Stability Control of a Ferromagnetic Pendulum via Electromagnetic Actuation

A. Popov (2404982)
A. Seleznev (2237318)

Linear systems, Signals and Control
Eindhoven University of Technology
January 31, 2026

Contents

1 Introduction

This project describes the control system of an inverted pendulum ending in a metal ball with a controlled magnet hanging above the pendulum. In this document, we shall describe, develop and compare two popular strategies for controlling a closed-loop system. Furthermore, we shall describe the physics behind the system and implement animations displaying the controllers in action.

2 Problem Specification

This section describes the physical system and the control problem addressed in this project. Initially, the dynamics of the inverted pendulum and the electromagnetic actuation mechanism are introduced in order to define the underlying system physical behavior. Based on that, a mathematical model of the system is proposed and linearized around the upright equilibrium point to obtain a representation suitable for controller design.

The purpose of this section is to define the problem to be solved, including the system states, inputs, outputs, and stability properties. This provides the essential basis for the controller design and further analysis.

2.1 Pendulum Dynamics

2.1.1 Inverted Pendulum Model

One of the major components of the designed system is an inverted pendulum. The pendulum consists of a rod with an attached metal ball. The rod is assumed to be rigid and of fixed length l, which imposes a holonomic constraint on the system [15]. As a result, the distance between the pivot point and the mass remains constant what restricts the motion of the pendulum to one dimension. Therefore, the system with one degree of freedom can be described by the angular position of the pendulum.

The angular deviation of the pendulum is denoted by ϕ(t) and measured with respect to the vertical direction. Due to the holonomic constraint, the position of the pendulum mass can be determined by this angular coordinate.

2.1.2 State Variables

Since the pendulum dynamics is determined by a second-order differential equation, the system is represented in state-space form using the following state variables

$$x_1 = \phi \quad \text{(angular position)} \tag{1}$$

$$x_2 = \dot{\phi} \quad \text{(angular velocity)} \tag{2}$$

These two variables and provide the minimal set of states required to fully describe the system's motion.

2.1.3 Dynamics and Equation of Motion

The motion of the pendulum is influenced by gravity. The gravitational force acting on the mass can be decomposed into radial and tangential components relative to the circular trajectory [18]. Due to the rigid rod, radial motion is prohibited by the holonomic constraint. Due to that, the radial component of the force is balanced by the reaction force of the rod. Therefore, only the tangential component of gravity contributes to the pendulum's motion.

The tangential gravitational force is given by

$$F_t = mgsin\phi \tag{3}$$

Applying Newton's second law along the tangential direction yields

$$F_t = ma_t \tag{4}$$

where the tangential acceleration for circular motion is related to the angular acceleration by

$$a_t = l\ddot{\phi} \tag{5}$$

Substituting these expressions leads to the nonlinear equation of motion

$$\ddot{\phi} = \frac{g}{l} sin\phi \tag{6}$$

2.2 State-Space Model

Using the defined state variables, the nonlinear dynamics of the inverted pendulum can be expressed in state-space form as

$$\dot{x}_1 = x_2 \tag{7}$$

$$\dot{x}_2 = \frac{g}{l} sin(x_1) \tag{8}$$

This nonlinear state-space model serves as the basis for further extension with an electromagnetic actuator.

2.3 Electromagnetic Actuation

2.3.1 Magnetic Actuation Concept

The electromagnetic actuator is introduced into the system to stabilize the inverted pendulum. The actuator generates a magnetic force that acts on the pendulum mass, producing a control torque around the pivot. The magnetic actuation principle is inspired by electromagnetic suspension (EMS) systems, where force is controlled through the current of an electromagnet [5]. However, the EMS model is not used as a full mechanical system in this project. This project shares the actuator concept with EMS, which captures the nonlinear dependence of magnetic force on current and distance.

2.3.2 Introducing Inertia

Previously, the inverted pendulum dynamics was introduced in the compact form with cancelled out inertia term

$$\ddot{\phi} = \frac{g}{l} sin(\phi) \tag{9}$$

This expression implicitly assumes a point mass m located at the end of a massless rod of length l. Under this assumption, the moment of inertia of the pendulum with respect to the pivot is

$$I = ml^2 \tag{10}$$

Considering the case where only gravity is taken into account, the inertia term cancels out during the derivation. That is why it does not appear explicitly in the final equation of motion. However, this simplified form is no longer sufficient when adding actuation torque. The pendulum dynamics must be expressed at the level of rotational motion using Newton's second law [11] to correctly incorporate multiple torque contributions

$$\tau = I\ddot{\phi} \tag{11}$$

This formulation makes the role of the system inertia explicit and ensures that both gravitational and magnetic torques are properly incorporated into the resulting angular acceleration.

2.3.3 Gravitational Torque

The gravitational force acting on the pendulum mass is

$$F_g = mg \tag{12}$$

which generates a torque about the pivot given by

$$\tau_g = mglsin(\phi) \tag{13}$$

This torque destabilizes the upright equilibrium by increasing the angular deviation.

2.3.4 Magnetic Torque

The magnetic force generated by the electromagnet is modeled as

$$F_m(d, i) = \frac{ki^2}{d^2} \tag{14}$$

where:

- i - is the electromagnet coil current
- d - is the distance between the magnet and the pendulum mass
- k - is the constant accounting for various conditions determined for each experiment

This force generates a torque about the pivot. Assuming the magnetic force acts approximately perpendicular to the pendulum rod within the operating region, the magnetic torque is expressed as

$$\tau_m = F_m l \tag{15}$$

The direction of this torque is chosen such that it counteracts the destabilizing gravitational torque.

2.3.5 Updated Pendulum Dynamics

Including both gravitational and magnetic torques, the rotational equation of motion becomes

$$I\ddot{\phi} = mglsin(\phi) + F_m l \tag{16}$$

Substituting $I = ml^2$ and dividing by ml^2 gives

$$\ddot{\phi} = \frac{g}{l}sin(\phi) + \frac{F_m}{ml} \tag{17}$$

Substituting the magnetic force expression specified earlier results in

$$\ddot{\phi} = \frac{g}{l}sin(\phi) + \frac{k}{ml}\frac{i^2}{d^2} \tag{18}$$

This is an updated equation, which describes the nonlinear dynamics of an inverted pendulum stabilized by magnetic actuation.

2.3.6 Dynamics of Electromagnetic Actuator

The electromagnet coil dynamics are modeled as an RL circuit [13]

$$L_c\dot{i} = -Ri + u \tag{19}$$

where:

- L_c - is the coil inductance
- $\dot{i}$ - is speed of current change
- R - is the coil resistance
- i - is the coil current
- u - control voltage

This equation captures the dynamic relationship between the control input and the magnetic force generation.

2.3.7 Updated State-Space Model

The set of state variables extends with coil current

$$x_1 = \phi \quad \text{(angular position)} \tag{20}$$

$$x_2 = \dot{\phi} \quad \text{(angular velocity)} \tag{21}$$

$$x_3 = i \quad \text{(coil current)} \tag{22}$$

Combining newly introduced state variables with pendulum and electromagnetic actuator dynamics results in the following nonlinear state-space model

$$\dot{x}_1 = x_2 \tag{23}$$

$$\dot{x}_2 = \frac{g}{l} sin(x_1) + \frac{k}{ml} \frac{x_3^2}{d^2} \tag{24}$$

$$\dot{x}_3 = -\frac{R}{L_c} x_3 + \frac{1}{L_c} u \tag{25}$$

2.4 System Linearization

The nonlinear state-space model derived for the electromagnetically actuated inverted pendulum accurately represents the physical behavior of the system. However, the presence of nonlinear terms causes the system dynamics to depend on the operating (equilibrium) point, which makes the standard linear control design techniques inapplicable.

