dot{\theta}$ | [radians·s$^{-1}$] | Angular velocity of the chassis |
| $\dot{x}$ | [m·s$^{-1}$] | Linear velocity of the robot |
| **Input** | **[unit]** | **Definition** |
| $\tau$ | [N·m] | Driving torque of the wheel |

From our coordinates in Figure 1, the states are related by using the arc length formula:

$$x = r(\theta + \varphi)\,. \tag{1}$$

Table 2: Parameters used to describe kinetics and kinematics of a free-standing robot on an inclined plane.

| Parameter | [unit] | Definition |
|---|---|---|
| $r$ | [m] | Radius of the wheels |
| $m$ | [kg] | Combined mass of both wheels and motors |
| $J$ | [kg·m$^2$] | Mass moment of inertia for $m$ about the axle |
| $M$ | [kg] | Mass of the robot chassis |
| $I$ | [kg·m$^2$] | Mass moment of inertia for $M$ about the COM |
| $l$ | [m] | Distance to the center of gravity from the wheel axle |
| $g$ | [m·s$^{-2}$] | Acceleration due to gravity (defined to be negative) |

Differentiating and noting linearity of differentiation gives

$$\dot{x} = r\left(\dot{\theta} + \dot{\varphi}\right) \tag{2}$$

$$\ddot{x} = r\left(\ddot{\theta} + \ddot{\varphi}\right) . \tag{3}$$

**Euler-Lagrange modeling**

The Euler-Lagrange equations describe the motion of a mechanical system in terms of its energy and can be used to derive the equations of motion for a self-balancing robot. Assuming a simplified model where the robot is a single rigid body with two wheels and a mass concentrated at its center of mass, as depicted in Figure 1, the Lagrangian $\mathcal{L}$ can be expressed by the kinetic co-energy, $\mathcal{T}^*$, and potential energy, $\mathcal{V}$, to be $\mathcal{L}(q,\dot{q}) = \mathcal{T}^*(q,\dot{q}) - \mathcal{V}(q)$. Derived from Newton's second law, the generalized Euler-Lagrange equation is given to be:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i - \frac{\partial \mathcal{D}}{\partial \dot{q}_i} , \tag{4}$$

where $\mathcal{D}$ is the dissipation function, $q_i$ is the $i$-th coordinate, and $\tau_i$ is the generalized force.

There are several mathematically messy steps involved in deriving the Euler-Lagrange model of our system, thus, this section is broken into first finding the Lagrangian, $\mathcal{L}$, by calculating the energy of the system, followed by finding the equations of motion for $\varphi$ and $\theta$, the generalized coordinates of our system.

**The Lagrangian**

The Lagrangian is the total energy. For a self-balancing robot, the potential energy can be modeled as a function of the angle between the robot's body and the vertical axis, $\theta$, and the distance between the robot's center of mass and the ground, $h$:

$$\mathcal{V} = Mgh , \tag{5}$$

where $g$ is the acceleration due to gravity and $h$ is calculated as $h = l\cos\theta$ . Then the potential energy of the system is given by

$$\mathcal{V} = Mgl\cos\theta . \tag{6}$$

The Kinetic co-energy of the system can be expressed as the kinetic energy of the wheel and the chassis together as

$$\mathcal{T}^* = \frac{1}{2}mv^2 + \frac{1}{2}MV^2 \tag{7}$$

where $v$ and $V$ are made up of linear and inertial components:

$$v^2 = \dot{x}^2 + \frac{J}{m}(\dot{\theta} + \dot{\varphi})^2 \tag{8}$$

$$V^2 = V_x^2 + V_y^2 + \frac{I}{M}\dot{\theta}^2\,, \tag{9}$$

respectively, where $V_x = \dot{x} + l\dot{\theta}\cos(\theta + \varphi)$, and $V_y = \frac{\mathrm{d}}{\mathrm{d}t}(l\cos\theta) = l\dot{\theta}\sin\theta$. Then, the kinetic co-energy of the robot expressed as the sum of the translational and rotational kinetic energies is given by:

$$\begin{aligned}
\mathcal{T}^* &= \frac{1}{2}m\left(\dot{x}^2 + \frac{r^2}{2}(\dot{\theta} + \dot{\varphi})^2\right) + \frac{1}{2}M\left(\dot{x}^2 + 2\dot{x}\dot{\theta}l\cos\theta + l^2\dot{\theta}^2 + \frac{I}{M}\dot{\theta}^2\right) && (10)\\
&= \frac{1}{2}(m+M)r^2(\dot{\theta} + \dot{\varphi})^2 + \frac{1}{2}J(\dot{\theta} + \dot{\varphi})^2 + Mr(\dot{\theta} + \dot{\varphi})\dot{\theta}l\cos\theta + \frac{1}{2}l^2M\dot{\theta}^2 + \frac{1}{2}I\dot{\theta}^2 && (11)\\
&= \frac{1}{2}(J + (m+M)r^2)(\dot{\theta} + \dot{\varphi})^2 + Mlr(\dot{\theta} + \dot{\varphi})\dot{\theta}\cos\theta + \frac{1}{2}(I + Ml^2)\dot{\theta}^2\,. && (12)
\end{aligned}$$

