Homework 1 - IFT6162

Alexandre Fournier, Simon Pino-Buisson & Justin Veilleux

October 2025

1 Supermarket Refrigeration

1.1 Results

These results are for a window size of 180 (classic size).

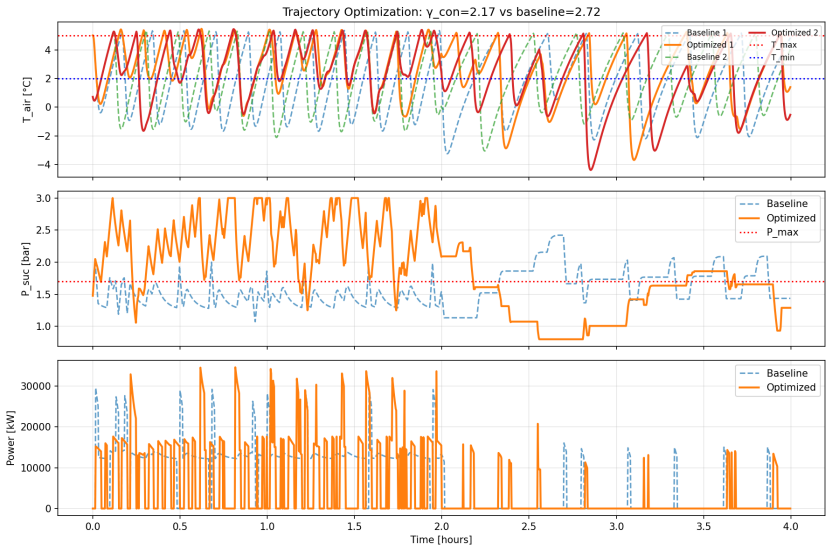

Figure 1: 4h simulation comparison between the implemented MPC and the baseline PI controller - window size=180

	γ_{con}	γ_{pow}	γ_{switch}
Baseline	2.724	6.802kW	0.003799
MPC	2.167	5.046kW	0.010449
Improvement	+20.4%	+25.8%	-175.0%

These results are for a window size of 90 (variant).

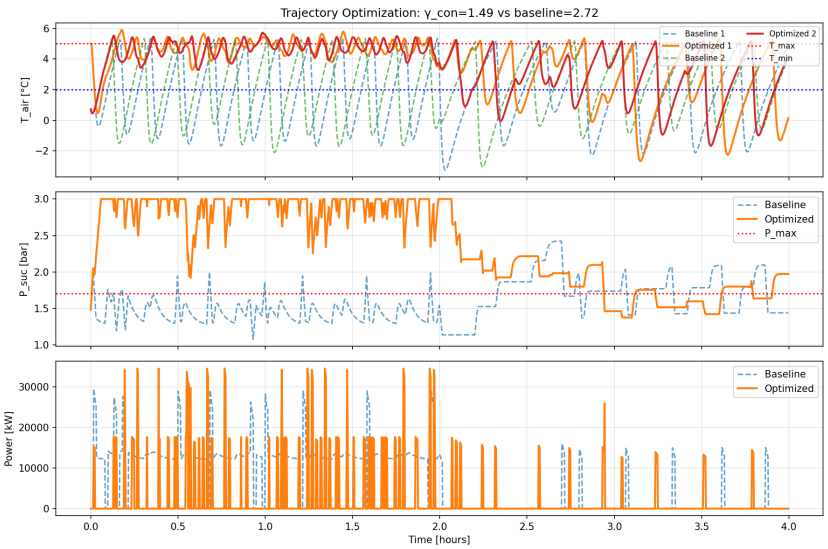

Figure 2: 4h simulation comparison between the implemented MPC and the baseline PI controller - window size=90

	γ_{con}	γ_{pow}	γ_{switch}
Baseline	2.724	6.802kW	0.003799
MPC	1.487	2.550kW	0.011602
Improvement	+45.4%	+62.5%	-205.4%

1.2 Discussion

1.2.1 Comparison

As seen in the results section, the MPC controller achieves results comparable to the PI controller baseline. Our model has a much higher switching cost than the baseline which could damage the compressors over long term usage. Using a higher switch cost weight leads to similar results. However, the implemented MPC improved the baseline γ_{con} and γ_{pow}. Our implementation manages to stay within the temperature bounds during the day, but starts oscillating and going under the lower bound at night. We can see that the MPC controller is not able to respect the bounds for P_{suc}, which can also damage the compressors.

While our model has worse performances for γ_{switch}, it slightly improves γ_{con} and γ_{pow}.

1.2.2 Trade-Offs - Energy vs Switching vs Constraint Tightness

As seen in the figures, optimized trajectory achieves a different trade-off compared to the baseline policy. The PID controller achieves low pressure constraint violations, while the optimized trajectory stays closer to the valid temperature range. Although in theory, the constrained optimization procedure should produce strictly feasible trajectories, it expends its entire iteration budget with both the default number of iterations (50) and a very large number of iterations (one billion) without producing a trajectory that respects the constraints. This suggests that either the algorithm itself is ill-suited for the task, or producing feasable trajectories would require decreasing the integration step size and increasing the window such that the runtime stops being practical for real-time control.

1.2.3 How Multiple Shooting Helps Constraining Path and Runtime per Window

In theory, multiple shooting should help enforce constraints along the path at each window segment intersection. However, in practice, we observe that it remains challenging to find feasible solutions that satisfy all constraints. The experiments also show that reducing the window size leads to shorter execution times, as multiple shooting yields smaller nonlinear programs (NLPs), thereby decreasing the computational load of the SLSQP solver for each window.