In this project, the objective is to stabilize the pendulum in the upright position, where only small deviations from the equilibrium are expected. Linearization provides a local approximation of the system dynamics around a specified equilibrium point, resulting in a linear time-invariant model with constant parameters. This linear representation is essential for controller design, particularly for the state vector control in the LQR controller and for the tuning and analysis of the classical PID controller.

2.4.1 Equilibrium point

The nonlinear system can be written as the output of two inputs

$$\dot{x} = f(x, u) \tag{26}$$

An equilibrium point $(\bar{x}, \bar{y})$ is defined as a constant state with input pair derivatives equal to zero

$$f(\bar{x}, \bar{y}) = 0 \tag{27}$$

As the goal is to stabilize the pendulum at an upright position, the equilibrium point should be

$$\bar{x} = \begin{bmatrix} 0 \\ 0 \\ i_0 \end{bmatrix} \tag{28}$$

where the first state represents the pendulum angle, the second - angular velocity, and i_0 denotes the steady-state coil current of the magnetic actuator. The corresponding steady-state control input voltage is given by

$$\bar{u} = Ri_0 \tag{29}$$

where R represents the magnet coil resistance. That ensures that the coil current remains constant at the equilibrium point.

2.4.2 Linearization about the Equilibrium Point

To analyze the system behavior near the operating point, the nonlinear function $f(x, u)$ is expanded in a first-order Taylor series about $(\bar{x}, \bar{u})$ [6]

$$\dot{x} \approx f(\bar{x}, \bar{u}) + \left.\frac{\partial f}{\partial x}\right|_{\bar{x},\bar{u}} (x - \bar{x}) + \left.\frac{\partial f}{\partial u}\right|_{\bar{x},\bar{u}} (u - \bar{u}) \tag{30}$$

Since $f(\bar{x}, \bar{u}) = 0$ at equilibrium, the linearized system becomes

$$\dot{x} = A(x - \bar{x}) + B(u - \bar{u}) \tag{31}$$

with system matrices defined as

$$A = \left.\frac{\partial f}{\partial x}\right|_{\bar{x},\bar{u}}, B = \left.\frac{\partial f}{\partial u}\right|_{\bar{x},\bar{u}} \tag{32}$$

In this linearized form, the model describes the dynamics of the system around a specified operating point.

2.4.3 Computing Linearized System Matrices

In the previous section, the following nonlinear state equations were derived

$$\dot{x}_1 = x_2 \tag{33}$$

$$\dot{x}_2 = \frac{g}{l} sin(x_1) + \frac{k}{ml}\frac{x_3^2}{d^2} \tag{34}$$

$$\dot{x}_3 = -\frac{R}{L_c} x_3 + \frac{1}{L_c} u \tag{35}$$

The elements of the state matrix A are determined by computing partial derivatives of each state equation with respect to state variables and evaluating them at the operating point $\bar{x}$.

Differentiating the first equation gives

$$\frac{\partial \dot{x}_1}{\partial x} = [0 \quad 1 \quad 0] \tag{36}$$

From the second equation,

$$\frac{\partial \dot{x}_2}{\partial x_1} = \frac{g}{l} * cos(x_1)\Big|_{x_1=0} = \frac{g}{l}, \frac{\partial \dot{x}_2}{\partial x_2} = 0, \frac{\partial \dot{x}_2}{\partial x_3} = \frac{2ki_0}{mld^2} \tag{37}$$

The third state equation result in

$$\frac{\partial \dot{x}_3}{\partial x} = [0 \quad 0 \quad -\tfrac{R}{L_c}] \tag{38}$$

Hence, the resulting linearized state matrix A is

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g}{l} & 0 & \frac{2ki_0}{mld^2} \\ 0 & 0 & -\frac{R}{L_c} \end{bmatrix} \tag{39}$$

Similarly, the input matrix B is obtained through partial differentiation with respect to the control input

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_c} \end{bmatrix} \tag{40}$$

The resulting linear time-invariant (LTI) model describes the local dynamics of the system around the specified equilibrium point. It captures the behavior of small deviations in angle, angular velocity, and coil current. This linearized model is a basis for further controller design.

2.5 Output Definition

Following the linearization, the inverted pendulum system is described by linear state-space model

$$\dot{x} = Ax + Bu, x = \begin{bmatrix} \phi \\ \dot{\phi} \\ i \end{bmatrix} \tag{41}$$

The output equation should be based on available measurements. In this project, the pendulum angle is measured at the pivot point. Therefore, the system output is specified as

$$y = \phi \tag{42}$$

Since the output corresponds directly to the first state variable, the linear output equation can be written as

$$y = Cx, C = [1 \quad 0 \quad 0] \tag{43}$$

The direct feedthrough matrix D is set to zero, since the measured output does not depend directly on the control input. The complete linear state-space representation of the system is therefore given by

$$\begin{cases} \dot{x} = Ax + Bx \\ y = Cx \end{cases} \tag{44}$$

The defined output is used as the feedback signal for the PID controller and for performance evaluation in the LQR-based control scheme.

2.6 System Stability

The control voltage u indirectly affects the output angle ϕ, which makes the system weakly coupled and causes divergence when applying control strategies. This can be further checked with the system poles. From the linearized system equation, we get the state and model:

$$x = [\phi, \dot{\phi}, i]$$

$$\ddot{\phi} = \frac{g}{l}\phi + \frac{2ki_0}{mld^2}i$$

$$\dot{i} = -\frac{R}{L_c}i + \frac{u}{L_c}$$

After we apply the Laplace transformation, we can derive a transfer function:

$$s^2\Phi(s) = \frac{g}{l}\Phi(s) + \frac{2ki_0}{mld^2}I(s)sI(s) =$$

$$sI(s) = \frac{R}{L_c}I(s) + \frac{U(s)}{L_c}$$

$$G(s) = \frac{\Phi(s)}{U(s)} = \frac{\frac{2ki_0}{mld^2L_c}}{(s^2 - \frac{g}{l})(s + \frac{R}{L_c})}$$

resulting in poles:

$$s_1 = -\frac{R}{L_c} = -200$$

$$s_{2/3} = \pm\sqrt{\frac{g}{l}} \approx \pm 5.72$$

Poles s_1 and s_2 are stable, but s_3 introduces instability, because it has a positive real part. Because $\frac{g}{l}$ can never be an imaginary number, this open-loop linearized system can not achieve stability, regardless of the parameters. Therefore, the design of control is essential to stabilize the system.

2.7 Open-loop System Behavior

In this section, the open-loop behavior of the electromagnetically actuated inverted pendulum is analyzed using continuous-time models. Both the nonlinear model and its linearized approximation are considered in order to gain insight into the dynamic properties of the system prior to controller design.

All simulations are performed in continuous time. This is motivated by the physical nature of the system, whose mechanical and electrical dynamics are inherently continuous and naturally described by differential equations. The objective is to analyze system stability rather than to address issues related to digital implementation. Therefore, discretization is intentionally omitted.

