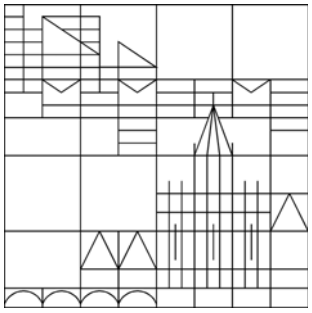


Universität Konstanz

Performance estimates for economic model predictive control and their application in proper orthogonal decomposition-based implementations

Lars Grüne
Luca Mechelli
Simon Pirkelmann
Stefan Volkwein

Konstanzer Schriften in Mathematik

Nr. 383, Mai 2019

ISSN 1430-3558

Konstanzer Online-Publikations-System (KOPS)
URL: http://nbn-resolving.de/urn:nbn:de:bsz:352-2-ywxqb2eru5d19

PERFORMANCE ESTIMATES FOR ECONOMIC MODEL PREDICTIVE CONTROL AND THEIR APPLICATION IN PROPER ORTHOGONAL DECOMPOSITION-BASED IMPLEMENTATIONS

Lars Grüne*, Luca Mechelli**, Simon Pirkelmann* and Stefan Volkwein**

*University of Bayreuth, Universitätsstraße 30,
95447 Bayreuth, Germany

**University of Konstanz, Universitätsstraße 10
78464 Konstanz, Germany

Abstract. In this paper performance indices for economic model predictive controllers (MPC) are considered. Since existing relative performance measures, designed for stabilizing controllers, fail in the economic setting, we propose alternative absolute quantities. We show that these can be applied to assess the performance of the closed loop trajectories on-line while the controller is running. The advantages of our approach are demonstrated by simulations involving a convection-diffusion-system. The method is also combined with proper orthogonal decomposition, thus demonstrating the possibility for both efficient and performant MPC for systems governed by partial differential equations.

Introduction. Model predictive control (MPC) is a well established method for the control of dynamical systems. In classical MPC, the task is to follow an a priori given trajectory which is usually specified by the user. The main focus in that setting is tracking the trajectory. This is achieved by penalizing deviations from a reference trajectory using a stage cost function which is positive definite w.r.t. the reference trajectory.

In order to verify that the tracking works as expected the MPC closed loop cost can be considered because for tracking type cost functionals it corresponds to the tracking error. This has been done in [5] where an a posteriori error estimator for the MPC performance was derived which relates the MPC performance to the cost of an optimal trajectory on an infinite horizon.

In economic MPC, one is interested in a controller that operates the system at a trajectory where the cost is minimal. In contrast to classical MPC, in economic MPC the optimal trajectory is not simply obtained as the reference trajectory in a tracking type cost functional, but it is rather determined by the interplay between the dynamics and the cost functional [1, 3]. Often this trajectory will be an equilibrium, although sometimes also more complicated operating behaviors may emerge [7, 8, 14, 17]. In this case the stage cost may be more general and in

2010 *Mathematics Subject Classification.* Primary: 93B40, 93B52.
Key words and phrases. Economic model predictive control, proper orthogonal decomposition.
The authors are supported by DFG grants GR 1569/16-1 and VO 1658/4-1.

particular will not be positive definite w.r.t. the optimal equilibrium. Economic MPC offers the advantage that the controller can detect an optimal equilibrium on its own, that is not explicitly prescribed by the user. This setting also applies to tracking type functionals for which perfect tracking is not possible, either because the system is not controllable to the desired reference or because exact tracking causes persistent nonzero control costs. In this situation, the error estimator from [5] does not work. Still, the user may want to have a way to judge the performance of the controller, for example in order to tune its parameters or to gain some insight into what the controller is doing.

The aim of this paper is to derive performance estimates for economic model predictive controllers. The estimates compare the cost of the solution produced by the MPC to the cost of a partially optimal trajectory. This gives a criterion that shows if the controller is performing nominally. This can also be used to decide if adjustments to the parameters used in the MPC implementations have to be made, e.g. by changing the length of the MPC horizon or discretization parameters. We will particularly focus on implementations based on proper orthogonal decomposition (POD) [9, 10], for which we will illustrate the behavior of our performance estimate numerically. The main reason to utilize model order reduction together with economic MPC relies on the fact that we can speed up the solution of the open-loop optimal control problems, getting a good approximation of the full order performance estimate, provided that a sufficiently large number of POD basis functions is chosen [12, 13]. Our results indicate that the performance estimate can be used as a reliable tool to assess the quality of the MPC closed loop solution during runtime. For POD based MPC, our performance estimate in particular provides an alternative to error estimates tailored for the POD method [15].

This paper is structured as follows. In the first section we introduce the problem statement and the notion of an optimal equilibrium. After that we present the economic model predictive control algorithm which will be used to compute an approximate solution to our problem. We then propose two kinds of performance indices which are used for rating the quality of the numerical solution. It is also shown that the second performance index remains valid even if it is computed using suboptimal solutions. We finish the paper by giving a short introduction to proper orthogonal decomposition, which is then used for the numerical verification of our findings in a concluding example.

1. Problem statement.

We consider the discrete-time dynamical system

$$x(k+1) = f(x(k), u(k)), \quad x(0) = x, \tag{1}$$

with $f : X \times U \to X$ and state space X and control space U are normed spaces. Here $k \in \mathbb{N}_0$ denotes time, $x(k) \in X$ the state of the system at time k and $u(k) \in U$ the control.

For $N \in \mathbb{N}$ let U^N denote the set of finite control sequences $u(0), \dots, u(N-1)$. A trajectory of the system starting from initial state $x \in X$ controlled by $u \in U^N$ is denoted by $x_u(\cdot; x)$. To incorporate state and control constraints in the problem statement the sets of admissible states will be denoted by $\mathbb{X} \subseteq X$ and the sets of admissible control values for $x \in \mathbb{X}$ by $\mathbb{U}(x) \subseteq U$. By $\mathbb{U}^N(x)$ we denote the sets of admissible control sequences for initial state $x \in \mathbb{X}$ up to time N, i.e., control sequences $u \in U^N$ satisfying

$$u(j) \in \mathbb{U}(x_u(j; x)) \text{ and } x_u(j+1; x) \in \mathbb{X}$$

for all $j = 0, \dots, N-1$. In the following we will assume that the set $\mathbb{U}^N(x)$ is nonempty for all $N \in \mathbb{N}$. The set $\mathbb{U}^\infty(x)$ denotes the natural extension of this definition to the infinite horizon.

Let $\ell : X \times U \to \mathbb{R}_0^+$ be the stage cost function. We consider the cost functional

$$J_\infty(x,u) = \sum_{j=0}^{\infty} \ell(x_u(j;x), u(j)) \tag{2}$$

Our aim is to compute a control sequence $u \in \mathbb{U}^\infty(x)$ that minimizes $J_\infty(x,u)$.

In general this is not a well-defined problem unless we impose additional assumptions on the stage cost. In order to arrive at a consistent problem statement, we assume that there exists an optimal equilibrium as defined in the following:

Definition 1.1 (Optimal equilibrium)**.** An equilibrium $(x^e, u^e) \in \mathbb{X} \times \mathbb{U}$ is called *optimal equilibrium* if it satisfies

$$\ell(x^e, u^e) \leq \ell(x,u) \text{ for all } (x,u) \in \mathbb{X} \times \mathbb{U} \text{ with } f(x,u) = x. \tag{3}$$

In model predictive control one often demands that the stage cost is positive definite w.r.t. to the optimal equilibrium, i.e. it satisfies $\ell(x^e, u^e) = 0$ and $\ell(x,u) > 0$ for $(x,u) \neq (x^e, u^e)$. We will not make this assumption. Instead, we introduce a shifted stage cost

$$\hat{\ell}(x,u) := \ell(x,u) - \ell(x^e, u^e) \tag{4}$$

Note that, while the shifted stage cost satisfies $\hat{\ell}(x^e, u^e) = 0$, it is not sign definite. Using this shifted stage cost function we define the corresponding shifted cost functional

$$\hat{J}_\infty(x,u) = \sum_{j=0}^{\infty} \hat{\ell}(x_u(j,x), u(j)) \tag{5}$$

We restate our problem in terms of this shifted cost functional:

$$\underset{u}{\text{minimize}}\ \hat{J}_\infty(x,u) \tag{6}$$

We also consider the shifted optimal value function

$$\hat{V}_\infty(x) := \min_u \hat{J}_\infty(x,u). \tag{7}$$

Under appropriate conditions (see [6, Lemma 8.17 and Section 8.5]) one can guarantee that $\hat{V}_\infty$ is bounded from below, thus making the minimization problem well-defined. In the following we will denote a minimizing optimal control sequence of (7) by u_∞^*.