The Lagrangian is then given by

$$\mathcal{L} = \frac{1}{2}(J + (m+M)r^2)(\dot{\theta} + \dot{\varphi})^2 + Mlr(\dot{\theta} + \dot{\varphi})\dot{\theta}\cos\theta + \frac{1}{2}(I + Ml^2)\dot{\theta}^2 - Mgl\cos\theta\,. \tag{13}$$

**Task 1 – Euler-Lagrange equations of motion**

Consider the Euler-Lagrange equation of motion with respect to some coordinate $q$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial\mathcal{L}}{\partial\dot{q}}\right) - \frac{\partial\mathcal{L}}{\partial q} = \tau_{\mathrm{ext}} + \tau_{\mathrm{D}}\,, \tag{14}$$

where $\tau_{\mathrm{ext}}$ is the external torque applied to the system and $\tau_{\mathrm{D}}$ is the torque of the disturbances then

1. Derive the Euler-Lagrange equation for coordinate $\varphi$.
2. Derive the Euler-Lagrange equation for coordinate $\theta$.
3. Put the non-linear model into the matrix form

$$\mathcal{M}(q)\ddot{q} + \mathcal{C}(q,\dot{q})\dot{q} + \mathcal{G}(q) = \bar{\tau}\,. \tag{15}$$

**Linearization**

Linearization is the process of approximating a nonlinear system with a simpler linear model. Nonlinear systems are typically more difficult to analyze and control than linear systems, as their behavior can be complex and unpredictable. Linearization allows us to simplify the analysis of nonlinear systems by approximating their behavior as a linear system around a specific operating point.

Linearization is necessary because many control design and analysis techniques are developed for linear systems. Therefore, by approximating a nonlinear system as a linear one around a specific operating point, we can:

**Simplify Analysis and Design** Linear systems are well understood, and there are many powerful tools available for their analysis and control.

**Apply Linear Control Techniques** Techniques such as LQR (Linear Quadratic Regulator), Kalman filtering, and state feedback control are designed for linear systems.

**Facilitate Stability Analysis** The stability of a system can be more easily analyzed in the linear domain, especially using techniques like eigenvalue analysis.

The linearization process involves two main steps:

**Finding the operating point** The first step is to find an operating point around which the system can be linearized. This point is usually chosen to be the system's equilibrium point or a nominal operating point that is close to the equilibrium point.

**Linearizing the system** Once the operating point is found, the next step is to linearize the system by approximating its behavior with a linear model. This is done for a system $\dot{x} = f(x,u)$ by taking Taylor's expansion around the operating point $\bar{x}$ and $\bar{u}$ given by:

$$\dot{x} \approx f(\bar{x},\bar{u}) + \left.\frac{\partial f}{\partial x}\right|_{\substack{x=\bar{x}\\u=\bar{u}}} (x-\bar{x}) + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=\bar{x}\\u=\bar{u}}} (u-\bar{u}) \tag{16}$$

where $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial u}$ are the Jacobian matrices of the system's dynamics with respect to the state and input variables, respectively, evaluated at the operating point.

**Task 2 – Linearizing the self-balancing robot model**

1. Find the input to the system and define $u$.
2. Find the equilibrium point or a nominal operating point for linearizing the system.
3. Form the equations for the system in the form $\dot{x} = f(x,u)$ for the state vector $x = \begin{bmatrix} q & \dot{q} \end{bmatrix}^\top = \begin{bmatrix} \varphi & \theta & \dot{\varphi} & \dot{\theta} \end{bmatrix}^\top$.
4. Compute the Jacobian matrix $\frac{\partial f}{\partial x}$.

5. Compute the Jacobian matrix $\frac{\partial f}{\partial u}$.
6. Use (16) with $\Delta x = x - \bar{x}$, and $\Delta u = u - \bar{u}$ and the previous tasks to compute the linearized system around the operating point using

$$\begin{aligned}\Delta \dot{x} &= \left.\frac{\partial f}{\partial x}\right|_{\substack{x=\bar{x}\\u=\bar{u}}} \Delta x + \left.\frac{\partial f}{\partial u}\right|_{\substack{x=\bar{x}\\u=\bar{u}}} \Delta u & (17)\\ &= A\Delta x + B\Delta u\,, & (18)\end{aligned}$$

where $A \in \mathbb{R}^{4\times 4}$ and $B \in \mathbb{R}^{4\times 1}$.