2.7.1 Open-loop Response of Nonlinear Model

The nonlinear state-space model in Figure 1 describes the full system dynamics, including the nonlinear coupling between the pendulum angle and the electromagnetic actuator. The model is simulated in open-loop configuration with zero input voltage applied to the actuator. A small initial angular deviation from the upright equilibrium is introduced.

The simulation results show that the pendulum does not return to the upright position. Instead, the angular deviation grows over time, indicating that the upright equilibrium is inherently unstable. This behavior is observed even for small initial disturbances, confirming that passive stabilization is not possible and that active control is required.

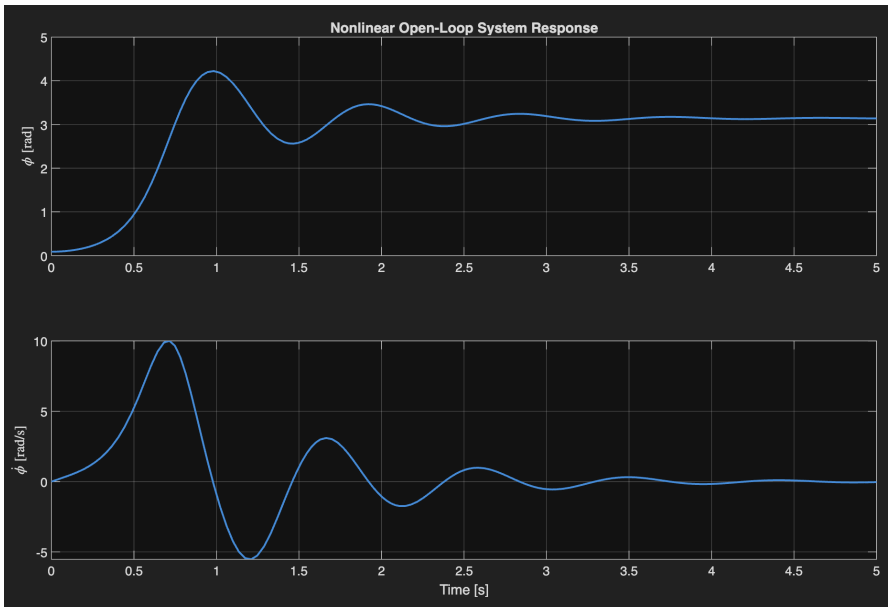

Figure 1: Nonlinear Open-loop Response

2.7.2 Open-loop Response of Linearized Model

To facilitate controller design, the nonlinear model is linearized around the upright equilibrium under the assumption of small angular deviations and zero steady-state current. The resulting linearized state-space model in Figure 2 provides a local approximation of the system dynamics.

The open-loop response of the linearized model is simulated using the same initial conditions as in the nonlinear. The results are consistent with instability of the nonlinear model, as the system states indicate the exponential growth.

The linearization does not change the fundamental stability properties of the system. Instead, it provides a simplified approximation that captures the local dynamics near the equilibrium point, suitable for applying linear control design techniques.

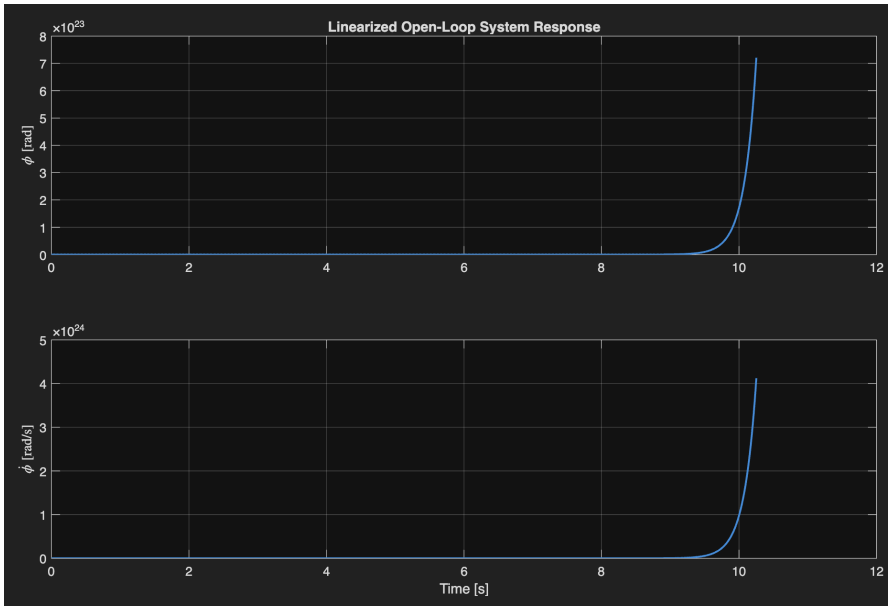

Figure 2: Linearized Open-loop Response

2.7.3 Implications for the Project

Both analyzed nonlinear and linear continuous-time models demonstrated unstable open-loop behavior around the upright equilibrium. In comparison with nonlinear model, which describes the full system dynamics, the linearized model represents the local behavior in the vicinity of the equilibrium point.

The consistency between the two models behavior justifies the use of the linearized representation for subsequent controller design. At the same time, the observed instability motivates the need for feedback control, which is addressed in the following sections.

3 Evaluation Criteria

The project's success is evaluated based on objective and quantitative criteria that assess both the stability and performance of the controlled system. Firstly, closed-loop stability is required for all implemented controllers, ensuring that the inverted pendulum can be stabilized around its upright equilibrium point.

Secondly, the performance of the transient response is evaluated using metrics in the time-domain such as the settle time and the overshoot. These parameters indicate how quickly and smoothly the system stabilizes around the desired equilibrium.

Thirdly, the required control effort is assessed using quantitative measures that include the maximum current, the root-mean-square (RMS) current, and the integrated squared current. These metrics provide insight into actuator energy efficiency.

Finally, controller performance is compared under the same model and initial conditions to ensure a fair and consistent evaluation. The project is considered successful if the implemented controllers stabilize the system and allow a quantitative comparison based on specified criteria.

3.1 Quantifiable control performance metrics

In order to compare the two approaches - LQR and PID, and their overall optimization quality, we will use specific measurable criteria that determine how well the system behaves after a disturbance in terms of reaching a steady state.

1. **Stability** - The ability of the system to return to its equilibrium state after a disturbance. [12]

2. **Settling time** - The time needed for the system to settle within a certain percentage of its final value.[12]
3. **Overshoot** - The maximum amount by which the system's response exceeds its final value.[12]
4. **Control effort** - The actuator demand required to stabilize the system, quantified in terms of coil current. [1]
5. **RMS current** - captures the average intensity of controller operation during simulation. [8]
6. **Peak current** - captures the highest possible current load during operation [8]

4 Setup

In this section, the technology stack specification is provided for this project. First of all, this document was created and developed in Overleaf. This online tool facilitates the creation of formulas and a document outline, which significantly reduces the time required for report editing.

For implementing and testing the developed system model, MATLAB was used. Furthermore, we are using Simulink to build block diagrams for the system's iterations and controllers. Based on these simulations, we create interactive animations for both controllers in 2D using MATLAB's built-in graphics drawing features. Furthermore, a 3D interactive playground for the LQR control mechanism was created using Three JS and Typescript. The playground could be accessed via https://systems-control.vercel.app/.

