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

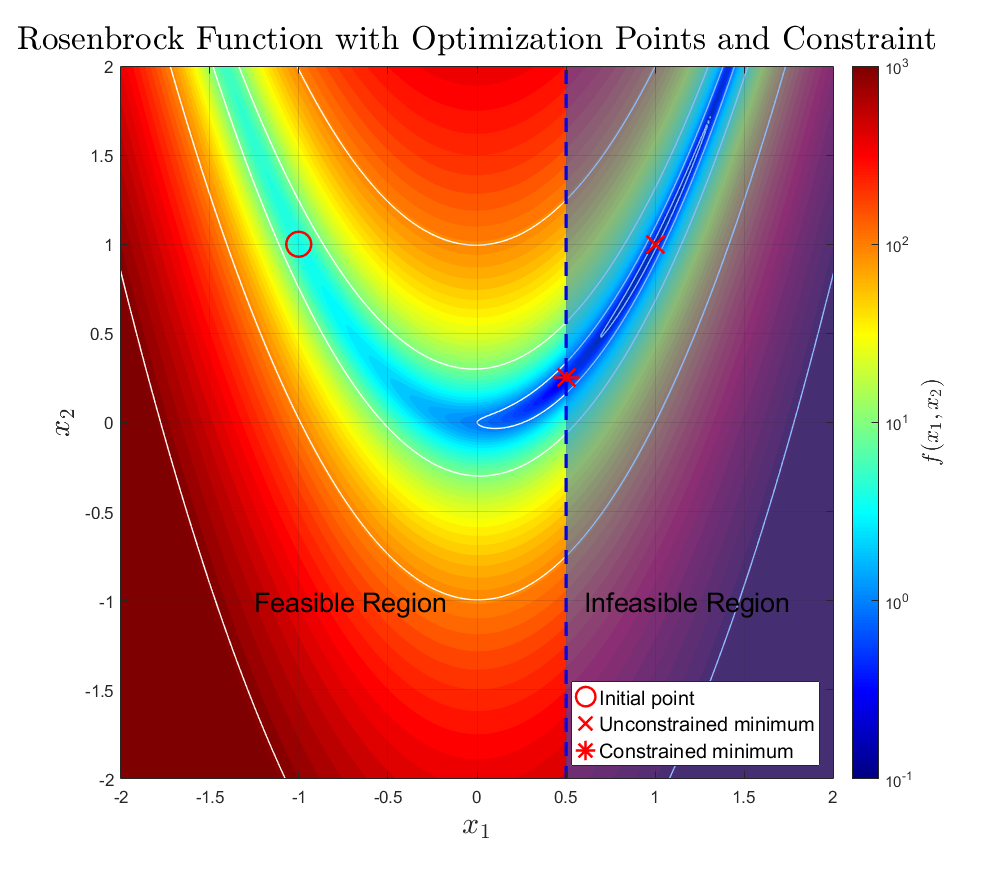
Os conteúdos gerados por IA poderão estar incorretos. 

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 feedback gains for the unstable system, for various values of R over H.

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* and *Figure 7* show the evolution of for the unstable and stable systems, respectively.

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.

In the unstable case, *Figure 5*reveals that small horizons and high fail to stabilise the system (). Increasing gradually brings below one, with faster improvement for lower , which confirms the theoretical prediction that longer horizons are critical to stabilise unstable dynamics.

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

Enlarging the prediction horizon in receding horizon control leads to several observable effects when compared to the infinite-horizon LQ gain .

As shown in *Figure 4*, which plots for the unstable system, the RH gain increases with and asymptotically approaches the corresponding gain. This convergence is faster for smaller values of . When , the gain is already equal to , and does not depend on . However, for , the effect of increasing is significant: it improves performance by better approximating the infinite-horizon optimal control. In *Figure 5*, we observe how the eigenvalues decrease with , crossing below 1 (i.e., entering the stable region) only after a sufficiently long horizon is reached — especially when is large. This highlights that in unstable systems, short horizons may fail to stabilize the system, and enlarging is crucial for ensuring stability.

## **P2.5 – Open-loop stable plant**

Finally, we repeated the study for a stable open-loop plant (Eq. (12)). As anticipated, the system remained stable across all and the gains were considerably lower than those required for the unstable case. The eigenvalues were consistently inside the unit circle and showed little sensitivity to horizon variation, confirming the system's intrinsic stability.

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

Os conteúdos gerados por IA poderão estar incorretos.

Figure 6: Optimal feedback gains for the stable system, for various values of R over H.