**Non-ideal motor and gearbox**

A non-ideal motor with gearbox as shown in Figure 2 is commonly modeled to understand the performance of the motor system under realistic conditions. In real-world scenarios, motors are rarely ideal; they experience friction, winding resistance, and inductance, which all have an impact on their performance. Similarly, gearboxes introduce additional non-idealities like friction, backlash, and mechanical losses, which also affect the motor's overall performance.

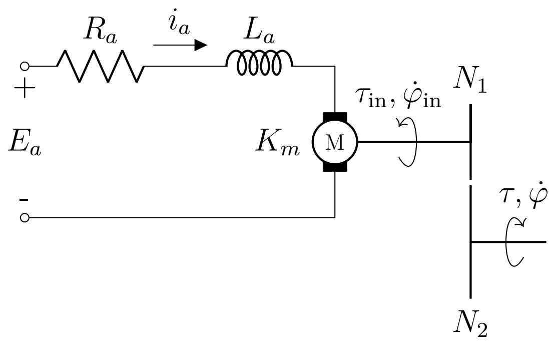


Figure 2: Schematic of a DC motor with a gearbox

The deadband model of a DC motor is a mathematical model used to describe the behavior of a motor when there is a region of input voltage where the motor does not respond. The deadband is a range of input voltages where the motor's output remains constant, despite changes in the input voltage.

In the deadband model, the motor's response to the input voltage is modeled as a piecewise linear function given by:

$$\tau_{\text{in}} = \begin{cases} K_m i_a + \tau_m & \text{if } K_m i_a < -\tau_m\,, \\ 0 & \text{if } -\tau_m \le K_m i_a \le \tau_m\,, \\ K_m i_a - \tau_m & \text{if } K_m i_a > \tau_m\,, \end{cases} \tag{19}$$

where $\tau_m$ is the torque required to be overcome before the rotor begins to move. To model these frictional losses in the gearbox, we use

$$\dot{\varphi}_{\text{in}} = N\dot{\varphi} \tag{20}$$

$$\tau = \eta N \tau_{\text{in}}\,, \tag{21}$$

where $\eta$ is the motor efficiency and $N$ is the gear ratio given by $N = \frac{N_2}{N_1}$.

### Task 3 – Modeling a non-ideal motor and gearbox

1. Sketch the torque-current characteristic in (19) of the DC motor with a deadband zone.

2. Use that the voltage over the motor is given by $V_m = K_m \dot{\varphi}_i$ and perform Kirchoff's current law around the circuit in Figure 2 to relate the armature current $i_a$, motor voltage $E_a$, and the angular velocity of the motor shaft $\dot{\varphi}_i$.

3. Express the armature current $i_a$ as a function of the output torque $\tau$ using the non-ideal gearbox model in (19).

4. Assume that the motor is operating in a steady-state (constant velocity). Express the angular velocity of the shaft $\dot{\varphi}$ as a function of the output torque $\tau$.

5. Use the stall data in Table 3 to determine the armature resistance $R_a$.

6. Use the no load data in Table 3 to determine the friction torque $\tau_m$ and the motor constant $K_m$.

7. Determine the gearbox efficiency $\eta$.

Table 3: Motor parameters used to model the motor

| Parameter | Values [unit] | Definition |
|---|---|---|
| $E_a$ | 12 [V] | Motor input voltage |
| $N$ | 20 [-] | Gear ratio |
| $\tau_{\text{out},s}$ | 7 [N·m] | Stall torque |
| $i_s$ | 18 [A] | Stall current |
| $\text{RPM}_s$ | 0 [RPM] | Rotations per minute when stalled |
| $\tau_{\text{out},NL}$ | 0 [N·m] | No load torque |
| $i_{NL}$ | 0.4 [A] | No load current |
| $\text{RPM}_{NL}$ | 160 [RPM] | Rotations per minute for the unloaded case |

### Extensions

The derived model from Task 1 can be extended to include disturbances due to friction, account for a non-flat surface, or add extra degrees of freedom. This section suggests some extension items for having a more accurate, but more complex, self-balancing robot model. Tables 4 and 5 define the additional states and inputs, and parameters, respectively, that could be useful to extend the robot model.