Project report, MATLAB scripts, Simulink models and animations are shared on Github repositories of the authors:

https://github.com/BRISINGR-01/Maths/tree/main/Systems Control.

https://github.com/xtr13m3/Magnetic actuation

5 Approach

This section provides a detailed overview of the essential theoretical background and details of PID and LQR controllers models implementation.

5.1 PID

5.1.1 Definition

A pid (pid) controller computes the control signal for a plant based on the sse of a closed-loop-system. It maps the error signal to an actuator input using the following formula:

$$u(t) = K_p\, e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{45}$$

where:

- $u(t)$ is the controller output (control input to the plant),
- $e(t)$ is the sse,
- K_p is the proportional gain,
- K_i is the integral gain,
- K_d is the derivative gain.

The gains K_p, K_i, and K_d are constant parameters tailored to the specific system. PID reduces sse, but has a negative effect on the stability.

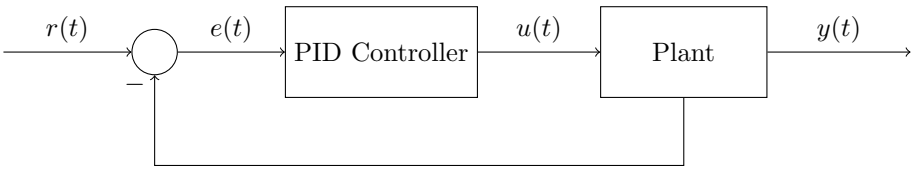

Figure 3: PID control system block diagram

5.1.2 Gains

Proportional Gain

K_p defines an output proportional to the error. A large error causes a large response and vice versa. If K_p is increased, then the feedback is more sensitive to the state error. Therefore, a large proportional gain can cause overshoot. If K_i and K_d are not present, then the system typically doesn't reach the desired state and leaves an offset between the actual and desired state.

Integral Gain

From a time perspective, if the proportional gain is viewed as the present, then the integral term represents the past - the previous state errors accumulate and affect the outcome. This fixes the problem with the offset, however it can also cause oscillation and slow down the response time.

Derivative Gain

K_d is not always used, but it can help dampen overshoot and oscillation. It predicts the error change as a derivative, depending on how quickly the error changes. It is highly susceptible to noise (hence why it is sometimes omitted).

5.2 PID Implementation

The open-loop stability analysis indicated that the linearized system contains an unstable pole associated with the mechanical dynamics of the inverted pendulum. In addition, the control voltage affects the pendulum angle only indirectly through the coil current, which introduces internal electrical dynamics. As a result, the system indicates a weak coupling between the control input and the mechanical output.

This structure has important implications for controller design. In particular, it suggests that directly applying a single-loop PID controller from the pendulum angle to the actuator voltage may lead to poor closed-loop performance or instability. To investigate this limitation, a classical single-loop PID controller is first implemented and evaluated.

5.2.1 Single-loop PID Control

In the single-loop PID configuration, the pendulum angle is used as the feedback signal. Based on that feedback, the controller directly supplies the actuator voltage. The controller structure follows the classic PID design described in the previous section.

The controller is applied to the linearized continuous-time model of the system, and the closed-loop response is evaluated through time-domain simulations.

The simulation results in Figure 4 show that the single-loop PID controller does not stabilize the system robustly. Depending on the chosen gains, the response either diverges or exhibits excessively large oscillations and control effort. This behavior is caused by the system architecture itself rather than poor tuning. The electromagnetic actuator introduces an additional integrator between the control input and mechanical states. That significantly limits the effectiveness of a single-loop PID design.

Based on these observations, the single-loop PID controller is unsuitable for the considered system.

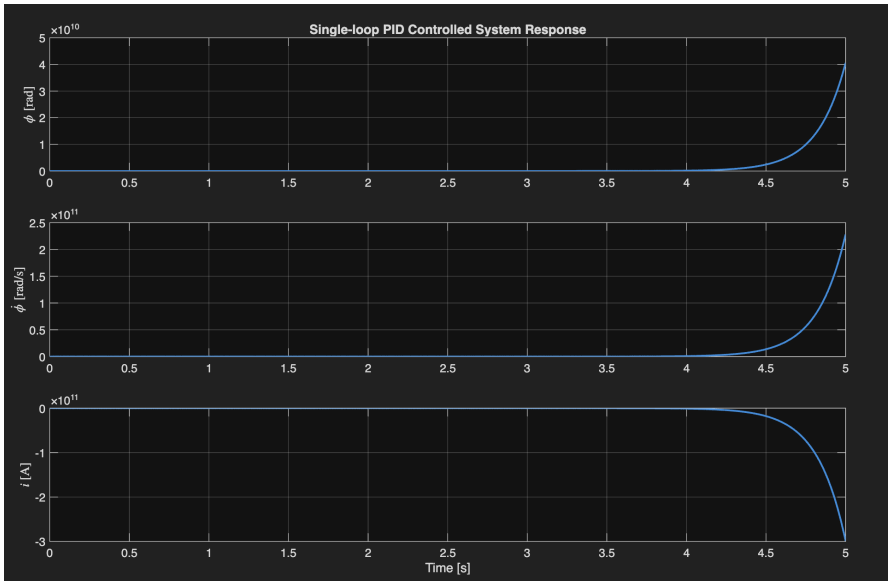

Figure 4: Single-loop PID control

5.2.2 Cascaded PID Control

The analysis of the single-loop PID control in previous section indicates that this design does not suit the purpose of pendulum stabilization. That motivates the consideration of a cascaded PID architecture. This design splits the system into two subsystems: an inner electrical loop and an outer mechanical loop. This aligns with the physical structure of the system and allows each subsystem to be controlled independently.

In cascaded PID architecture in Figure 5, the inner loop controls the coil current using a PI controller what ensures stable electrical dynamics. The outer loop controls the pendulum angle using a PD controller and sets the current reference for the inner loop [19]. Such a separation of the system effectively improves the coupling between the control input and the mechanical states and addresses the instability identified in the open-loop analysis.

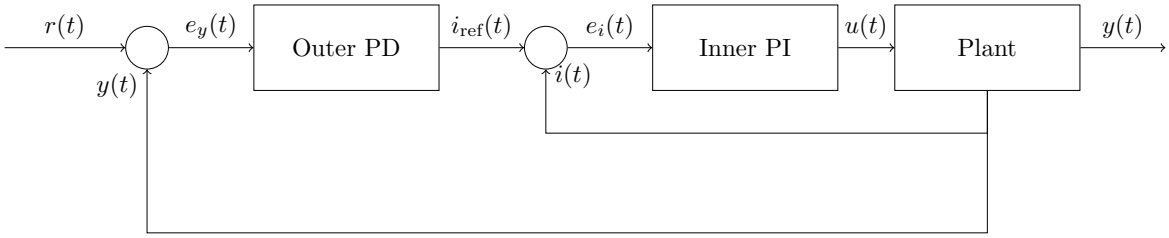

Figure 5: Cascaded PID control block diagram

5.2.3 MATLAB implementation

The term appearing in the linearized state matrix that couples the coil current to the angular acceleration is

$$\frac{2ki_0}{mld^2}. \tag{46}$$

This term represents the linearized effect of the electromagnetic actuation on the pendulum dynamics. For convenience, this coupling term is denoted by the parameter α in the subsequent controller implementation, such that

$$\alpha = \frac{2ki_0}{mld^2}. \tag{47}$$

The value of α depends on the system parameters and the chosen equilibrium point. For the parameter values considered in this project, α evaluates to approximately 2 and is therefore used as a constant parameter in the controller simulation. This simplification allows the focus to remain on the control design and performance comparison rather than on detailed parameter identification.

