Uma imagem com Tipo de letra, logótipo, Gráficos, captura de ecrã

Descrição gerada automaticamente

Instituto Superior Técnico

Distributed Predictive Control and Estimation

MEEC

Laboratory Report

**Group: 18**

**Authors:**

Alexandre Leal (103098)alexandre.leal@tecnico.ulsiboa.pt

Diogo Sampaio (103068) diogo.sampaio@tecnico.ulisboa.pt

Francisco Tavares (103402)

Marta Valente (103574)

[francisco.carreira.tavares@tecnico.ulisboa.pt](mailto:francisco.carreira.tavares@tecnico.ulisboa.pt)

[marta.valente@tecnico.ulisboa.pt](mailto:marta.valente@tecnico.ulisboa.pt)

2nd Semester – 4th Quarter – 2024/2025

*The group of students identified above guarantees that the text of this report and all the software and results delivered were entirely carried out by the elements of the group, with a significant participation of all of them, and that no part of the work or the software and results presented was obtained from other people or sources.*

## **P1 – Basic on Constrained Optimization**

For the vector , the Rosenbrock function is defined as:

(1)

As illustrated in *Figure 1*, the Rosenbrock function exhibits a narrow and curved valley, with a single global minimum at:

(2)

We will compute a cost function (both with and without constraints) in MATLAB, aiming to identify its minimum and visualise the results using clear and informative graphical outputs.

**Unconstrained Minimum**

Since the function is non-negative, any point at which must correspond to a global minimiser. By setting each squared term in to zero, we obtain:

(3)

This is the only stationary point, which can be verified by evaluating the gradient:

(4)

Subsequently, the quasi-Newton method was applied using MATLAB’s fminunc function, with an initial guess of:

(5)

The optimisation converged to the point:

(6)

Which closely approximates the theoretical minimum, confirming the effectiveness of the method.

**Constrained Minimum**

Afterwards, we impose a constraint on the search space:

(7)

As there are no stationary points within the feasible region defined by this constraint, the constrained minimiser must lie on the boundary . Applying the method of Lagrange multipliers with:

(8)

We solve:

(9)

To confirm this result, we applied MATLAB’s *fmincon* function with an initial guess of ), which returned:

(10)

Once again, the solution closely matches the theoretical value, confirming the validity of the analytical approach.

**Graphical Interpretation**

For a better understanding of the constraint effect, we generated these three figures:

