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.Uma imagem com texto, Saturação de cores, captura de ecrã, arco-íris

Os conteúdos gerados por IA poderão estar incorretos.

Figure 1: Optimisation Points and Constraint. Figure 2: Heatmap and Contours.

*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.

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

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

Os conteúdos gerados por IA poderão estar incorretos.

Figure 3: 3D Surface Plot.

## **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.

## **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)

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)

This formulation allows us to observe how the finite-horizon controller approaches the infinite-horizon solution asincreases. 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.

*Figure 4* shows how, for an unstable system (), the gains increase with horizon length and approach the dashed reference values . This demonstrates the horizon’s role in improving control aggressiveness and reducing tracking error.

## **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, plotted against horizon 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 receding horizon gains for the stable system, for various values of R, over different values of 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, plotted against horizon 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.

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

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.

A close-up of a graph

AI-generated content may be incorrect.

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

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**

## **P4.1 – Problem Formulation**

...

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

...

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

...

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

...

## **P4.5 – Safety Constraint**

...

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

...

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

...

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

...