An approximation of the infinite horizon optimal control sequence u_∞^* can be computed by using MPC (model predictive control) as described in the following section.

2. Economic model predictive control.

We solve problem (6) approximately with MPC. The idea is as follows. Instead of solving the problem on the infinite horizon we fix $N \in \mathbb{N}$ and consider a finite horizon cost functional.

Definition 2.1 (MPC cost functional)**.** The MPC cost functional is defined as

$$J_N(x,u) = \sum_{j=0}^{N-1} \ell(x_u(j;x), u(j)). \tag{8}$$

In each step of the MPC algorithm the following optimization problem is solved.

Definition 2.2 (MPC optimal control problem and optimal value function)**.** Consider the problem

$$\underset{u\in\mathbb{U}^N(x)}{\text{minimize}} \quad J_N(x,u). \tag{9}$$

The corresponding optimal value function is defined by

$$V_N(x) := \inf_{u\in\mathbb{U}^N(x)} J_N(x,u).$$

We will assume that a minimizer of this problem always exists and denote it by u_N^*, or $u_{N,x}^*$ if we want to make the dependence on the initial state x explicit. For the optimal control sequence it holds $V_N(x) = J_N(x, u_{N,x}^*)$.
Note that the MPC problem is formulated again in terms of the original stage cost ℓ instead of its shifted version. This is convenient from an implementation point of view since using $\hat{\ell}$ instead would require knowledge of the cost of the optimal equilibrium which might be unavailable. The optimal control sequences and trajectories of the finite horizon problem using ℓ resp. $\hat{\ell}$ does not differ since we only shift the cost functional by a constant value $(-N\ell(x^e, u^e))$.

The model predictive control algorithm given in Algorithm 1 produces an approximate solution to problem (6).

Algorithm 1 MPC algorithm

for each time instant $k = k_0, k_0+1, \dots$ **do**
1. Measure the current state $x = x(k)$ of the system.
2. Solve the optimal control problem (9) in order to obtain the optimal control sequence $u_{N,x}^*$.
3. Apply the first element of $u_{N,x}^*$ as a control to the system during the next sampling period, i.e. use the feedback law $\mu_N(x) := u_{N,x}^*(0)$.

end for

The trajectory of the system generated by Algorithm 1 will be called *closed-loop trajectory*. We will denote it by $x_{\mu_N}(\cdot, x)$ for initial value $x = x(k_0) \in \mathbb{X}$. The cost of the closed loop for K time steps is defined by

$$J_K^{cl}(x,\mu_N) = \sum_{j=0}^{K-1} \ell(x_{\mu_N}(j,x), \mu_N(x_{\mu_N}(j,x))).$$

Under appropriate assumptions it can be shown that the control trajectory generated by MPC approximates u_∞^*, i.e. the solution to problem (6) ([3, 4]). In fact, there are proofs of convergence both for the costs and for the trajectories themselves [6]. However, the results only show convergence as the horizon $N \to \infty$. In particular, they do not quantify the error between the MPC solution and the solution on the infinite horizon. From a practical point of view this would be desirable, e.g. in order to guarantee certain performance bounds or as a means to tune the MPC horizon until convergence is achieved up to a certain degree.

In the following section we will address this by considering two different types of performance indices for the MPC trajectories.

3. MPC performance indices.

3.1. Relative performance index.

We will first revisit a relative performance index initially proposed in [5]. This performance index is based on the relaxed dynamic programming principle. The relaxed dynamic programming principle (see [6, Theorem 4.11], [11]) states that if we can guarantee the existence of some $\alpha \in (0,1]$ such that

$$V_N(x(k)) \geq \alpha \ell(x(k), \mu_N(x(k))) + V_N(f(x(k), \mu_N(x(k)))) \tag{10}$$

holds along the MPC trajectory $x = x_{\mu_N}$, then the following performance bound for the closed loop cost on the infinite horizon can be established:

$$J_\infty^{cl}(x, \mu_N) \leq \frac{1}{\alpha} V_N(x). \tag{11}$$

In addition stability of the MPC closed loop trajectory can also be deduced.

In order to use this for evaluating the cost of the MPC closed loop we have to check if $\alpha \in (0,1]$ can be found such that (10) holds for each $k \in \mathbb{N}_0$. The closer α is to 1 the better the performance of the MPC controller will be.

In practice we compute the performance index by

$$\alpha(k) := \frac{V_N(x(k)) - V_N(f(x(k), \mu_N(x(k)))}{\ell(x(k), \mu_N(x(k)))} \tag{12}$$

after each MPC step for all k. This can be accomplished on-line, however, the performance index for time step k becomes available only after the open loop problem at time step $k+1$ has been solved. In that sense we obtain an *a posteriori* estimator.

This relative performance index was originally designed for stabilizing MPC controllers in mind, where the stage cost is positive definite w.r.t. the stabilized equilibrium. In our setting this is not the case, i.e. in general we have $\ell(x^e, u^e) \neq 0$. As a consequence, for any MPC trajectory that converges to the optimal equilibrium the performance index satisfies $\alpha(k) \to 0$ as $k \to \infty$ since the numerator in (12) approaches zero while the denominator approaches a positive number. This means the relative performance estimate does not give a meaningful value of the true performance.

A remedy could be to work with the shifted stage cost $\hat{\ell}$ instead, i.e. considering the modified relaxed dynamic programming inequality

$$\begin{aligned} \alpha \hat{\ell}(x(k), \mu_N(x(k))) &\leq \hat{V}_N(x(k)) - \hat{V}_N(f(x(k), \mu_N(x(k)))) \\ &= V_N(x(k)) - V_N(f(x(k), \mu_N(x(k)))). \end{aligned} \tag{13}$$

Then by summing up we can estimate

$$\begin{aligned} \alpha \underbrace{\sum_{j=0}^{K-1} \hat{\ell}(x(j), \mu_N(x(j)))}_{=: \hat{J}_K^{cl}(x, \mu_N)} &\leq \sum_{j=0}^{K-1} V_N(x(j)) - V_N(x(j+1)) \\ &= V_N(x(0)) - V_N(x(K)). \end{aligned} \tag{14}$$

Since ℓ is nonnegative we can further estimate

$$V_N(x(0)) - V_N(x(K)) \leq V_N(x(0)) \tag{15}$$

which implies that

$$\hat{J}_K^{cl}(x, \mu_N) \leq \frac{1}{\alpha} V_N(x(0)). \tag{16}$$

Alternatively, it also holds that

$$V_N(x(0)) \leq V_K(x(0)) \tag{17}$$

for $K \geq N$, yielding the estimate

$$\hat{J}_K^{cl}(x, \mu_N) \leq \frac{1}{\alpha} V_K(x(0)). \tag{18}$$

This upper bound for the closed loop cost is useful only if $\ell(x^e, u^e)$ is close to zero (and thus $\hat{J}_K^{cl}(x, \mu_N) \approx J_K^{cl}(x, \mu_N)$). In the general case the estimate will be too conservative.

3.2. **Absolute performance index.** Since the relative performance index is of limited use if the stage cost is not positive definite, we now want to derive an absolute performance index that overcomes these limitations.

Theorem 3.1 (Absolute performance index)**.** *Consider the dynamical system* (1) *with general stage cost* $\ell : X \times U \to \mathbb{R}$ *controlled by an MPC controller* μ_N*. For* P *in* $\{0, \dots, N-1\}$ *and* $K \geq N$ *define the quantities*

$$\varepsilon_N^1(k) := V_N(x(k)) - V_N(x(k+1)) - \hat{\ell}(x(k), \mu_N(x(k))), \tag{19a}$$

$$\varepsilon_{N,P}^2(K) := V_{N-P}(x_{u^*_{N,x}}(P, x)) - V_{N-P}(x_{u^*_{N,x(K)}}(P, x(K))), \tag{19b}$$

$$\varepsilon_{N,P}^3(K) := P\ell(x^e, u^e) - J_P(x(K), u^*_{N,x(K)}) \tag{19c}$$

and let

$$E_{N,P}(K) := \sum_{k=0}^{K-1} \varepsilon_N^1(k) - \varepsilon_{N,P}^3(K) - \varepsilon_{N,P}^2(K). \tag{20}$$

Then the equation

$$E_{N,P}(K) = J_P(x, u^*_{N,x}) + (K - P)\ell(x^e, u^e) - J_K^{cl}(x, \mu_N) \tag{21}$$

holds.