Uma imagem com Saturação de cores, captura de ecrã, Gráficos

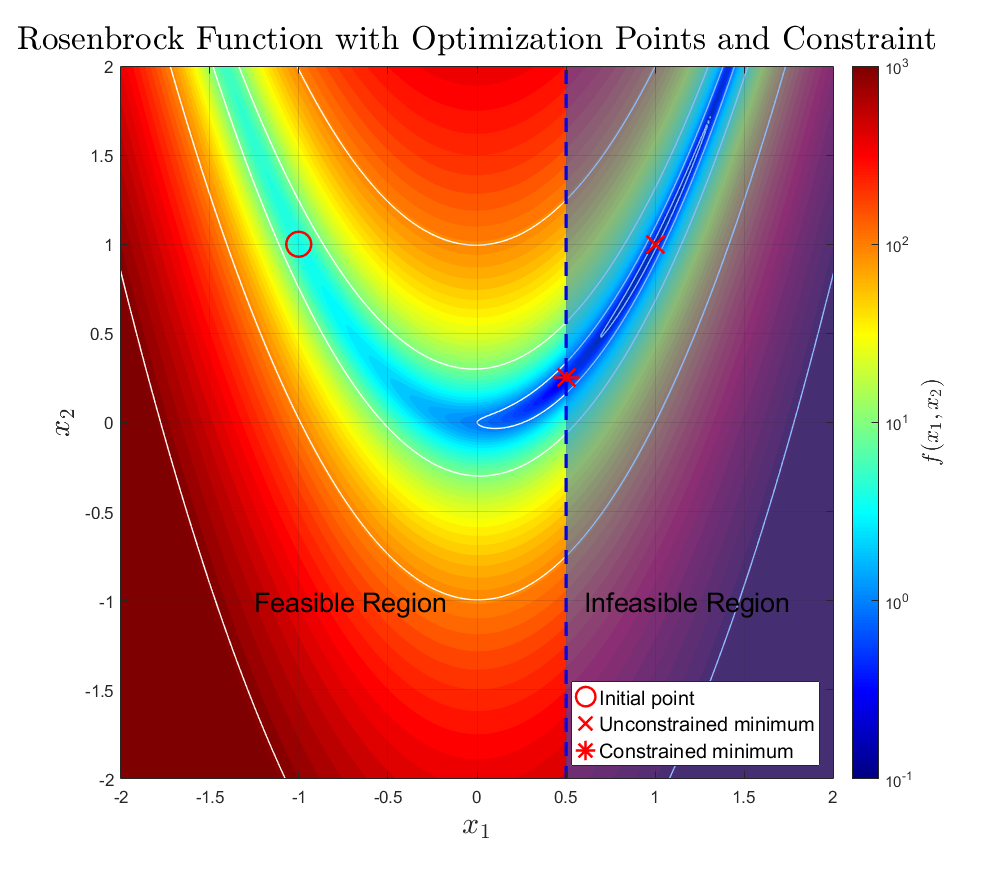
Os conteúdos gerados por IA poderão estar incorretos. 

Figure 1: Contour Plot of the Rosenbrock Function.

Figure 2: Rosenbrock Function with Optimization Points and Constraint.

Figure 1 presents a logarithmic heatmap of the Rosenbrock function overlaid with white contour lines. This plot displays only the function shape without including any optimisation points or annotations. In contrast, Figure 2 shows the initial point (o), the unconstrained solution (x) and the constrained minimum (\*), along with the shaded infeasible region defined by and its corresponding dashed boundary.

Uma imagem com diagrama, file, Gráfico, Saturação de cores

Os conteúdos gerados por IA poderão estar incorretos.

Figure 3: 3D Surface of the Rosenbrock Function.

Finally, *Figure 3* shows a 3D surface plot of the Rosenbrock function, highlighting the curved valley that leads to the global minimum.

## **P2 – Basic on Receding Horizon Control**

In this section, we analyse how optimal state feedback gains evolve under infinite and finite-horizon formulations. For an unstable and stable open-loop plant:

(11)

(12)

The goal is to understand how the control horizon and input penalisation influence system behaviour and stability.

We can represent both systems in standard state-space form:

The optimal control law is linear state feedback:

Where for the infinite-horizon Linear Quadratic problem, and for the finite-horizon Receding Horizon control, which will be analysed in the following sections.

## **P2.1 – LQ State Feedback Gain**

We begin by computing the infinite-horizon Linear Quadratic gain ​, obtained via MATLAB’s *dlqr* function, for the unstable plant. The associated cost function is:

(13)

And considering , since the signals are scalar,

For the LQ gain obtained was . As expected, smaller values of led to more aggressive control strategies, resulting in larger gains , while higher values of produced more conservative gains. This trade-off illustrates the balance between tracking performance and energy expenditure in optimal control design. These infinite-horizon gains serve as reference values when evaluating the performance of the finite-horizon controller.

## **P2.2 – Optimal RH Gain**

We then evaluated the optimal gains for predictive controllers with a finite horizon , using the associated cost function:

(14)

which has the solution

From a control perspective, this reflects how the system “looks ahead” further into the future as grows, making better decisions that resemble those of the infinite-horizon controller. As expected, the gains converge to the gain ​, with convergence occurring more rapidly for smaller values of . When , the gain remains constant for all , since control effort is not penalised.

Uma imagem com texto, file, Gráfico, diagrama

Os conteúdos gerados por IA poderão estar incorretos.