*Figure 6* shows finite-horizon gains for a stable system with , across the selected values of . For , the gain remains constant since the control effort is not penalised. For , the gain increases with and converges towards the gain (dashed lines). The convergence is faster for smaller , indicating that a short horizon is often sufficient to achieve near-optimal behaviour in stable systems.

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.

*Figure 7* demonstrates that for the stable case (), all eigenvalues remain well within the stability boundary, regardless of – This indicates that short horizons are sufficient.

In this exercise, we fixed the state weighting matrix to to maintain focus on the effects of varying the input penalization and prediction horizon. We selected a representative range of , capturing behaviors from fully aggressive control () to highly conservative strategies (). These values allow for a clear comparison of control effort versus performance trade-offs. The horizon was varied from 1 to 15, which proved sufficient to observe the convergence of the receding horizon gain to the infinite-horizon optimal gain , as well as the stabilization effects on the closed-loop eigenvalue. This parameter range successfully revealed the contrasting impact of horizon length on stable versus unstable systems, demonstrating that large horizons are critical for stabilizing unstable plants, while shorter horizons suffice in the stable case. These choices ensure the results are both illustrative and representative of practical control scenarios.

## **P3 – Model Identification**

To implement Model Predictive Control (MPC), we require a dynamic model of the plant. While first-principles modelling based on heat transfer laws is possible for the TCLab system, we will use data-driven linear system identification, which is more scalable and general. This approach assumes no prior knowledge of the internal dynamics and relies only on input-output data.

We aim to obtain a discrete-time, linear, time-invariant model valid near a steady-state operating point , where:

(15)

We define deviations from equilibrium:

(16)

and assume the system can be approximated by the incremental model:

(17)

where is a Gaussian disturbance. This model is used for prediction, observer design, and controller synthesis.

In the first experiment (

*Figure* 8), the heater input was set to 25% to drive the system to an equilibrium around 40 °C. Small step variations of ±5–10% were then applied, allowing sufficient settling time. This structure captures the incremental response and supports the use of a SISO model, as Temperature 2 remained largely unaffected. The resulting dataset was used to estimate the model using MATLAB’s *ssest* function.

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).

For validation, we performed a second open-loop experiment using faster and larger amplitude input changes (see *Figure 9*). Here, the system was excited with larger and faster input changes without full settling. Despite this, Temperature 1 exhibited smooth and consistent dynamics, remaining within the operating range. The identified model was simulated under this new input, and the predicted output matched the measured temperature with low mean squared error, confirming the model’s ability to generalise.

We selected a state dimension , which minimised the MSE and provided a good fit on both datasets without overfitting. However, higher-order models would lead to overfitting due the high amount of noise present in our measurements. This model will be used in the following stages for MPC and Kalman filter implementation.

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

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

After closing the loop with our basic unconstrained MPC controller in the TCLab\_simulation.m script, we carried out a systematic tuning of the prediction horizon and the control‐effort weight . Figure 10 illustrates the closed‐loop temperature response and heater output for horizons . As increases, the controller drives the temperature to the setpoint more quickly, closely approaching the ideal infinite‐horizon behaviour; however, beyond further increases in produce only negligible performance gains, as the curves for and become virtually indistinguishable.

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

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

At the same time, *Figure 11* shows that the average solver execution time (expressed as a percentage of the sample period ​) grows roughly quadratically with : it remains below 0.01 % of ​​ for , but climbs rapidly thereafter, reaching about 0.12% at . This demonstrates the classic trade‐off in MPC design—larger horizons improve setpoint tracking but incur higher computational cost.

We also examined the effect of the control‐effort weight , which penalizes aggressive changes in the heater command. Smaller values of yield faster temperature regulation at the expense of larger, potentially noisy control inputs, while larger produces smoother, more conservative actuation. By testing across several orders of magnitude, we found that offers a good compromise: the temperature reaches the desired setpoint in under 100 s, although the controller occasionally applies aggressive inputs. No constraints have been imposed at this stage; we will address them in a later phase. Combining these insights, we selected a prediction horizon of and , which together deliver near–infinite‐horizon controller performance while ensuring solver runtimes remain sufficiently low for real‐time implementation on the TCLab platform.

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

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 goal is to regulate the output temperature to the equilibrium value ​, using the incremental formulation and , where is the steady-state input required to maintain .

The MPC operates on incremental variables by applying the transformation (from eq.16), 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 . Without any constraints on the control input, the MPC generates an initial that exceeds 100%, which is not physically feasible. 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

Figure 14: Incremental input/output

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 test, the MPC is extended to track a reference with , using a feedforward formulation. To achieve this, a change of variables is introduced:

( 19 )

Where and are the steady-state solutions to:

( 20 )

The MPC optimisation problem then minimizes the tracking error in ​ and control effort in , subject to constraints updated accordingly:

( 21 )

Initially, the system behaves as expected: the output tracks the reference increment smoothly, with and the input converging to its new s. This is observed in *Figure 15*, where the output reaches the new setpoint after the step at , and in *Figure 16*, where approaches the reference step of 5°C.