Proof. Summing up (19a) along the MPC closed loop trajectory yields

$$\begin{aligned} \sum_{k=0}^{K-1} \varepsilon_N^1(k) &= \sum_{k=0}^{K-1} [V_N(x(k)) - V_N(x(k+1)) + \ell(x^e, u^e) - \ell(x(k), \mu_N(x(k)))] \\ &= V_N(x) - V_N(x(K)) + K\ell(x^e, u^e) - \underbrace{\sum_{k=0}^{K-1} \ell(x(k), \mu_N(x(k)))}_{= J_K^{cl}(x, \mu_N)} \end{aligned} \tag{22}$$

or equivalently

$$V_N(x) - V_N(x(K)) = J_K^{cl}(x, \mu_N) - K\ell(x^e, u^e) + \sum_{k=0}^{K-1} \varepsilon_N^1(k). \tag{23}$$

By the dynamic programming principle for any $P \in \{0, \dots, N-1\}$ we can rewrite the terms on the left-hand side as

$$V_N(x) = J_P(x, u^*_{N,x}) + V_{N-P}(x_{u^*_{N,x}}(P, x)) \tag{24}$$

and

$$V_N(x(K)) = J_P(x(K), u^*_{N,x(K)}) + V_{N-P}(x_{u^*_{N,x(K)}}(P, x(K))). \tag{25}$$

Now consider

$$\begin{aligned} V_N(x) - V_N(x(K)) &= J_P(x, u^*_{N,x}) - J_P(x(K), u^*_{N,x(K)}) + V_{N-P}(x_{u^*_{N,x}}(P,x)) \\ &\quad - V_{N-P}(x_{u^*_{N,x(K)}}(P, x(K))) \\ &= J_P(x, u^*_{N,x}) - P\ell(x^e, u^e) \\ &\quad + \underbrace{V_{N-P}(x_{u^*_{N,x}}(P,x)) - V_{N-P}(x_{u^*_{N,x(K)}}(P, x(K)))}_{=\varepsilon^2_{N,P}(K)} \\ &\quad + \underbrace{P\ell(x^e, u^e) - J_P(x(K), u^*_{N,x(K)})}_{=\varepsilon^3_{N,P}(K)}. \end{aligned} \tag{26}$$

Combining (23) and (26) yields

$$J_P(x, u^*_{N,x}) - P\ell(x^e, u^e) + \varepsilon^3_{N,P}(K) + \varepsilon^2_{N,P}(K) = J^{cl}_K(x, \mu_N) - K\ell(x^e, u^e) + \sum_{k=0}^{K-1} \varepsilon^1_N(k)$$

and by reordering we obtain

$$\begin{aligned} J_P(x, u^*_{N,x}) + (K-P)\ell(x^e, u^e) - J^{cl}_K(x, \mu_N) &= \sum_{k=0}^{K-1} \varepsilon^1_N(k) - \varepsilon^3_{N,P}(K) - \varepsilon^2_{N,P}(K) \\ &= E_{N,P}(K). \end{aligned}$$

This concludes the proof. □

Theorem 3.1 states that the quantity $E_{N,P}(K)$ measures the difference between the MPC closed loop cost for K steps and the cost of a trajectory that consists for the first P steps of a finite horizon open loop and after that of the cost of the optimal equilibrium. The quantities ε^1_N, $\varepsilon^2_{N,P}$ and $\varepsilon^3_{N,P}$ that compose the error estimate can all be computed on-line (assuming that $\ell(x^e, u^e)$ is known).

Remark 1. It should be noted that $E_{N,P}(K)$ could also be determined by just computing the right-hand side of equation (21) directly. However, the error estimate $E_{N,P}(K)$ offers several advantages. For one, the information is available much sooner; after just a few steps it is possible to determine whether something is going wrong. In addition, the step-by-step data also provides more detailed information about the error. It could be that the individual error estimates are large, but partially cancel each other out and then deliver a small error after P steps. This would mean, however, that MPC only provides a good solution "by chance", which one would not recognize without the error estimator. ◊

3.3. **Interpretation of the absolute performance index.** In this section we give a more in-depth insight into what the individual terms of performance index from Theorem 3.1 tell us about the quality of the MPC closed loop solution.
For this we first revisit two key assumptions introduced in [6]. The first assumption we need is the turnpike property. It demands that open-loop solutions of problem (9) spend most of their time close to the optimal equilibrium (x^e, u^e).

Assumption 3.2 (Turnpike property)**.** There exists $\sigma \in \mathcal{L}$[1] such that for each optimal trajectory $x_{u^*_{N,x}}(k, x)$, $x \in \mathbb{X}$ and all $N, P \in \mathbb{N}$, $P \leq N$, there is a set

[1] $\mathcal{L} := \{\delta : \mathbb{R}^+_0 \to \mathbb{R}^+_0 \mid \delta \text{ is continuous and strictly decreasing with } \lim_{t\to\infty} \delta(t) = 0\}$.

$\mathcal{Q}(x,P,N) \subseteq \{0,\dots,N\}$ with at most P elements and

$$|x_{u^*_{N,x}}(k,x) - x^e| \leq \sigma(P) \tag{27}$$

for all $k \notin \mathcal{Q}(x,P,N)$.

The second crucial assumption is a continuity property of the optimal value functions V_N.

Assumption 3.3 (Continuity property of V_N)**.** Assume there exist functions $\gamma_V \in \mathcal{K}_\infty$ and $\omega \in \mathcal{L}$ such that for each $N \in \mathbb{N}$ and each $x \in \mathbb{X}$ the optimal value functions V_N in a neighborhood of x^e satisfy

$$|V_N(x) - V_N(x^e)| \leq \gamma_V(\|x - x^e\|) + \omega(N). \tag{28}$$

The basis for the first error term ε^1_N is the following lemma from [6].

Lemma 3.4 (cf. Lemma 8.26 in [6])**.** *Let Assumption 3.2 and Assumption 3.3 hold. Then the equation*

$$V_N(x) = V_{N-1}(x) + \ell(x^e,u^e) + R_2(x,N) \tag{29}$$

holds with $|R_2| \leq \nu_2(\|x-x^e\|) = \gamma_V(\sigma(\lfloor N/2 \rfloor)) + 2\omega(\lfloor N/2 \rfloor - 1)$. □

Consider the following relation for the MPC closed loop cost, which directly follows from the dynamic programming principle.

$$V_N(x(k)) = \ell(x(k),\mu_N(x(k))) + V_{N-1}(x(k+1)). \tag{30}$$

Applying Lemma 3.4 to this equation yields

$$V_N(x(k)) = \ell(x(k),\mu_N(x(k))) + V_N(x(k+1)) - \ell(x^e,u^e) - R_2(x(k),N). \tag{31}$$

By rearranging the previous equation to

$$-R_2(x(k),N) = V_N(x(k)) - V_N(x(k+1)) + \ell(x^e,u^e) - \ell(x(k),\mu_N(x(k))) = \varepsilon^1_N(k) \tag{32}$$

we immediately see that ε^1_N corresponds exactly to the error term from the Lemma 3.4. This means we can interpret ε^1_N as a measure for the improvement that an optimal trajectory with horizon length N offers compared to a shorter trajectory with length $N-1$ that is augmented with the cost of one step on the optimal equilibrium. The other error terms can be interpreted as follows. $\varepsilon^2_{N,P}(K)$ measures the difference between the final piece of optimal trajectories starting in x and $x(K)$, respectively, cf. Figure 1. If Assumption 3.2 is satisfied and we choose P such that the final pieces of the optimal trajectories start near the turnpike, this means the performance index $\varepsilon^2_{N,P}$ measures the difference of the so-called leaving arcs. This can be used as an indicator of how much the leaving cost has improved between the first and the most recent step of the MPC.

In contrast, the error term $\varepsilon^3_{N,P}(K)$ measures the difference between the initial piece of the optimal trajectory starting in $x(K)$ and the cost of the optimal equilibrium for P steps. This is motivated by an observation from Lemma 6.3 in [3], which states that an optimal trajectory originating near the optimal equilibrium will stay near the optimal equilibrium for some time because this is the cheapest option. A "good" MPC controller will drive the state $x(K)$ to a neighborhood of the optimal equilibrium. If this is not the case it can be detected by performance index $\varepsilon^3_{N,P}$ which will yield a larger error.