The cascaded PID controller is simulated by numerically integrating the closed-loop system dynamics using ode45. This MATLAB function solves the continuous-time state-space equations and provides the time-domain response of the system.

The extract of the script with the main logic:

```
% Params
g  = 9.81;
l  = 0.2;
R  = 4.0;
Lc = 0.02;

alpha = 2.0;

%% Controller gains
Kp_phi = 40;
Kd_phi = 8;

Kp_i = 20;
Ki_i = 400;

% Initial conditions
phi0     = 0.05;
phiDot0 = 0.0;
i0       = 0.0;
z_i0     = 0.0;

x0 = [phi0; phiDot0; i0; z_i0];

% Simulation
tspan = [0 5];

[t, x] = ode45(@(t,x) cascadedPID_ode(t,x,...
    g,l,R,Lc,alpha,...
    Kp_phi,Kd_phi,...
    Kp_i,Ki_i), tspan, x0);
```

5.2.4 Closed-loop Performance of Cascaded PID

The cascaded PID controller is implemented using the linearized continuous-time model. Time-domain simulations demonstrate that the cascaded structure successfully stabilizes the inverted pendulum around the upright equilibrium.

Compared to the single-loop PID, the cascaded controller in Figure 6 exhibits a well-damped response, reduced settling time, and bounded control effort. The pendulum angle converges smoothly to zero, and the current remains within acceptable limits throughout the transient.

To quantify the closed-loop behavior, standard performance metrics in Figure 7 are evaluated, including settling time, overshoot, peak current, and RMS current.

These results confirm that the cascaded PID architecture provides a significant improvement in stability and performance over the single-loop PID controller.

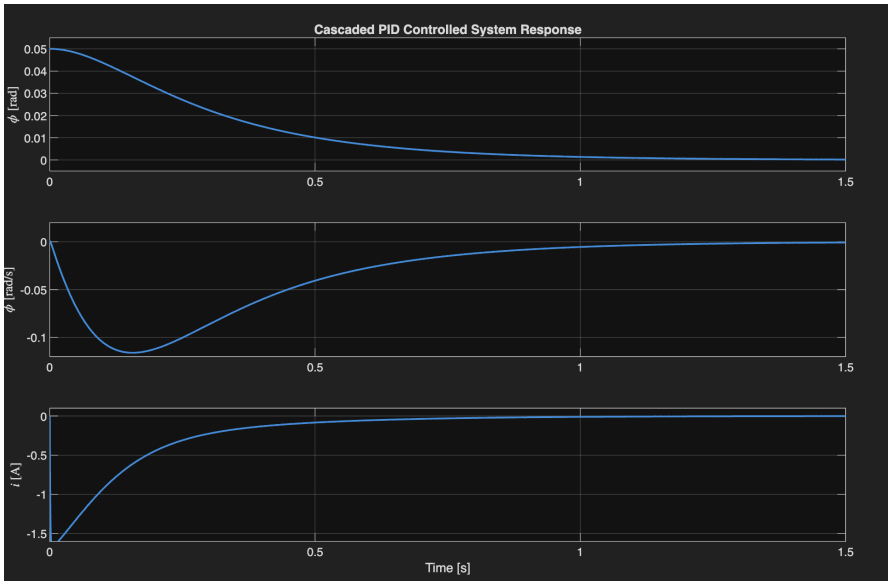

Figure 6: Closed-loop Cascaded PID

```
=== Cascaded PID Performance Metrics ===
    SettlingTime_s    Overshoot_percent    PeakCurrent_A    IntegratedCurrentSquared    RMS_Current_A
    ______________    _________________    _____________    ________________________    _____________

        1.0754            0.00068248           1.6766              0.23669                0.22984
```

Figure 7: Cascaded PID performance metrics

5.2.5 PID tuning

The PID gains were selected using an iterative simulation-based tuning procedure. For the single-loop PID controller, various combinations of proportional, integral, and derivative gains were tested. However, there was no set of gains able to provide robust stabilization due to the indirect coupling between the control input and the mechanical dynamics.

The cascaded PID controller was tuned sequentially. Initially, the inner current loop was tuned using a PI controller to achieve fast and stable electrical dynamics. Subsequently, the outer angle loop was tuned using a PD controller to ensure a well-damped mechanical response while keeping the required control effort within acceptable limits. Since the control architecture is cascaded, automatic tuning tools such as the MATLAB PID Tuner were not used. The PID tuner is designed for a single-loop controller configuration. The final gains were therefore selected to balance stability, transient performance, and actuator effort.

5.2.6 Implications for the Project

The investigation of PID-based control strategies shows that a classical single-loop PID controller is not suitable for stabilizing the electromagnetically actuated inverted pendulum due to the indirect coupling between the control input and the mechanical states. By contrast, the cascaded PID controller, motivated by the system stability analysis, effectively stabilizes the system and achieves satisfactory performance. However, the tuning of cascaded PID gains remains complex and system-specific. That motivates the exploration of a more systematic LQR controller in the next section.

5.3 LQR

5.3.1 Definition

The LQR is a control mechanism widely used in various fields, including aerospace, robotics, and industrial automation. It assumes that the system dynamics are linear and that the cost function

is quadratic, which might not be applicable for highly nonlinear systems or those with complex constraints. LQR is used in a closed-loop-system to calculate the optimal feedback-gain K, similarly to pole-placement.

The control-vector u can be obtained as follows:

$$u(t) = -Kx(t)$$

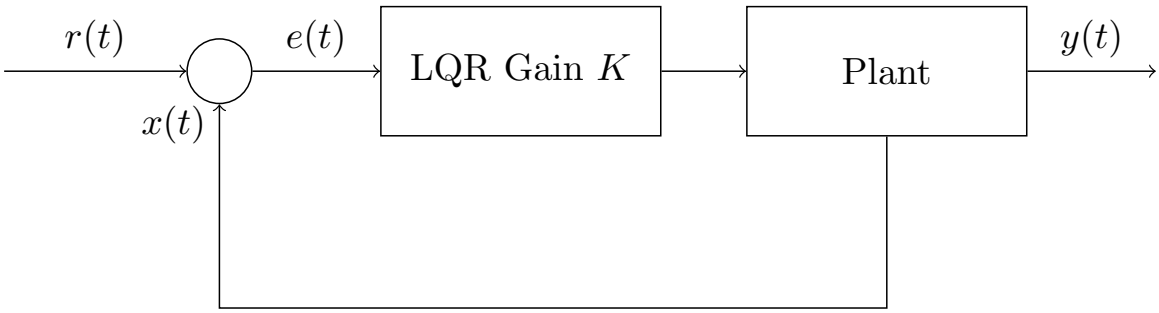

Figure 8: LQR state-feedback control block diagram

5.3.2 Cost function

What makes LQR so useful is the cost function used to determine how optimally the system behaves with respect to the state and the control-effort. It is calculated as follows:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt$$

where:

- x is the state-vector,
- u is the control-vector,
- Q is a positive semi-definite $m \times m$ matrix that penalizes deviations in the state variables where m is the number of states,
- R is a positive definite $n \times n$ matrix that penalizes the use of control-effort where n is the number of inputs

The system is the most optimal when **J** is the lowest.

