## **P4.2 – Unconstrained MPC: Horizon & Weight Tuning**

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 and with reduced oscillation, closely approaching the ideal infinite‐horizon behavior; however, beyond further increases in produce only negligible performance gains, as the curves for and become virtually indistinguishable.

At the same time, Figure 12 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 0.03 % at . This demonstrates the classic trade‐off in MPC design—larger horizons improve setpoint tracking but incur higher computational cost.Uma imagem com texto, diagrama, file, captura de ecrã

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

Os conteúdos gerados por IA podem estar incorretos.

Figura 212: Closed‐loop temperature and heater output for R={0.01, 0.02, 0.05, 0.1, 0.5, 1} at H=20

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

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. Testing across several orders of magnitude, we found that offers a good balance: the temperature reaches the desired setpoint in under 300 s without generating excessive control effort. Combining these insights, we chose 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, Gráfico, file, diagrama

Os conteúdos gerados por IA podem estar incorretos.

Figure 12: Average solver execution time vs. horizon

## **P4.6 – Augmented-State Kalman Filter Design & Open-Loop Test**

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

From our identification experiments we know the measurement‐noise variance

so that the process‐noise covariance on the original deviation state is

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

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 X 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 X 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.

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 60 s without noticeable oscillation, and is therefore adopted for our controller.

Plot showing the model response to different values of

## **P4.7 – Estimator-Based MPC: Closed-Loop Disturbance Compensation**

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 X 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 X** 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.

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.