The choice of P is subject to further investigation. Figure 2 shows different choices of P for an MPC horizon N that has been chosen sufficiently large such that the

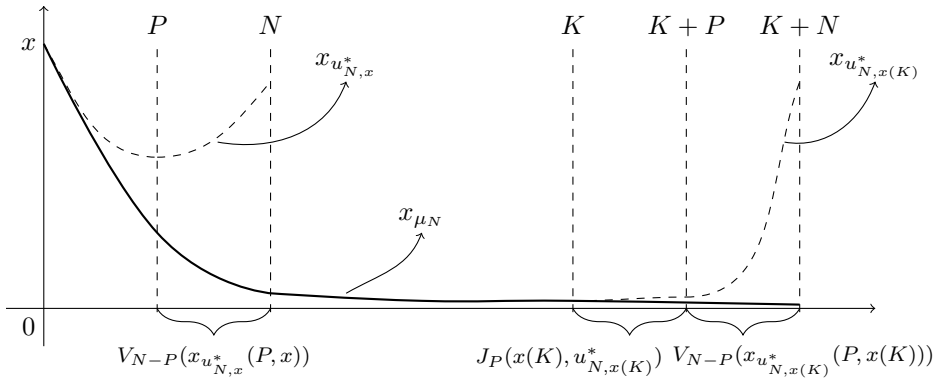


FIGURE 1. Illustration of the quantities used for the computation of the performance indices $\varepsilon^2_{N,P}(K)$ and $\varepsilon^3_{N,P}(K)$.

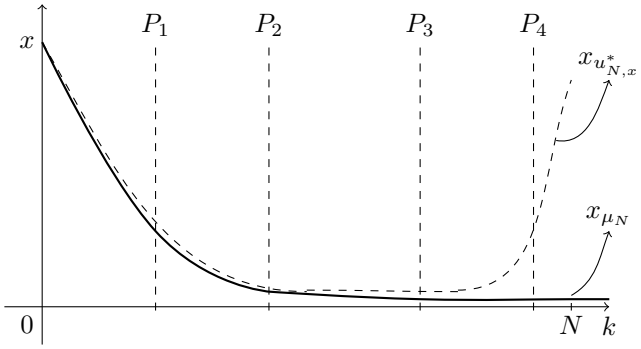


FIGURE 2. Explanation of how the choice of P influences the quantity $E_{N,P}$.

open loop trajectory is very close to the turnpike. If we chose P too small, e.g. $P = P_1$ in the figure, then we will compare the MPC closed loop trajectory to a trajectory that first follows the open loop for P steps and then suddenly jumps to the optimal equilibrium, thus making it an unfair comparison. The quantity $E_{N,P}(K)$ will be reflected by a negative value. Similarly, if P is too big, e.g. $P = P_4$ the trajectory we compare against will include part of the leaving arc of the open loop and so Ideally, P is chosen such that initial the open loop state at time P is very close to the turnpike, so somewhere between P_2 and P_3. We conjecture that the best choice for P, in the sense that it yields the most information, is in the middle of the horizon, so that it can both capture the improvement of the leaving arcs described by $\varepsilon^2_{N,P}$ and the proximity to the optimal equilibrium from $\varepsilon^3_{N,P}$.

Remark 2. While the absolute performance index is more useful than the relative one, there is still room for improvement. Ideally, we would have an estimate that relates the MPC closed loop cost to the cost of an infinite horizon optimal trajectory. For this the following two options could be considered:

1. Using Lemma 4.3 from [4] we can replace the finite horizon optimal control sequence $u^*_{N,x}$ by the infinite horizon control sequence $u^*_{\infty,x}$. This means (21)

becomes

$$J_P(x, u^*_{\infty,x}) + (K-P)\ell(x^e, u^e) - J^{cl}_K(x, \mu_N) = E_{N,P}(K) - R_3(x, P, N). \tag{33}$$

In this case the passing from N to ∞ introduces an additional unknown error $R_3(x, P, N)$, i.e. now $E_{N,P}(K)$ does not exactly describe the difference but is only an approximation. The quality of the approximation depends on both the choice of P and N which is inconvenient if the estimate should be used for tuning the horizon. For large P the error estimate $E_{N,P}(K)$ also contains the cost of the leaving arc while $J_P(x, u^*_{\infty,x})$ does not.

2. Alternatively, one could directly consider the difference

$$J_K(x, u^*_{\infty,x}) - J^{cl}_K(x, \mu_N) \tag{34}$$

and try to derive an error estimate for this quantity. This would be the most powerful tool for rating the MPC performance. Unfortunately, a direct extension of the ideas in the previous section seems out of reach. ◊

3.4. **Performance index for suboptimal solutions.** So far, the performance indices were based on the optimal control sequence u^*_N. Often, the optimal control sequence is not known exactly but only an approximation is computed, which is optimal up to some tolerance. An example for this is when $\tilde{u}$ is a reduced-order suboptimal solution. In an example later on we will apply this for POD as in [12].

In this section we will show that the absolute performance index from Section 3.2 can still be used as a performance indicator, even when only an approximation $\tilde{u}$ of the optimal control sequence u^*_N is available. In this case the components of the performance index ε^1_N, $\varepsilon^2_{N,P}$ and $\varepsilon^3_{N,P}$ are computed using the approximate control sequence $\tilde{u}$. This means equations (19a) through (19c) are changed to

$$\begin{aligned}
\tilde{\varepsilon}^1_N(k) &:= J_N(x(k), \tilde{u}_{x(k)}) - J_N(x(k+1), \tilde{u}_{x(k+1)}) - \hat{\ell}(x(k), \tilde{\mu}(x(k))),\\
\tilde{\varepsilon}^2_{N,P}(K) &:= J_{N-P}(x_{\tilde{u}_x}(P, x), \tilde{u}_x(\cdot + P)) - J_{N-P}(x_{\tilde{u}_{x(K)}}(P, x(K)), \tilde{u}_{x(K)}(\cdot + P)),\\
\tilde{\varepsilon}^3_{N,P}(K) &:= P\ell(x^e, u^e) - J_P(x(K), \tilde{u}_{x(K)})
\end{aligned} \tag{35}$$

where $\tilde{\mu}$ denotes the feedback law defined using the suboptimal solution $\tilde{u}_{x(k)}$ for initial state $x(k)$, i.e. $\tilde{\mu}(k) := \tilde{u}_{x(k)}(0)$.

Going through the proof of Theorem 3.1 we notice that basically the same arguments hold for a suboptimal control sequence. The only thing that changes is that for equation (24) we do not apply the dynamic programming principle but instead the simple observation that

$$J_N(x, \tilde{u}_x) = J_P(x, \tilde{u}_x) + J_{N-P}(x_{\tilde{u}_x}(P, x), \tilde{u}_x(\cdot + P)) \tag{36}$$

which follows directly from the definition of the cost functional (and analog for equation (25)). The main result of Theorem 3.1 then becomes

$$\sum_{k=0}^{K-1} \tilde{\varepsilon}^1_N(k) - \tilde{\varepsilon}^3_{N,P}(K) - \tilde{\varepsilon}^2_{N,P}(K) = J_P(x, \tilde{u}_x) + (K-P)\ell(x^e, u^e) - J^{cl}_K(x, \tilde{\mu}) \tag{37}$$

which estimates how well the closed loop trajectory generated by the suboptimal feedback law $\tilde{\mu}$ performs compared to a trajectory consisting of the very first suboptimal open loop control $\tilde{u}_x$ and of the optimal equilibrium. That said, the performance estimate is limited in the sense that it does not relate the performance of the suboptimal trajectory to the optimal one.