Table 4: The additional state used to describe the motion of the self-balancing robot and the driving torque inputs of the individual wheels.

| State | [unit] | Definition |
|---|---|---|
| $\psi$ | [radians] | Yaw angle of robot |
| $y$ | [m] | Linear displacement of robot (See Figure4) |
| $\dot{\psi}$ | [radians·s$^{-1}$] | Heave velocity (yaw rate) |
| $\dot{y}$ | [m·s$^{-1}$] | linear velocity of the robot (See Figure 4) |
| **Input** | **[unit]** | **Definition** |
| $\tau_\mathrm{L}$ | [N·m] | Driving torque of the left wheel |
| $\tau_\mathrm{R}$ | [N·m] | Driving torque of the right wheel |

Table 5: Additional parameters used to describe kinetics and kinematics of a free-standing robot in the extension tasks.

| Parameter | [unit] | Definition |
|---|---|---|
| $h_0$ | [m] | Initial height of the chassis center of gravity |
| $c$ | [-] | Angular viscous damping between the chassis and the axle |
| $b$ | [-] | Linear viscous damping between the wheel and the ground |
| $w$ | [m] | Width of the chassis |
| $\alpha$ | [radians] | Angle of the inclined plane |

**Extension 1 – Adding friction**

Add linear viscous damping between the wheel and the ground and angular viscus damping between the chassis and the axle. The new Euler-Lagrange equation should be of the form

$$\mathcal{M}(q)\ddot{q} + \mathcal{C}(q,\dot{q})\dot{q} + \mathcal{D}\dot{q} + \mathcal{G}(q) = \bar{\tau}\,, \tag{22}$$

where $\mathcal{D}$ is the damping matrix.

**Extension 2 – Modeling on an inclined plane**

1. Derive the Euler-Lagrange model for the self-balancing robot on an inclined plane as depicted in Figure 3.

2. Find the relationship between $\theta$ and $\tau$ for a stationary robot.

**Extension 3 – Modeling with an additional degree of freedom**

Derive the Euler-Lagrange model with an additional degree of freedom (coordinate $\psi$ in Figure 4), considering the left and right wheel toques as separate inputs.

**Extension 4 – Finding the critical "saving" angle of the robot**

Since the motor voltage is limited to $-E_{a,\max} \leq E_a \leq E_{a,\max}$, then there is a range of angles $\theta^- \leq \theta \leq \theta^+$ for which the robot can be "saved". Determine the critical angles $\theta^-$ and $\theta^+$ for the self-balancing robot.

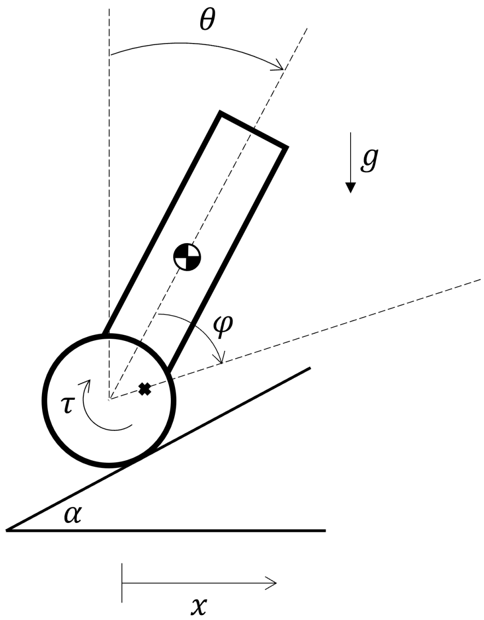


Figure 3: Schematic of a self-balancing robot on an inclined plane.

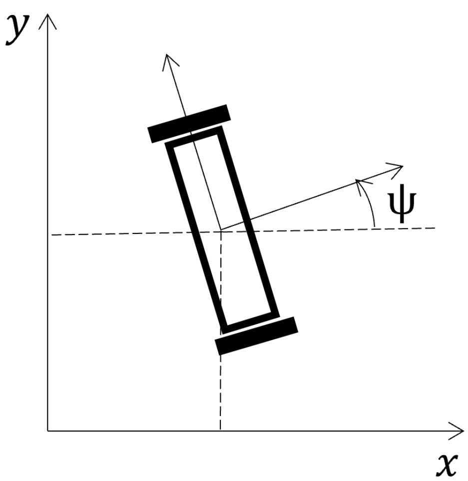


Figure 4: Schematic of a robot moving in 2-D space.

**Acknowledgments**
This laboratory was developed by Roxanne R. Jackson, Magnus Axelson-Fisk, and Tom J. Jüstel. ChatGPT (AI-generated text) was used to assist in writing the content of this laboratory.