Figure 4: Optimal RH gains for the unstable system, for various values of R over H. With each R LQ gain as dashed lines.

As expected, lower values of result in larger gains, as the controller is allowed to be more aggressive. Conversely, when , the gain increases more gradually with , reflecting a more cautious control action due to the high penalization on input effort. Notably, for , the gain remains constant regardless of , since control energy is not part of the optimization objective.

## **P2.3 – Closed-Loop Eigenvalue**

To assess closed-loop stability, we analysed the eigenvalue . A system is considered stable if the magnitude of the eigenvalue satisfies . *Figure 5* show the evolution of for different input weights , as a function of the prediction horizon .

Uma imagem com file, Gráfico, texto, diagrama

Os conteúdos gerados por IA poderão estar incorretos.

Figure 5: Absolute eigenvalues for the unstable system, for various values of R over H.

*Figure 5*reveals that small horizons and high fail to stabilise the system (). Increasing gradually brings below one, with faster improvement for lower . These results confirm the theoretical prediction that longer horizons are essential to stabilise unstable dynamics, and that smaller penalties on the control action (lower ) improve responsiveness and stability.

## **P2.4 – Different Horizon values**

Enlarging the prediction horizon in receding horizon control brings clear advantages. As shown in *Figure 4*, the RH gain increases with and asymptotically approaches the infinite-horizon gain , with faster convergence for lower values of . For , the gain remains fixed at , since control effort is not penalised. When , longer horizons allow better anticipation of future states, improving performance and aligning the RH controller with the LQ optimum.

This effect is reflected in *Figure 5*, where the eigenvalue magnitude decreases as increases. Stability (i.e., ) is only achieved after a sufficiently long horizon, particularly when is high. Therefore, increasing is essential to stabilise unstable systems and to ensure the RH controller mimics the behaviour of the optimal LQ solution.

## **P2.5 – Open-loop stable plant**

We repeated the study for the open-loop stable system described by Eq. (12), where . As expected, the system remained stable for all tested values of the prediction horizon and control weight , with the closed-loop eigenvalues consistently inside the unit circle. The resulting optimal gains were significantly lower than in the unstable case and exhibited a much smoother dependence on .

Uma imagem com texto, file, captura de ecrã, Gráfico

Os conteúdos gerados por IA poderão estar incorretos.

Figure 6: Optimal RH gains for the stable system, for various values of R over H. With each R LQ gain as dashed lines.

*Figure 6* shows the evolution of the receding horizon gains as increases, for different values of . As in the previous case, we observe that for , the gain remains constant, since control effort is not penalised. For the gain increases with horizon length and converges towards the infinite-horizon solution ​, drawn as dashed lines. This convergence is faster for lower values of , indicating that in stable systems, even short horizons suffice to closely approximate the optimal gain.

Below *Figure 7* presents the absolute value of the closed-loop eigenvalue . In contrast with the unstable system, stability is guaranteed from the outset, regardless of or . The eigenvalue magnitude remains well below 1, and its evolution across is minimal, reinforcing the notion that the system's inherent stability reduces the need for long-term prediction.

For both stable and unstable plants, we explored a representative range .. The horizon was swept from to , which was sufficient to observe convergence and stability trends. These results highlight a key distinction: for unstable systems, large is critical to achieving stabilization and LQ-like performance; for stable plants, small already ensures both stability and near-optimal behaviour. Therefore, increasing the horizon is far more advantageous in the unstable case.

Uma imagem com texto, captura de ecrã, file, Gráfico

Os conteúdos gerados por IA poderão estar incorretos.

Figure 7: Absolute eigenvalues for the stable system, for various values of R over H.

## **P3 – Model Identification**

To implement Model Predictive Control (MPC), we require a dynamic model of the plant. While a physics-based model based on heat transfer principles is feasible for the TCLab system, we adopt a data-driven system identification approach instead. This method is more scalable and general, relying solely on input-output data without requiring prior knowledge of the plant’s internal structure.