4. **Proper Orthogonal Decomposition.** Suppose that X is a Hilbert space. We indicate with $\langle \cdot, \cdot \rangle_X$ its inner product and with $\| \cdot \|_X$ its induced norm. We define the *snapshots subspace* $\mathcal{X}$ as

$$\mathcal{X} = \operatorname{span} \left\{ x(k) \,\middle|\, k = k_0, k_0 + 1, \dots, k_0 + K \right\} \subset X \tag{38}$$

with $\mathsf{d} = \dim \mathcal{X} \geq 1$. Let $\{\psi_i\}_{i=1}^{\mathsf{d}} \subset X$ be an orthonormal basis for $\mathcal{X}$. Then each snapshot can be expressed as

$$x(k) = \sum_{i=1}^{\mathsf{d}} \langle x(k), \psi_i \rangle_X \, \psi_i \quad \text{for } k = k_0, k_0 + 1, \dots, k_0 + K. \tag{39}$$

The method of proper orthogonal decomposition (POD) consists of selecting an orthonormal basis $\{\psi_i\}_{i=1}^{d}$ in $\mathcal{X}$ such that for every $\ell \in \mathbb{N}$ with $\ell \leq \mathsf{d}$ the mean square error between the snapshots x and their corresponding ℓ-th partial sum of (39) is minimized:

$$\begin{aligned} \min \; & \sum_{k=k_0}^{k_0+K} \theta_k \left\| x(k) - \sum_{i=1}^{\ell} \langle x(k), \psi_i \rangle_X \, \psi_i \right\|_X^2 \\ \text{s.t. } & \{\psi_i\}_{i=1}^{\ell} \subset X \text{ and } \langle \psi_i, \psi_j \rangle_X = \delta_{ij} \text{ for } 1 \leq i, j \leq \ell, \end{aligned} \tag{40}$$

where δ_{ij} is the Kronecker delta and $\theta_k > 0$ are properly chosen positive weights.

Definition 4.1. A POD basis of rank ℓ is a solution $\{\psi_i\}_{i=1}^{\ell}$ to (40). With $X^\ell = \operatorname{span}\{\psi_1, \dots, \psi_\ell\} \subset \mathcal{X} \subset X$ we denote the subspace spanned by the first ℓ POD basis functions.

Using a Lagrangian framework, the solution to (40) is characterized by the following optimality conditions (cf. [2, 9]):

$$\mathcal{R}\psi = \lambda \psi, \tag{41}$$

where the operator $\mathcal{R} : X \to X$ given by

$$\mathcal{R}\psi = \sum_{k=k_0}^{k_0+K} \theta_k \langle x(k), \psi \rangle_X \, x(x) \quad \text{for } \psi \in X$$

is a compact, nonnegative and self-adjoint operator. Therefore, there exists an orthonormal basis $\{\psi_i\}_{i \in \mathbb{N}}$ for X and an associated sequence $\{\lambda_i\}_{i \in \mathbb{N}}$ of nonnegative real numbers so that

$$\mathcal{R}\psi_i = \lambda_i \psi_i, \quad \lambda_1 \geq \cdots \geq \lambda_{\mathsf{d}} > 0 \quad \text{and} \quad \lambda_i = 0, \quad \text{for } i > \mathsf{d}. \tag{42}$$

Moreover, the snapshots subspace is given as $\mathcal{X} = \operatorname{span}\{\psi_i\}_{i=1}^{\mathsf{d}}$. Following [2], the a-priori error formula

$$\sum_{k=k_0}^{k_0+K} \theta_k \left\| x(k) - \sum_{i=1}^{\ell} \langle x(k), \psi_i \rangle_X \, \psi_i \right\|_X^2 = \sum_{i=\ell+1}^{\mathsf{d}} \lambda_i. \tag{43}$$

holds true for the POD basis $\{\psi_i\}_{i=1}^{\ell}$ of rank ℓ.

Remark 3. In our numerical example, (1) describes a semidiscrete PDE. To evaluate $f(x(k), u(k))$ numerically, the space X has to be discretized, e.g., by piecewise finite elements, therefore this procedure can be in general costly. We will utilize the POD method to speed-up the computational time. In Section 5, we will give more details about the reduce order approach in the PDE numerical example setting. $\lozenge$

Algorithm 2 MPC-POD algorithm

Set $k = k_0$ and solve the optimal control problem (9) in order to obtain the optimal control sequence $u^*_{N,x}$.
Solve (1) with $u = u^*_{N,x}$ to generate the POD snapshots $\{x(k)\}_{k=k_0}^{k_0+N}$.
Compute a POD basis $\{\psi_i\}_{i=1}^{\ell}$ of rank ℓ.
Apply the first element of $u^*_{N,x}$ as a control to the system during the next sampling period, i.e. use the feedback law $\mu_N(x) := u^*_{N,x}(0)$.
for each time instant $k = k_0 + 1, \dots$ **do**
1. Measure the current state $x = x(k)$ of the system.
2. Solve the optimal control problem (9), evaluating $f(x(k), u(k))$ with the POD model, in order to obtain the optimal control sequence $\tilde{u}_{N,x}$.
3. Apply the first element of $\tilde{u}_{N,x}$ as a control to the system during the next sampling period, i.e. use the feedback law $\tilde{\mu}_N(x) := \tilde{u}_{N,x}(0)$.

end for

In Algorithm 2 the MPC-POD method is described. We solve the first open-loop optimal control problem (9) approximating the dynamics with the finite element method, in order to generate good snapshots. It is well known, in fact, that the quality of the POD approximation depends on the snapshots subspace; see e.g. [2, 9, 13]. Therefore, using as snapshots the first open-loop optimal solution, we expect that the POD approximation error is sufficiently small for our purpose. Once the POD basis is computed, we solve the next open-loop optimal controls problem on a reduced order level, getting computational speed-up; see Section 5.

5. **Numerical example.** In this section the state variable is indicated with the letter y, due to the fact that the letter $\boldsymbol{x}$ is used to represent the spatial coordinates in $\mathbb{R}^2$. All the tests in this section have been made on a Notebook Lenovo ThinkPad T450s with Intel Core i7-5600U CPU @ 2.60GHz and 12GB RAM. Let $U(N) = L^2(k_0, k_0 + N; \mathbb{R}^m)$ and $X = H^1(\Omega)$, where $\Omega := [0,5] \times [0,5] \subset \mathbb{R}^2$ is a bounded set with Lipschitz-continuous boundary $\Gamma = \Gamma_{\mathsf{c}} \cup \Gamma_{\mathsf{out}}$, where $\Gamma_{\mathsf{c}} \cap \Gamma_{\mathsf{out}} = \emptyset$. We consider the time-dependent bounded linear mapping $\mathcal{F}(\cdot,\cdot) : X \times \mathbb{R}^m \to X'$ as

$$\begin{aligned}\langle \mathcal{F}(\phi, \mathfrak{u}), \varphi \rangle_{X',X} = &- \int_\Omega \nabla\phi \cdot \nabla\varphi + \big(\boldsymbol{v} \cdot \nabla\phi\big)\varphi \, \mathrm{d}\boldsymbol{x} - 5000 \int_{\Gamma_{\mathsf{out}}} \varphi\phi \mathrm{d}\boldsymbol{s} \\ &- \int_{\Gamma_c} \varphi\phi \mathrm{d}\boldsymbol{s} + 5000 y_{\mathsf{out}} \int_{\Gamma_{\mathsf{out}}} \varphi \mathrm{d}\boldsymbol{s} + \sum_{i=1}^{4} \mathfrak{u}_i \int_{\Gamma_c} b_i \varphi \mathrm{d}\boldsymbol{s}\end{aligned}$$

for $\phi, \varphi \in X$, $\mathfrak{u} = (\mathfrak{u}_i) \in \mathbb{R}^m$ a.e. in $[0, +\infty)$. Now, the dynamical system, which we are interested to solve, reads as follow

$$y_t(t) = \mathcal{F}(y(t), u(t)) \in X' \text{ a.e. in } (0, +\infty), \quad y(0) = y_\circ \text{ in } L^2(\Omega). \tag{44}$$

where $y_\circ(\boldsymbol{x}) = 15 + \sin(2\pi x_1)\cos(2\pi x_2)$. Note that the state y solution of (44) is the weak solution of the following convection-diffusion equation

$$\begin{aligned} y_t(t,\boldsymbol{x}) - \Delta y(t,\boldsymbol{x}) + \boldsymbol{v}(\boldsymbol{x})\cdot\nabla y(t,\boldsymbol{x}) &= 0, && \text{a.e. in } [0,+\infty)\times\Omega,\\ \frac{\partial y}{\partial \boldsymbol{n}}(t,\boldsymbol{s}) + y(t,\boldsymbol{s}) &= \sum_{i=1}^{4} u_i(t) b_i(\boldsymbol{s}), && \text{a.e. on } [0,+\infty)\times\Gamma_{\mathsf{c}},\\ \frac{\partial y}{\partial \boldsymbol{n}}(t,\boldsymbol{s}) + 5000 y(t,\boldsymbol{s}) &= 5000 y_{\mathsf{out}}(t), && \text{a.e. on } [0,+\infty)\times\Gamma_{\mathsf{out}},\\ y(0,\boldsymbol{x}) &= y_\circ(\boldsymbol{x}), && \text{a.e. in } \Omega. \end{aligned} \tag{45}$$