A graph of a normal and a normal and a normal and a normal and a normal and a normal and a normal and a normal and a normal and a normal and a normal and a normal and

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

AI-generated content may be incorrect.

Figure 15: Absolute input/output

Figure 16: Incremental input/output

However, when the model is perturbed by increasing the parameter by 10%, the controller fails to maintain accurate tracking. In *Figure 15*, we see that the output stabilises below the intended reference, while in *Figure 16*, converges to a value slightly lower than 5°C. Meanwhile, the control input adjusts but eventually settles, indicating that the controller has no mechanism to correct for the disturbance-induced offset.

This behaviour clearly reveals a limitation of feedforward MPC when used without disturbance estimation: any persistent model mismatch or unmeasured disturbance results in a steady-state error. Because the cost function penalises deviation from reference and control effort, but does not account for modelling bias, the optimiser finds a solution that stabilises the system, yet cannot force .

In conclusion, although the feedforward tracking scheme provides accurate results when the model is correct, it lacks robustness to disturbances or parameter variations. This motivates the need for augmenting the system with a **Kalman filter** or equivalent observer to estimate and compensate for constant disturbances—ensuring zero steady-state error even in the presence of model uncertainty. This approach is addressed in later sections.

## **P4.5 – Safety Constraint**

This test imposes a safety constraint on the output: , while commanding a reference - which violates this limit. Imposing this as a hard constraint can cause infeasibility in the optimizer (quadprog). 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:

( 22 )

and updates the cost function to include a penalty term:

( 23 )

where ensures that constraint violations are discouraged unless strictly necessary.

In the **dense formulation**, the optimization variable becomes:

( 24 )

A graph of a function

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

AI-generated content may be incorrect.

Figure 17: Absolute input/output response.

Figure 18: Incremental input/output response.

The controller's behaviour is illustrated in Figure 17 andFigure 18(with output constraint and soft constraint relaxation). In Figure 17 (absolute values), the top plot shows the system attempting to track the elevated reference. However, the output stabilizes just below the 55℃ limit, honouring the constraint. The control input , shown in the bottom plot, exhibits saturation and then oscillatory behavior as the controller optimally adjusts to the imposed restriction.

The incremental plots in *Figure 18* confirm this response. The output increment peaks near +10℃ (i.e., targeting 53℃ to 54℃ over the base temperature), while displays increasingly aggressive fluctuations. These variations suggest that the controller uses the slack variables to achieve the best possible tracking performance within feasible bounds.

Despite constraint violation being mathematically possible via slack variables, the controller **prioritizes safety** and only marginally breaches the limit when strictly necessary. The optimizer (quadprog) remained feasible throughout the simulation, demonstrating that the soft constraint formulation is robust and reliable even under challenging references.

## **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:

( 25 )

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

( 26 )

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

( 27 )

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 19* compares the true temperature (blue) with the filter’s output estimate (red dashed). With , he 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 19: Measured vs Estimated Output

*Figure 20* 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.

Figure 20: Estimated Input Disturbance

The choice of ​ balances convergence speed against estimation stability. Larger ​​ increases the Kalman gain, yielding faster transient response but larger overshoot and oscillations in ; smaller makes the estimate smoother but slower to reach its true value. Empirically, setting ​ achieves a settling time of approximately 600 s without noticeable oscillation, and is therefore adopted for our controller.

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

When we close the loop with the Kalman-filtered state estimate and feed the MPC its own disturbance estimate (used to compute the steady-state offsets ), the result is perfect set-point tracking except where the safety limit intervenes.

*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.

Figure 22: Relative Ouputand Disturbance Errors

Because is effectively the integral of past prediction errors, injecting it into is mathematically equivalent to adding an integral term (as in a PI controller). That is, the MPC “learns” the persistent offset and cancels it—exactly what a PI’s I-action does. The slack variable in the QP only becomes nonzero during the 60°C interval, acting like an anti-windup: it prevents the internal disturbance estimate (and thus the controller output) from growing unbounded when the hard temperature limit is reached.

In short, combining MPC with Kalman-based disturbance estimation yields perfect reference tracking, disturbance rejection and constraint handling in one unified framework—mirroring the behaviour of a classic PI with anti-windup but with explicit prediction and constraint management.

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

...