We aim to identify a linear time-invariant model around a steady-state operating point , where:

(15)

We define deviations from equilibrium:

(16)

The system is approximated by the following incremental model:

(17)

where is a zero-mean Gaussian disturbance. This model will be used for prediction, observer design, and control synthesis in the upcoming stages.

In the first open-loop experiment (*Figure 8*), we drove the system to an equilibrium near 40 °C using a constant heater input of 25%. Then, small step changes (±5–10%) were applied with sufficient settling time, enabling the extraction of incremental responses . This validated the use of a SISO model, as Temperature 2 remained largely unaffected. Using MATLAB’s *ssest* function, we estimated the model matrices based on the collected data.

In the second open-loop experiment (*Figure 9*), we introduced faster and larger input changes, which better excited the system dynamics. Although these inputs did not allow for full steady-state settling, Temperature 1 remained within its operating region and displayed consistent behavior.

The identified model was then validated on this second dataset. Despite increased excitation and noise, the predicted output closely tracked the measured temperature, with a low mean squared error (MSE), confirming the model’s generalization capability.

A graph of a graph of a graph

AI-generated content may be incorrect. A close-up of a graph

AI-generated content may be incorrect.

Figure 8: Open-loop experiment used for model validation.

Figure 9: Open-loop experiment used for system identification. Heater input (top), temperature sensor readings (bottom).

After testing models with state dimensions from to , we observed a minimum mean squared error (MSE) of 1.2064 for , with higher errors for other orders: 1.8977 for , 1.6736 for , 1.6826 for , and 2.5949 for . Although increasing the state dimension initially improved the fit, higher-order models began to overfit the data, which is especially problematic given the significant measurement noise we encountered. We attempted to mitigate this by performing identification in different environments, but the noise persisted, indicating a likely hardware-related issue. Despite this limitation, the model with offered the best trade-off between complexity and performance, and we deemed it sufficiently accurate for the following stages of MPC and Kalman filter design.

## **P4 – MPC and Kalman Filter Design**

## **P4.2 – Effects of changing and on the controller**

After implementing the basic unconstrained MPC loop in the TCLab\_simulation.m script, we systematically studied the effects of varying the prediction horizon and the control penalty . Figure 10 *Figure 10* shows how the temperature output and control input evolve for horizons for a fixed . As expected, increasing improves tracking performance—responses become faster and less oscillatory, approaching the infinite-horizon solution. However, beyond further increases in produce only negligible performance gains, as the curves for and become virtually indistinguishable.

To evaluate computational cost, *Figure 11* reports the average solver time as a percentage of the sampling period ​ across increasing . While solver time remains below 0.01 % for , it grows sharply beyond that point, reaching 0.12% at . This confirms the expected quadratic complexity growth with horizon length and illustrates the trade-off between performance and real-time feasibility. Based on this we selected a prediction horizon of and intend to combine it with the previously mention .

Uma imagem com texto, diagrama, file, Gráfico

Os conteúdos gerados por IA podem estar incorretos. Uma imagem com texto, Gráfico, file, diagrama

Os conteúdos gerados por IA podem estar incorretos.

Figure 10: Closed‐loop temperature and heater output for H={2,3,5,10,20,100} at R=0.01.

Figure 11: Average solver execution time vs. horizon

We also investigated the influence of the control penalty , which balances control aggressiveness versus smoothness. *Figure 12* shows the response for fixed with , . Smaller values enable faster convergence to the setpoint but lead to more abrupt control inputs, potentially stressing actuators. On the other hand, larger values produce smoother commands at the cost of slower convergence. For , the controller responds very aggressively, with initial control values significantly exceeding the physical actuation limit of 100%. Despite this unrealistic input, we chose to proceed with this value at this stage because actuation constraints will be explicitly enforced in the next section. This allows us to isolate and understand the effect of on performance without the confounding influence of saturation.