This setting represents a squared room Ω, where 4 controls u_i are placed on the boundary Γ_{c} with the following shape functions

$$b_1(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x}\in\{0\}\times[0.0,1.0],\\ 0 & \text{otherwise,}\end{cases} \qquad b_2(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x}\in[1.0,2.0]\times\{1\},\\ 0 & \text{otherwise,}\end{cases}$$
$$b_3(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x}\in\{1\}\times[3.0,4.0],\\ 0 & \text{otherwise,}\end{cases} \qquad b_4(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x}\in[2.0,3.0]\times\{0\},\\ 0 & \text{otherwise.}\end{cases}$$

as shown in Figure 3, while on Γ_{out} we parametrize through Robin boundary conditions the exchange of heat between the outside and the inside of the room. The convection field is shown in Figure 3 and it is generated solving the incompressible Navier-Stokes equation

$$\begin{aligned} v_t + (v\cdot\nabla)v - \nu\Delta v &= -\nabla p && \text{in } Q,\\ \nabla\cdot v &= 0 && \text{in } Q,\\ p &= 0 && \text{in } [0,6)\times\Gamma_{\text{ol}} = \{x_1 = 5.0, x_2\in[4,5]\}\\ v &= \tilde{v} && \text{in } [0,6)\times\Gamma_{\text{il}} = \{x_1 = 0.0, x_2\in[0,1]\}\\ v &= 0 && \text{in } \Gamma - \Gamma_{\text{ol}}\cup\Gamma_{\text{il}}\\ v(0) &= 0 && \text{in } \Omega \end{aligned} \tag{46}$$

up to the time $t = 5.0$ and taking $\boldsymbol{v}(\boldsymbol{x}) = v(5.0,\boldsymbol{x})$. In (46), p is the pressure of the air in the room, $\nu = 0.01$ and

$$\tilde{v}(t,\boldsymbol{x}) = (4.5(4.0x_2(1-x_2)), 0.0).$$

In this scenario, we have an inflow $\tilde{v}(t,\boldsymbol{x})$ on the bottom left side of Ω, which is constant in time and has maximum magnitude of 4.5, and an outflow on the top right part of the domain. Applying the Finite Element method (FE) in space and the implicit Euler method in time we get a discrete trajectory $y(k)$, which is used as sampled solution for the discrete time system (1). For the optimal control problem (9), we choose as stage cost

$$\ell(y_u(k;y_\circ),u(k)) := \frac{1}{2}\sum_{i=1}^{4}\Delta t\,|u_i|^2 + \frac{100}{2}\Delta t\|y_u(k;y_\circ) - y_Q(k)\|_X^2$$

where $y_Q(k) = 18.0$ for all $k\in\mathbb{N}$ and Δt is the time discretization step. To solve the optimal control problem we compute the solution of the optimality system; cf. [16]. As already mentioned in Section 4, we apply the POD method to speed up the computation of the solution of (44). In order to do that, once a POD basis of rank ℓ is computed according to Algorithm 2, we define the bounded linear mapping

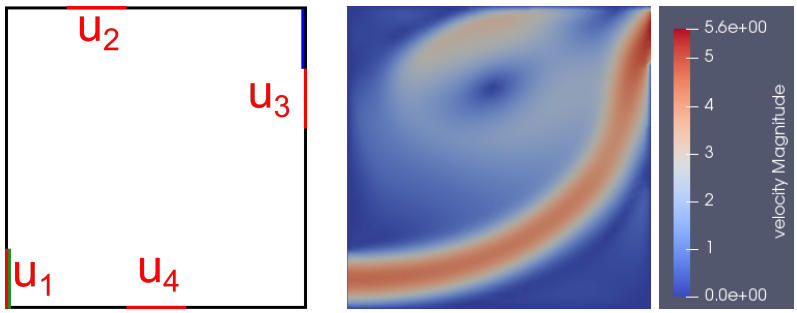


FIGURE 3. Boundary $\Gamma = \Gamma_{\text{out}} \cup \Gamma_c$ and velocity field $\boldsymbol{v}(x)$ (green=inflow, blue= outflow).

$\mathcal{F}^\ell(\cdot,\cdot)$ as

$$\langle \mathcal{F}^\ell(\phi,\mathfrak{u}),\psi\rangle_{X',X} = -\int_\Omega \nabla\phi\cdot\nabla\psi + (\boldsymbol{v}\cdot\nabla\phi)\psi\,\mathrm{d}\boldsymbol{x} - 5000\int_{\Gamma_{\text{out}}}\psi\phi\mathrm{d}\boldsymbol{s} - \int_{\Gamma_c}\psi\phi\mathrm{d}\boldsymbol{s} + 5000 y_{\text{out}}\int_{\Gamma_{\text{out}}}\psi\mathrm{d}\boldsymbol{s} + \sum_{i=1}^4 \mathfrak{u}_i\int_{\Gamma_c} b_i\psi\mathrm{d}\boldsymbol{s}$$

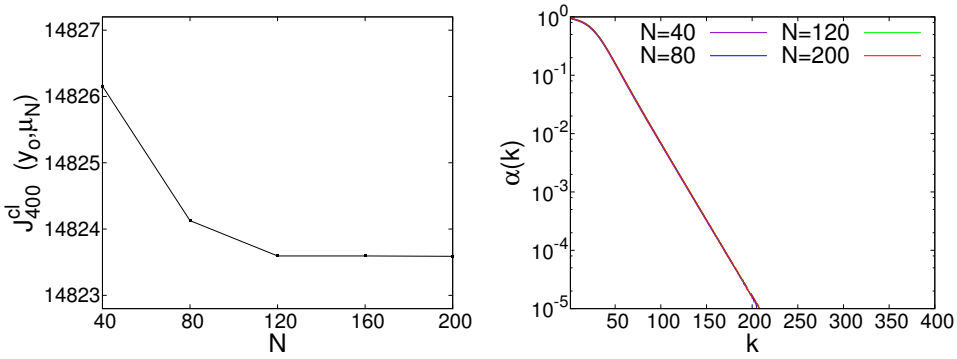
for $\phi\in X$, $\mathfrak{u}=(\mathfrak{u}_i)\in\mathbb{R}^m$ and $\psi\in X^\ell$ a.e. in $[0,+\infty)$. Therefore, the respective reduce order dynamical system is

$$y_t^\ell(t) = \mathcal{F}^\ell(y^\ell(t),u(t)) \in X' \text{ a.e. in } (0,+\infty), \quad y^\ell(0) = y_\circ \text{ in } L^2(\Omega). \tag{47}$$

The advantage in computational speed-up relies on the dimension of the space we use to approximate the solution, in fact in the FE-Galerkin formulation the dimension of the discrete test functions space is equal to the number of FE nodes M, instead for the POD-Galerkin formulation its dimension is $\ell \ll M$.

5.1. **Results.** In the following we present the numerical results as well as some discussion of our interpretations. We start by considering results obtained by solving the optimal control problems in the MPC algorithm using only the FE method. Figure 4 shows the closed loop cost of the MPC (left) and the relative error estimate α (right) that was discussed in Section 3.1. It can be observed that the closed loop cost at time $k = 400$ in Figure 4(a) still improves if we increase the MPC horizon N but this is not reflected by the performance index depicted in Figure 4(b) which for all horizon lengths quickly decays to zero. This confirms our claim that the relative performance index is not helpful in this setting.

The absolute performance indices shown in Figure 5, on the other hand, clearly demonstrate that the performance improves when the MPC horizon is increased. Figure 5(a) shows the sum of the performance indices ε_N^1 up to time K. Recall that the individual values $\varepsilon_N^1(k)$ correspond to the improvement a larger horizon offers over a smaller horizon. Accordingly, the sum measures the accumulated improvement of the performance that would be gained by increasing the horizon.
In Figure 5(b) the quantity $\varepsilon_{N,P}^2(k)$ for a fixed $P = 30$ is depicted. As stated in Section 3.3 this can be interpreted as the improvement of the leaving arc of the MPC open loop trajectories compared to the very first leaving arc. Up to a horizon of $N = 120$ a huge improvement is visible, while for even longer horizon it seems to saturate. This is in accordance with the observations in Figure 4(a).