5.3.3 Parameters

Since Q affects the state variable, increasing it leads to faster convergence, but higher control-effort. On the other hand, increasing R slows down convergence and also decreases the effort. For example, this means that larger values of Q would place the poles of the closed-loop-system matrix further left in the s-plane, resulting in the state decaying faster to zero.

5.3.4 Continuous-time Algebraic Riccati Equation (CARE)

An alternative to brute-forcing or using learning algorithms to find K, the *Riccati Equation* can be used to calculate the optimal feedback gain. It is derived from introducing a matrix P such that $P = P^T$. The following matrix equation arises:

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \tag{48}$$

By substituting the control vector u for $-Kx(t)$ and optimizing for u we find that:

$$K = R^{-1}B^T P$$

Since equation 48 is linear, this simplifies the analysis and implementation. Because it is quadratic, the graphical representation of the cost map has a convex shape, thus it has a calculable global minimum.

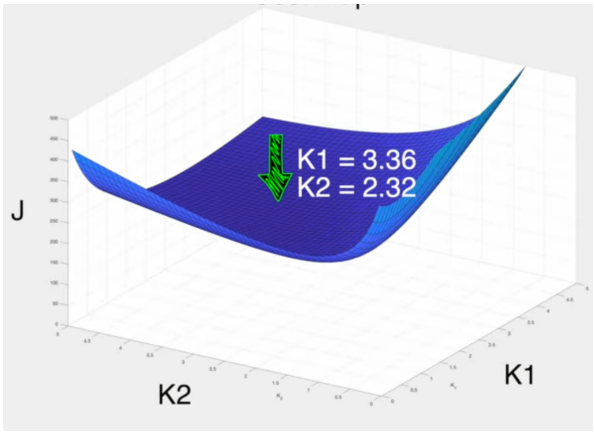

Figure 9: Example cost map

Picture 9 is simply an example of a graphical representation between J and K.

5.3.5 Stability

Similarly to the PID implementation, there were issues with computing the gains in Matlab, due to the initial instability of the system. To account for that the system was split in two subsystems. The first one is the electrical side of the system. It takes current as input and from its matrices we compute one gain K_i. We can think of it as an "inner" faster loop.

$$A_i = \frac{-R}{L_c} \qquad B_i = \frac{1}{L_c} \qquad x = i$$

$$C_i = 1 \qquad Q_i = 1 \qquad R_i = 1e-2$$

The outer loop takes care of the mechanical part and gives the other two gains $K_{o1/2}$.

$$A_o = \begin{bmatrix} 0, 1 \\ \frac{g}{l}, 0 \end{bmatrix} \qquad B_o = [0, \frac{2ki_0}{mld^2}]$$

$$C_o = [1, 1] \qquad Q_o = \begin{bmatrix} 100, 0 \\ 0, 1 \end{bmatrix} \qquad R_o = 1$$

Together they are used to define the final gain matrix $K = [K_{o1}, K_{o2}, K_i]$. Here is an example matlab script used to compute the gains and check of system stability using, the eigenvalues of the closed loop state matrix A_{cl}:

```
%% Inner Loop
Ai = -R/Lc;
Bi = 1/Lc;
Qi = 1;
Ri = 1e-2;
Ki = lqr(Ai, Bi, Qi, Ri);

%% Outer Loop
Ao = [0 1; g/l 0];
Bo = [0; alpha];
Qo = diag([100 1]);
```

```
Ro = 1;
Ko = lqr(Ao, Bo, Qo, Ro);

K = [Ko(1) Ko(2) Ki];

A = [ 0 1 0;g/l 0 alpha;0 0 -R/Lc ];
B = [0; 0; 1/Lc];

if all(real(eig(A-K*B)) >= 0)
    disp('SYSTEM IS UNSTABLE');
end
```

The system presented above is not guaranteed to be stable. If the Q and R values are changed or any of the parameters, it could cause the system to become unstable. To counter this, the k coefficient should be adjusted accordingly. For this project it seems fit to penalize voltage control by reducing R_i and increasing $Q_o(0,0)$ which controls ϕ to 0.01 and 100 respectively. The matlab result is as follows:

```
Inner current gain Ki = 6.7703
Outer gains Ko = [10.9315 1.2634]

Closed-loop poles:
 1.0e+02 *
 -5.3425 + 0.0000i
 -0.0213 + 0.0004i
 -0.0213 - 0.0004i
```

6 Simulation framework

The simulation framework consists of a set of MATLAB scripts and Simulink models used for controller design, system simulation, performance evaluation, and further visualization. To ensure reproducibility of results, a particular sequence of actions is specified further along with description of scripts, models and visualization of results.

6.1 Scripts

Considering MATLAB code, there are three main files:

The `upd_cascaded_PID.m` script implements the cascaded PID controller. It defines the system parameters, derives the linearized model, and specifies the gains for the inner and outer loops. The script also contains numerical simulation routines based on the ODE solver to validate the controller behavior prior to Simulink implementation. Furthermore, PID performance metrics are output to the console, and the system response is visualized in a plot.

The `upd_LQR.m` script implements LQR controller. It defines the state-space matrices of the system and computes the optimal state-feedback gain. The closed-loop stability is verified through eigenvalue analysis and time-domain simulations, performed using an ODE solver to obtain performance metrics. Similar to previously described script, `upd_LQR.m` also provides a console output with performance matrix and a plot visualizing system response.

Finally, `upd_compare_controllers.m` script loads results of both PID and LQR MATLAB simulations and provides a visual comparison of both control designs in terms of control effort, response time and energy-related performance indicators.

To ensure correct replication of simulation results, `upd_cascaded_PID.m` and `upd_LQR.m` should be ran prior to `upd_compare_controllers.m`. The latter script relies on values received from each controller individual simulation.

6.1.1 Miscellaneous scripts

During initial stage of the project realization, `Pendulum_nonlinear.m`, `linearized_pendulum.m`, `single_loop_PID.m` scripts simulating system's nonlinear, linearized non-actuated and single-loop PID behavior were developed as a step preceding the control design. These scripts provide an overview of the earlier steps in our approach. These scripts are also attached to the project report.

6.2 Simulink

The first model `PID_simulink.slx` represents a block diagram of the cascaded PID architecture proposed earlier in MATLAB script. The model contains the inner current control and outer angle control loops. The plant dynamics is represented using state-space block.

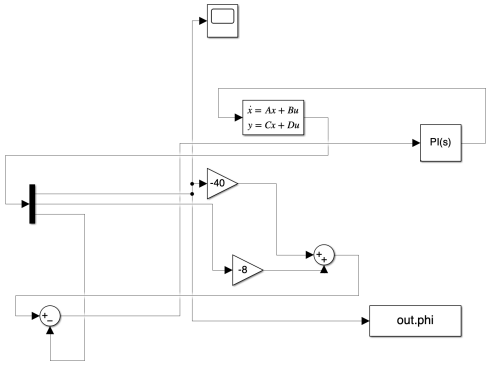

Figure 10: Simulink block diagram for PID

The second model `LQR_simulink.slx` represents a block diagram of the LQR architecture as a full state-feedback system. The same plant model is reused as for PID under identical conditions what allows a direct unbiased comparison between controllers.