Uma imagem com texto, file, diagrama, captura de ecrã

Os conteúdos gerados por IA podem estar incorretos.

Figure 12: Closed‐loop temperature and heater output for R={0.01, 0.02, 0.05, 0.1, 0.5, 1} at H=20

## **P4.3 – Control signal constraints**

The following figures show the closed-loop response of the unconstrained MPC controller using a linear incremental model. The controller regulates to using , constrained such that ∈ [0, 100]%.

The MPC operates on incremental variables, aiming to drive . The optimization vector , consisting of predicted control increments, is constrained via:

( 18 )

ensuring that the absolute input is respected at all time steps. These constraints are passed to the solver via the quadprog formulation.

In our implementation, we chose . With R = 0.01, the unconstrained MPC would generate u > 100%, but the active constraint enforces saturation at 100%. However, with the constraint active, we observe that saturates momentarily at 100%—as seen in the bottom plot of *Figure 13* —before decreasing and stabilising around . This behaviour confirms that the controller respects the imposed bounds while still achieving stable and efficient regulation.

A graph of a function

AI-generated content may be incorrect. A graph of a function

AI-generated content may be incorrect.

Figure 13: Absolute input/output for the constrained mpc\_solve for H=20 and R=0.01.

Figure 14: Incremental input/output for the constrained mpc\_solve for H=20 and R=0.01.

In *Figure 13*, the top plot shows that the output rapidly rises from ambient temperature (~23°C) and converges smoothly to the target equilibrium , as expected. The bottom plot confirms that the input , initially saturated near 100%, decreases quickly and stabilises around .

The corresponding incremental signals in *Figure 14* validate this behaviour. The top plot shows that , starting near –20°C, converges precisely to zero without overshoot, indicating accurate tracking. Simultaneously, (bottom plot) decays to zero after an initial step of approximately 77%, confirming that the controller successfully settles the system and ceases to act once equilibrium is reached.

These results demonstrate excellent closed-loop performance. The response is fast, smooth, and compliant with input constraints, validating the MPC formulation and confirming that the identified model accurately captures the system dynamics.

## **P4.4 – Reference Tracking with Feedforward**

In this experiment, the existing MPC formulation is extended to track a reference shift of approximately using a feedforward control scheme. To achieve this, we compute the steady-state deviations and necessary to maintain the output increment , by solving:

(19 )

A change of variables is then introduced:

(20)

The MPC regulator minimizes the tracking error in and control effort in , using the same prediction horizon and weight as before. The control constraints are updated to enforce ensuring feasibility.

*Figure 15* confirms that with the nominal model, the system closely tracks the reference increment: Δy smoothly approaches 5 °C and Δu converges to a steady value. However, when the model is perturbed by increasing the parameter ​ by 10%, the behaviour changes. As shown in *Figure 16*, this perturbation introduces a persistent offset in Δy of approximately 1.8°C, even though the control input Δu is adjusted. The error settles instead of decaying to zero, illustrating that the controller cannot reject constant disturbances or biases. The corresponding control error Δu reaches up to 12.5%, as the optimizer attempts to compensate but lacks model awareness of the disturbance.

Uma imagem com texto, diagrama, número, file

Os conteúdos gerados por IA podem estar incorretos. Uma imagem com texto, file, Gráfico, captura de ecrã

Os conteúdos gerados por IA podem estar incorretos.

Figure 15: Incremental output and control comparison with reference step at t = 250 s.

Figure 16: Tracking error between nominal and perturbed responses (Δy and Δu).

This limitation arises because the MPC operates on an internal model assumed to be exact. Any mismatch in steady-state behavior (such as from environmental drift or imperfect identification) leads to a misaligned equilibrium. The cost function penalizes control action and deviation from the desired trajectory but does not account for the structural modeling error itself.