(a) MPC closed loop cost at the final time $k = 400$ for different horizon lengths N.

(b) Logarithmic plot of the evolution of the relative performance index α for different horizon lengths N.

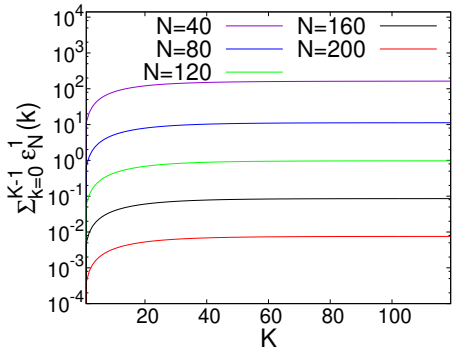
FIGURE 4. MPC closed loop cost and relative performance index

Next, in Figure 5(c) we show the performance index $\varepsilon^3_{N,P}(k)$, again for $P = 30$, whose absolute value can be interpreted as a measure of the proximity of the initial part of the MPC open loop to the optimal equilibrium. Again, we see that for sufficiently long horizon ($N \geq 120$) we arrive in close proximity to the optimal equilibrium. Moreover, as the simulation time continues to increase, Figure 6(a) shows that the error estimate can still effectively distinguish the convergence for different horizon lengths.

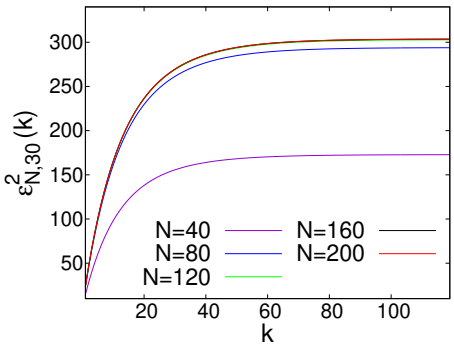
Finally, the last plot in Figure 5 shows the quantity $E_{N,P}(K)$ which is composed of the other error estimates in Figure 5(a) - 5(c). According to Theorem 3.1, this tells us how the MPC closed loop performance compares to the performance of a composite trajectory consisting by parts of the very first open loop and of the optimal equilibrium. We can see from the plot that for sufficiently long simulation time $E_{N,P}$ converges to some value and as the MPC horizon N increases we also observe convergence. Ideally, we would see a convergence to zero, but this can in general not be expected if the P has not been chosen in the right way (cf. the discussion to Figure 2). In fact the value of the performance index $E_{N,P}$ is heavily influenced by the choice of P as seen in Figure 6(b). The figure also shows that for every horizon N choosing a P that is somewhere in the middle of the horizon seems to offer the best results, because then the absolute value of $E_{N,P}$ nicely captures the turnpike behavior of the initial open loop we compare against. This will be investigated in more detail in future work.

In the previous simulations everything was computed using a FE-Galerkin discretization in space. Now, we compare these results with the one obtained using the POD method. To simplify the notation, in this section, we remove the $\sim$ superscript for the suboptimal solution, unless when it is strictly necessary.

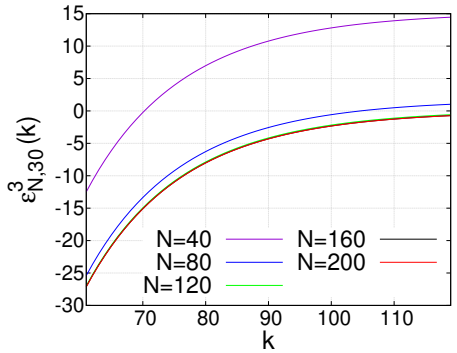
Remark 4. In step 3 of Algorithm 2, we can apply the first element of $u_{N,y_\circ}$ and compute the quantities defined in (35) either with the reduced order model (POD-POD) or the full one (POD-FE). Figure 7 shows that using the full order model improves the results: this is not surprising since we avoid further approximation errors. Let us mention that the POD-FE is not much more costly compared to



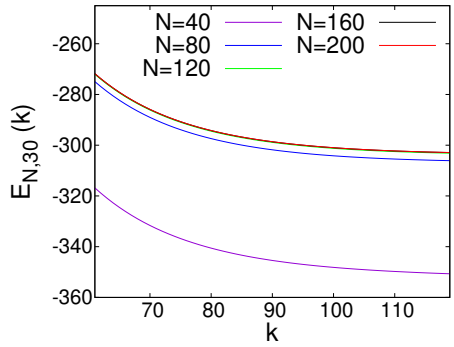
(a) Logarithmic plot of the cumulative performance index ε^1_N over time k for different horizon lengths N.



(b) Plot of the performance index $\varepsilon^2_{N,P}$ for $P = 30$ over time k for different horizon lengths N.



(c) Plot of the performance index $\varepsilon^3_{N,P}$ for $P = 30$ over a selected time window and for different horizon lengths N.

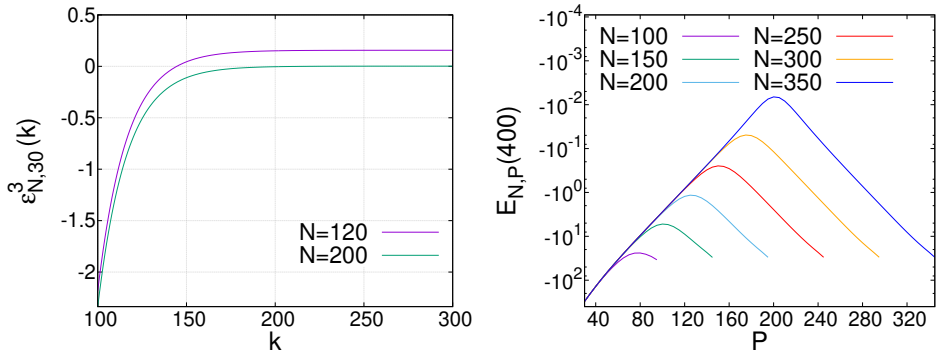


(d) Plot of the quantity $E_{N,P}$ over time k for $P = 30$ and different horizon lengths N.

FIGURE 5. Plots of the individual components of the absolute performance index.

the POD-POD one, since it requires only one extra full-order state equation solve. Moreover, we want to point out that the difference between POD-FE and POD-POD absolute performance indices can be used as an estimate of the POD approximation quality. This evaluation is in general cheaper compared to the a-posteriori error estimator in [15], but such comparison needs further investigations that will be presented in future works. ◊

According to Remark 4, in what follows we mean always the POD-FE method, when we refer to the POD results. As one can see from Figure 8, as soon as the horizon N increases, more POD basis functions are needed to have a good approximation error. In fact, despite the fact that, when $N = 40$, $\ell = 15$ basis guarantee a relative error smaller than 5% for all K, they are not enough when $N = 200$. For this case the relative error between the FE and POD performance index is always close to 1 no matter which K is chosen. With $\ell = 25$ basis, instead, these values are really close to each other. The reason for this behaviour is hidden in Algorithm 2: when N increases more snapshots are included in $\mathcal{X}$ as well as more



(a) Peformance index ε^3_N converging to different values for longer simulation times.

(b) Performance index $E_{N,P}(K)$ at the final time $K = 400$ for varying P and N.

FIGURE 6. Plots of performance index $\varepsilon^3_{N,P}$ for longer simulation times and of the influence of P on the value of $E_{N,P}$.