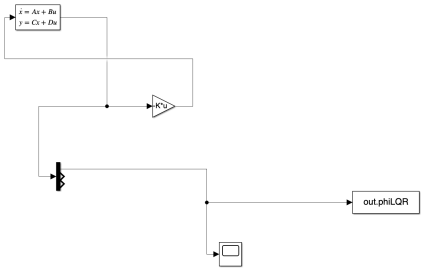

Figure 11: Simulink block diagram for LQR

Each model should be ran after the corresponding MATLAB script to generate the required values in the workspace. Both Simulink models are attached to this report.

6.3 Visualization 2D

Simulink outputs simulated values into the workspace for further post-processing and visualization. Afterwards, `PID_animation.m` and `LQR_animation.m` scripts load simulation output variables from the workspace. Based on these inputs, these scripts generate 2D graphics of the inverted pendulum with an actuator above it. Furthermore, the UI contains a button adding disturbance, which demonstrates controller's robustness.

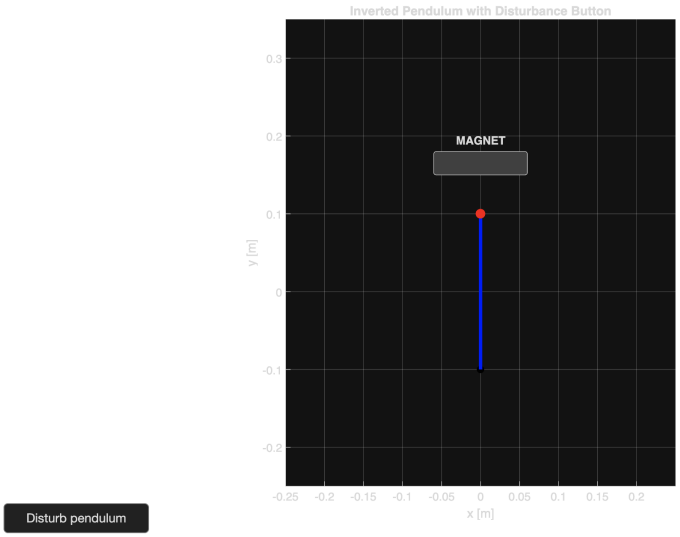

Figure 12: 2D animation of controller behavior

6.4 Visualization 3D

A second visualization 13 was created outside MATLAB as a web app. It is made using JavaScript, so the MATLAB native functions were not available. Furthermore, the environment was strictly digital and more restrained in terms of computational resources, unlike MATLAB, which attempts to replicate an analog signal. For this purpose, the Runge–Kutta method known as RK4 was applied to account for the difference in the environment. It was required since both models behaved erratically when put exactly as in MATLAB. So far, only the LQR method has been successfully stabilized. A detailed explanation of the development and issues of this system is out of scope for this document.

Figure 13: 3D Visualization for LQR

7 Results and analysis

This section presents a comparative evaluation of the PID and LQR controllers designed for the linearized electromagnetic inverted pendulum. Both controllers were assessed using identical system models, initial conditions, and performance indicators to ensure objective comparison.

7.1 Transient Response

Figure 14 illustrates the time-domain response of the pendulum angle for both controllers. The first two sections display the comparison of the angle and torque, respectively. The PID controller shows a faster reduction of the initial angular deviation, reaching the equilibrium point faster than the LQR controller. This behavior is reflected in the shorter settling time achieved by the PID controller.

In contrast, the LQR controller demonstrates a smoother transient response with a smoother reduction of angular deviation. Although the settling time is longer, the response remains well-damped. Both

controllers demonstrate negligible overshoot, indicating stable and non-oscillatory dynamics.

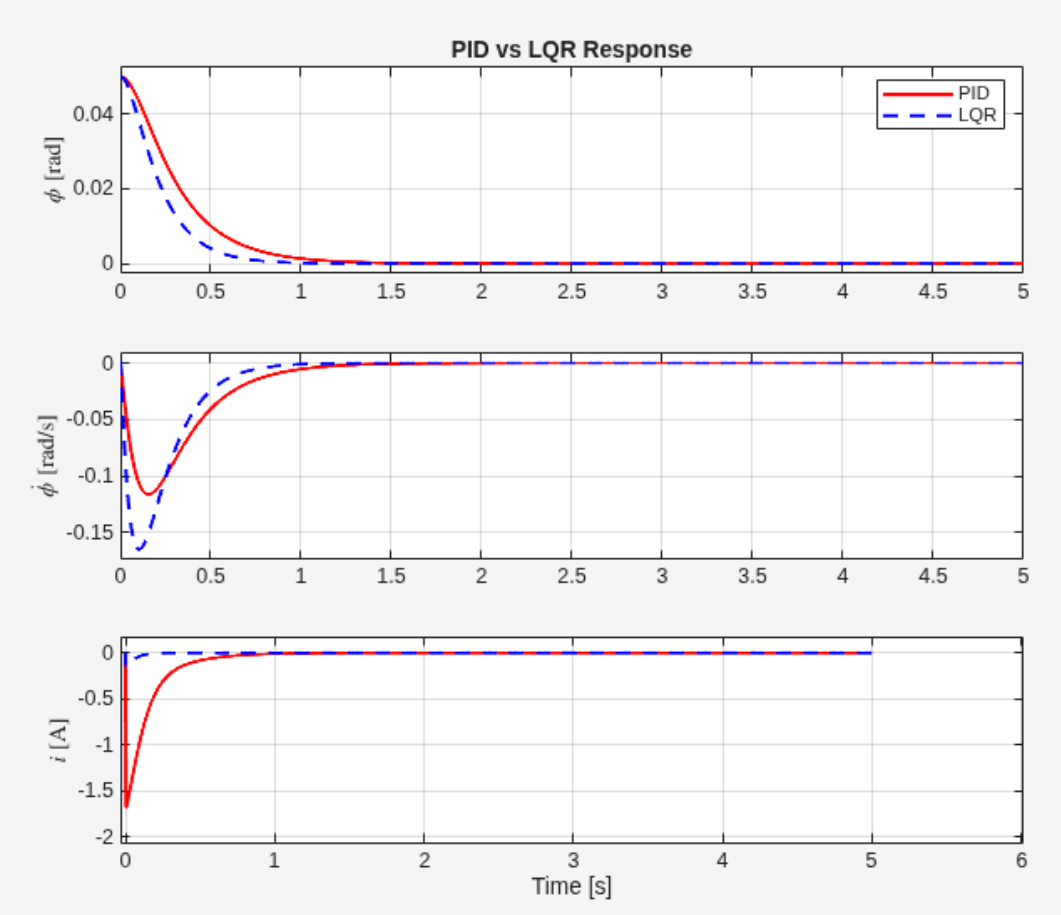

Figure 14: Controllers Comparison

7.2 Control Effort

The required control effort in terms of the coil current is shown in the third graph in Figure 14. A considerable difference between the two controllers can be observed. The PID controller applies a large initial current to rapidly counteract the disturbance of the pendulum, which results in a high peak current and higher average current levels throughout the transient phase.

In contrast, the LQR controller significantly reduces the magnitude of the applied current. The control action remains smooth and bounded, with peak current values smaller than those of the PID controller. This behavior reflects the optimization-based nature of the LQR approach, which balances state regulation against control effort.

7.3 Quantitative Performance Metrics

Figure 15 summarizes the quantitative performance metrics for both controllers. Based on this table, the PID controller has a shorter settling time, indicating faster stabilization. However, this advantage comes at the cost of substantially higher parameters related to current consumption.