In conclusion, while the feedforward tracking scheme is effective under nominal conditions, it is not robust to parameter shifts or constant disturbances. This motivates the need for disturbance estimation (e.g., via an augmented Kalman filter), which will be introduced in later sections to achieve zero steady-state error despite model mismatch.

## **P4.5 – Safety Constraint**

This test imposes a safety constraint on the output: , while commanding a reference - which violates this limit. Initially, we implemented this as a hard constraint in the MPC formulation. However, as expected, the optimization problem became infeasible during simulation. Specifically, quadprog returned an *exitflag = -2*, indicating that no admissible control sequence could satisfy both the system dynamics and the output constraint while tracking . This happens because, under the current system dynamics and thermal inertia, reaching would require inputs that inevitably push above at some steps, violating the constraint. To overcome this, the constraint is softened by introducing slack variables , allowing for controlled constraint violation with penalization.

The soft constraint formulation modifies the original constraint:

( 19 )

and updates the cost function to include a penalty term:

( 20 )

where ensures that constraint violations are discouraged unless strictly necessary.

In the dense formulation, the optimization variable becomes:

( 21 )

Uma imagem com texto, diagrama, file, captura de ecrã

Os conteúdos gerados por IA podem estar incorretos.

Figure 17: Absolute temperature and control response under soft output constraint.

Figure 17 shows the absolute input/output response of the system under this soft constraint formulation. The top plot confirms that the temperature rises to track the elevated reference but stabilizes just below the 55°C safety limit, rarely exceeding it. The blue dashed line marks the reference, while the magenta dashed line marks the safety threshold.

The bottom plot shows the control effort. We observe that the input saturates at 100% initially, followed by an oscillatory settling phase—expected behavior as the MPC balances setpoint tracking, constraint compliance, and actuator limits.

Despite the mathematical possibility of violating the constraint due to slack variables, the controller only does so when strictly necessary and keeps the violations minimal. This validates the robustness of the soft constraint approach, maintaining feasibility throughout the simulation with quadprog always returning exitflag = 1. This formulation guarantees operational safety without rendering the control task infeasible, making it an effective strategy for real-time MPC in constrained thermal systems.

## **P4.6 – Kalman Filter Design for Disturbance and State Estimation**

We treat the unknown steady‐state offset of the heater as a constant input disturbance by augmenting the deviation state with , giving:

( 22 )

From our identification experiments we know the measurement‐noise variance and we can infer that the covariance of is

( 23 )

We then introduce a tuning parameter , representing the (constant) disturbance variance, and form the augmented covariance

( 24 )

Applying the standard discrete‐time Kalman‐filter equations to the model with noise covariances and measurement variance , we obtain the steady‐state estimator gain . The filter is initialized with a small offset in the first output estimate to test convergence; thereafter each step consists of the usual predict–measure–correct updates, yielding simultaneous estimates of and .

*Figure 18* compares the true temperature (blue) with the filter’s output estimate (red dashed). With , the two curves merge after ≈ 60 s, showing that the filter rapidly reconstructs the true heater temperature.

Uma imagem com texto, captura de ecrã, file, Gráfico

Os conteúdos gerados por IA podem estar incorretos.

Figure 18: Measured vs Estimated Output

*Figure 19* shows the estimated disturbance It varies between 0% to 5%, matching roughly the intentionally injected plant offset.

Uma imagem com texto, Tipo de letra, file, Gráfico

Os conteúdos gerados por IA podem estar incorretos. Uma imagem com texto, file, Gráfico, diagrama

Os conteúdos gerados por IA podem estar incorretos.

Figure 19: Estimated Input Disturbance

Figure 20: Estimated Disturbance for different

*Figure 20* compares the effect of three values of ​ on the disturbance estimate . A small value results in a very smooth but slow estimate, taking over 400 s to converge. In contrast, a large value yields faster convergence but introduces significant overshoot and noise. The intermediate value ​ offers a good compromise, settling in about 200–250 s with limited oscillation. For this reason, it was selected as the default in our design.