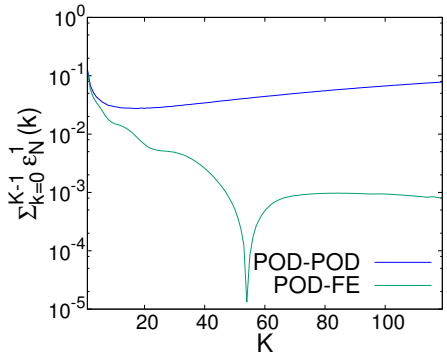
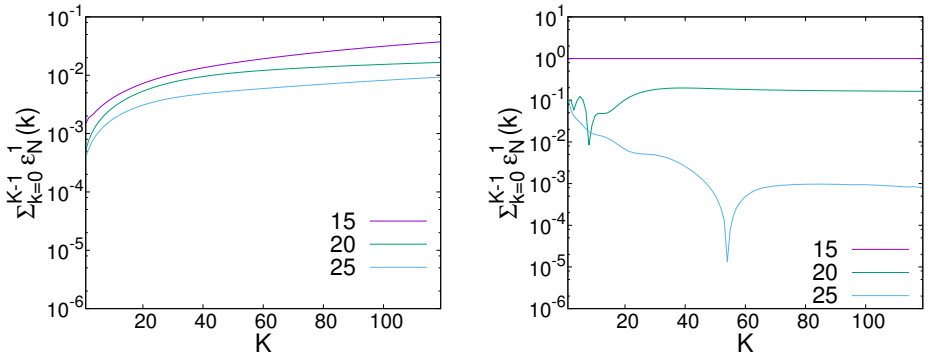


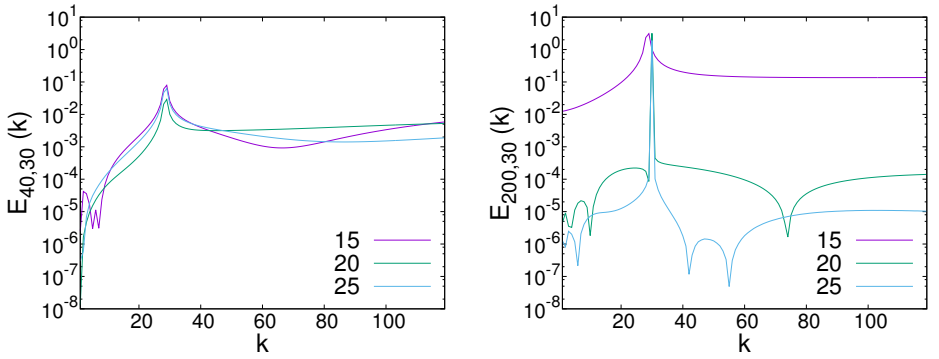
FIGURE 7. Comparison between POD-FE and POD-POD relative errors for $\sum_{k=0}^{K-1} \varepsilon^1_N(k)$ with $\ell = 25$ POD basis functions.

information about the system dynamics for further time instances. This implies that for small N the POD eigenvalues λ_i decay faster and therefore selecting few POD basis functions may already reconstruct the snapshots space according to (43) with a good approximation error. On the other hand, the amount of information contained in the snapshots may not be enough to reconstruct the FE trajectories when k increases, implying that the relative error increases for larger K; cf. Figure 8(a). This is also the reason why for large enough ℓ, e.g. $\ell = 25$, the POD method performs better for larger horizons. Similar considerations can be done for the results shown in Figure 9. Here, a particular behaviour for the relative error can also be observed when $k = P = 30$. In this case, $E_{N,P}(P)$ measures how far the open-loop cost functional $J_P(y_\circ, u^*_{N,y_\circ})$ is from $J^{cl}_P(y_\circ, \mu_N)$, cf. (20). Thus, for sufficiently large N this value is really close to zero and really small (see Table 1), especially compared to its individual components described in (19a)-(19c). This implies a worse POD approximation at this step and thus explains the peaks in the relative error plot Figure 9. As last, in Table 1 we report the relative errors for the control $u^*_{200,y^\circ}(120)$, indicated with rel_err_u, together with the closed-loop



(a) $N = 40$ (b) $N = 200$

FIGURE 8. Relative error between the FE and POD cumulative performance index ε_N^1 over time k for different N and number of POD basis functions $\ell = 15, 20, 25$.



(a) $N = 40$ (b) $N = 200$

FIGURE 9. Relative error between the FE and POD $E_{N,P}$ over time k for $P = 30$, different N and number of POD basis functions $\ell = 15, 20, 25$.

cost $J_{120}^{cl}(y_\circ, \mu_{200})$ and $E_{200,30}(30)$. Moreover, also the computational time to run Algorithm 1 and Algorithm 2 is reported. According to the results, the POD method provides suboptimal results with a small approximation error, gaining a time speed-up. We want to point out that the reported computational time for the POD algorithm includes the time spent to generate the snapshots, as well as the one needed to evaluate the quantities (19a)-(19c) with the full order model.

Conclusion. In this paper we introduced a performance index for economic MPC working on data that is readily available from the MPC algorithm at runtime. We have shown that the performance index exactly reports the difference of the cost of the MPC closed loop trajectory compared to a composite trajectory consisting for one part of an optimal open loop trajectory and for the other part of the optimal equilibrium. By numerical simulations we have verified that the estimator works

TABLE 1. Test results for $N = 200$, $P = 30$ and $K = 120$ for full (FE) and reduced (POD) order models.

Method	ℓ	rel_err_u	$J^{cl}_{120}(y_\circ, \mu_{200})$	$E_{200,30}(30)$	Alg. Time	Speed-up
FE	–	–	7791.565	-0.0014	1182 s	–
POD	20	0.00055	7791.591	0.0031	346 s	3.4
POD	25	0.00008	7791.569	-0.0041	396 s	3.0

well for an example involving a controlled convection-diffusion-equation, both for FE and POD-based discretizations.

We believe that the method presents a versatile tool to evaluate the performance of MPC controllers on-line. For POD based solution it could also be used as an indicator that shows when the POD approximation deteriorates. Accordingly, in the future we will extent the method so that it enables automatic adjustment of the MPC horizon. In addition, we surmise that it is even possible to use the method as an error estimator for POD, which it turn can be used to determine the appropriate number of POD basis functions.

REFERENCES

[1] D. Angeli, R. Amrit, and J. B. Rawlings, On average performance and stability of economic model predictive control, *IEEE transactions on automatic control*, **57** (2012), 1615–1626.

[2] P. Holmes, J. Lumley, G. Berkooz and C. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge, 2012.

[3] L. Grüne, Economic receding horizon control without terminal constraints, *Automatica*, **49** (2013), 725–734.

[4] L. Grüne, Approximation properties of receding horizon optimal control, *Jahresbericht der Deutschen Mathematiker-Vereinigung*, **118** (2016), 3–37.

[5] L. Grüne and J. Pannek, Practical nmpc suboptimality estimates along trajectories. *Systems & Control Letters*, **58** (2009), 161–168.

[6] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control. Theory and Algorithms*, 2^{nd} edition, Springer, 2017.

[7] L. Grüne and S. Pirkelmann, Closed-loop performance analysis for economic model predictive control of time-varying systems, in *Proceedings of the 56th IEEE Conference on Decision and Control (CDC 2017)*, (eds. R. Middleton and D. Nesic), 2017, 5563–5569.

[8] L. Grüne and S. Pirkelmann, Economic model predictive control for time-varying system: Performance and stability results, *Optimal Control Applications and Methods* (2018).

[9] M. Gubisch and S. Volkwein, Proper orthogonal decomposition for linear-quadratic optimal control, in *Model Reduction and Approximation: Theory and Algorithms*, (eds. M. Ohlberger, P. Benner, A. Cohen and K. Willcox), SIAM (2017), Philadelphia, PA, 5–66.

[10] K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems, *Numerische Mathematik 90*, **90** (2001), 117–148.

[11] B. Lincoln and A. Rantzer, Relaxing dynamic programming, *IEEE Transactions on Automatic Control*, **51** (2006), 1249–1260.

[12] L. Mechelli and S. Volkwein, POD-based economic model predictive control for heat-convection phenomena, *Lecture Notes in Computational Science and Engineering*, **126** (2019), 663–670.

[13] L. Mechelli and S. Volkwein, POD-based economic optimal control of heat-convection phenomena, in Numerical Methods for Optimal Control Problems, (M. Falcone, R. Ferretti, L. Grüne and W.M. McEneaney), Springer (2018), 63–87.

[14] M. A. Müller and L. Grüne, Economic model predictive control without terminal constraints for optimal periodic behavior, *Automatica*, **70** (2016), 128–139.

[15] F. Tröltzsch and S. Volkwein, POD a-posteriori error estimates for linear-quadratic optimal control problems, *Computational Optimization and Applications*, **44** (2009),83–115.

[16] F. Tröltzsch, *Optimal Control of Partial Differential Equations. Theory, Methods and Applications*, American Math. Society, Providence, 2010.

[17] M. Zanon, L. Grüne, and M. Diehl, Periodic optimal control, dissipativity and mpc, *IEEE Transactions on Automatic Control*, **62** (2017), 2943–2949.

E-mail address: lars.gruene@uni-bayreuth.de
E-mail address: luca.mechelli@uni-konstanz.de
E-mail address: simon.pirkelmann@uni-bayreuth.de
E-mail address: stefan.volkwein@uni-konstanz.de