Although the LQR controller is slower in terms of settling time, it demonstrates superior performance with respect to control efficiency. The reduced peak and average current values indicate lower actuator stress and improved energy efficiency.

```
--- Controller Performance Comparison ---
     SettlingTime_s    Overshoot_percent    PeakCurrent_A    RMS_Current_A
    ________________    _________________    _____________    _____________

PID      1.7619             0.003995            1.701           0.3328
LQR      2.3942             0.034813          0.050721          0.015467
```

Figure 15: Performance comparison

7.4 Analysis

The comparison of PID and LQR controllers indicates a trade-off between response time and required control effort. The PID controller stabilizes the system faster than the LQR. However, it requires more substantial control effort. This results in shorter settling time, along with large peak current and poor energy-related performance metrics.

In contrast, the LQR controller demonstrates considerably smoother response while requiring much less control input. Based on the nature of this type of controller, it limits the applied current to avoid excessive energy consumption. As a result, the system stabilizes around the equilibrium more slowly, but with significantly lower load on actuator.

Overall, the results indicate that neither controller could be named as a more favorable option to use. Faster stabilization provided by a PID controller could be applicable in systems where the response time is critical while applied current limitations are not. On the contrary, the LQR controller could be used in systems where energy efficiency and durability are the main focus. However, it comes at the cost of a slower transient response.

8 Conclusions

In this project, we studied and applied fundamental concepts from classical and modern control theory using a magnetically actuated inverted pendulum as a case study. The work provided practical insight into magnetic actuation, a technology that is increasingly relevant in modern engineering applications. A key learning outcome was the ability to translate physical system behavior into mathematical models derived from first principles of physics, and to use these models as a basis for controller design and analysis.

Prior to starting the project realization, the evaluation criteria were specified in order to have objective and quantitative parameters to assess whether the project achieved its goals. We were able to develop nonlinear and linearized models, which accurately describe the dynamics of the actuated inverted pendulum. Based on these models, PID and LQR controllers were designed, which stabilize the system around the desired equilibrium point. That was a sound basis for further quantitative comparison of both controllers. The comparison was conducted under the identical initial conditions what ensured unbiased results. Consequently, the performance difference was analyzed and presented in the results section of this report. As a result, the evaluation criteria were met.

As concrete deliverables of this project, we developed MATLAB scripts for a nonlinear and linearized inverted pendulum system actuated by an electromagnet, single-loop and cascaded PID, LQR controller and controller performance comparison script. For cascaded PID and LQR, we created Simulink models of the designed control approaches. Based on these scripts and models, we implemented a 2D interactive animation in MATLAB and 3D animation using a web app in order to visualize the behavior. The accompanying report consolidates the theoretical background, modeling approach, controller design, and quantitative comparison, providing a complete overview of the project accomplishment process.

References

[1] Michal Bartys and Bartolomiej Hryniewicki. "The Trade-Off between the Controller Effort and Control Quality on Example of an Electro-Pneumatic Final Control Element". In: *Actuators* (2019). URL: https://www.mdpi.com/2076-0825/8/1/23.

[2] James Bennett et al. "PID Tuning Via Classical Methods". In: *Wayback Machine* (2006). URL: https://web.archive.org/web/20080616062648/http://controls.engin.umich.edu/wiki/index.php/PIDTuningClassical#Ziegler-Nichols_Method.

[3] Eureka Patsnap. "What is a Linear Quadratic Regulator (LQR)". In: *eureka.patsnap* (2025). URL: https://eureka.patsnap.com/article/what-is-a-linear-quadratic-regulator-lqr.

[4] F.L. Lewis. *Linear Quadratic Regulator (LQR) State Feedback Design*. 2019. URL: https://lewisgroup.uta.edu/Lectures/lqr.pdf.

[5] Xinrui He. "Magnetic Levitation: Electrodynamic Suspension and Electromagnetic Suspension Technologies". In: *Symposium: Advances in Sustainable Aviation and Aerospace Vehicle Automation* (2026).

[6] Kamran Iqbal. *1.7: Linearization of Nonlinear Models*. 2025. URL: https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Introduction_to_Control_Systems_(Iqbal)/01%3A_Mathematical_Models_of_Physical_Systems/1.07%3A_Linearization_of_Nonlinear_Models.

[7] Dr. Vipin Jain. "Steady State Error: What is it? (Steady-State Gain, Value and Formula)". In: *electrical4u* (2024). URL: https://www.electrical4u.com/steady-state-error-analysis/.

[8] Anthony Lodi. *Design 2: Peak vs RMS Current*. 2020. URL: https://www.ti.com/content/dam/videos/external-videos/ja-jp/7/3816841626001/6182578228001.mp4/subassets/peak_vs_rms_current.pdf.

[9] Bill Messner and Dawn Tilbury. *Introduction: PID Controller Design*. Accessed: 2025-01-07. 2025. URL: https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID.

[10] MicrocontrollersLab.com. *PID Controller Design using Simulink MATLAB*. 2021. URL: https://microcontrollerslab.com/pid-controller-design-simulink/.

[11] Isaac Newton. *40: Newton's Laws of Motion- Rotational Versions*. 2026. URL: https://phys.libretexts.org/Courses/Prince_Georges_Community_College/General_Physics_I%3A_Classical_Mechanics/40%3A_Newtons_Laws_of_Motion-_Rotational_Versions.

[12] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall, 2010.

[13] LibreTexts Physics. *14.5: RL Circuits*. 2026. URL: https://phys.libretexts.org/Bookshelves/University_Physics/University_Physics_(OpenStax)/University_Physics_II_-_Thermodynamics_Electricity_and_Magnetism_(OpenStax)/14%3A_Inductance/14.05%3A_RL_Circuits.

[14] Muhammad Sufyan. "PID Explained: Theory, Tuning, and Implementation of PID Controllers". In: *Wevolver* (2025). URL: https://www.wevolver.com/article/pid-explained-theory-tuning-and-implementation-of-pid-controllers.

[15] Jeremy Tatum. *Holonomic Constraints*. 2025. URL: https://phys.libretexts.org/Bookshelves/Classical_Mechanics/Classical_Mechanics_(Tatum)/13%3A_Lagrangian_Mechanics/13.03%3A_Holonomic_Constraints.

[16] The MathWorks. *What Is Linear Quadratic Regulator (LQR) Optimal Control? — State Space, Part 4*. 2023. URL: https://www.youtube.com/watch?v=E_RDCFOlJx4&t=5s.

[17] The MathWorks. *Why the Riccati Equation Is important for LQR Control*. 2023. URL: https://www.youtube.com/watch?v=ZktL3YjTbB4.

[18] Duke University. *The Pendulum Differential Equation*. 2025. URL: https://sites.math.duke.edu/education/postcalc/predprey/pendback.html.

[19] Zhao Wang et al. *Cascade Digital PID Control Design for Power Electronic Converters*. 2022. URL: https://nl.mathworks.com/company/technical-articles/cascade-digital-pid-control-design-for-power-electronic-converters.html.

[20] Whiteknight. *Control Systems*. Wikibooks, 2006. URL: https://en.wikibooks.org/wiki/Control_Systems.

[21] J. G. Ziegler and Nathaniel B. Nichols. "Optimum Settings for Automatic Controllers". In: *Journal of Fluids Engineering* (1942). URL: https://api.semanticscholar.org/CorpusID:41336178.