## **P4.7 – Implementing MPC with State Estimation and Disturbance Compensation for Reference Tracking**

In this experiment, the MPC controller is closed with the Kalman-filtered state estimate and disturbance estimate The estimated disturbance is incorporated into the computation of the steady-state feedforward terms and , ensuring zero steady-state error under constant disturbances. This scheme enables precise setpoint tracking—even in the presence of modeling uncertainty—except where the safety constraint becomes active.

*Figure 21* shows four successive holds at 50°C, 40°C, 60°C and 45°C (black dotted). The solid blue curve is the true temperature and the red dashed is the one-step-ahead prediction from the MPC+Kalman. At 50°C and 40°C the two lie virtually on top of each other—and on the reference—once the initial transient settles (≈200 s). At the 60°C command the soft constraint (magenta dashed) becomes active, so the plant “plates out” just under 55°C. As soon as the reference drops back to 45°C the controller resumes zero-error tracking.

Uma imagem com texto, file, Gráfico, diagrama

Os conteúdos gerados por IA podem estar incorretos.

Figure 21: Measured vs Estimated Output with Constraints

Figure 22 plots the relative output error in percent. After each step it spikes briefly (while the Kalman filter catches up), then falls to nearly zero—showing that our feed-forward compensation entirely removes steady-state bias. The bottom panel shows converging in about a minute to the true, constant disturbance and staying there.

Uma imagem com texto, file, Gráfico, número

Os conteúdos gerados por IA podem estar incorretos. Uma imagem com texto, captura de ecrã, file, Gráfico

Os conteúdos gerados por IA podem estar incorretos.

Figure 22: Relative Output and Disturbance Errors

Figure 23: Constraint Relaxation Over Time

As shown in *Figure 23*, the slack variable η—used to soften the safety constraint—remains strictly zero throughout the simulation except during the 60°C setpoint hold. This confirms that the output constraint becomes active only when the reference exceeds the safety threshold. During this period, the optimizer allows controlled constraint violation, preventing infeasibility while keeping violations minimal.

This formulation effectively mimics the behavior of a PI controller with anti-windup. The estimated disturbance acts as an integral term, accumulating past prediction errors and offsetting them in the feedforward input . When the safety constraint is active, the slack variable prevents the controller from accumulating unbounded error—functionally equivalent to anti-windup clamping in classical control.

## **P5 – Application to the Real System**

The controller and observer developed in Part 4 were implemented in the real system using the *TCLab\_closedloop* script. This experiment replicated the test conditions of P4.7: the reference temperature was successively switched between 50 °C, 40 °C, 60 °C (above the safety threshold), and 45 °C. The results are shown in *Figure 24*, with the temperature response in the top subplot and the corresponding control input in the bottom subplot.

The system successfully tracks all references while respecting input constraints and enforcing the soft safety constraint of 55 °C. When a reference of 60 °C is commanded, the temperature initially rises and briefly exceeds the limit (peaking near 56.5 °C) but then returns and stabilizes just below the threshold. This behavior reflects the soft constraint formulation, where temporary violations are allowed but penalized, always ensuring optimizer feasibility (exitflag = 1).

A graph of a graph showing a number of different types of data

AI-generated content may be incorrect.

Figure 24: Absolute temperature and control for the real system.

This transient overshoot can be partly attributed to the fact that the identified model is quite noisy. During identification, significant measurement noise was observed, and despite multiple attempts under different conditions, it persisted — likely due to hardware limitations. As a result, the Kalman filter and MPC were designed using a model that has limited accuracy under fast transitions, especially when extrapolating beyond the range of excitation in the identification dataset.

Even so, the zero-offset tracking is preserved in steady-state, and the Kalman filter continues to estimate disturbances effectively. Overall, the real system behavior is close to simulation, with the main discrepancies arising from real-world imperfections not captured in the model.