Trajectory based suboptimality estimates for receding horizon controllers

L.Grüne[†], J. Pannek[‡]

[†] *Mathematical Institute, University of Bayreuth, Germany,* lars.gruene@uni-bayreuth.de
[‡] *Mathematical Institute, University of Bayreuth, Germany,* juergen.pannek@uni-bayreuth.de

Keywords : nonlinear control, sampled-data, model predictive control, suboptimality
In this paper we develop and illustrate methods for estimating the degree of suboptimality of receding horizon schemes with respect to infinite horizon optimal control. The proposed *a posteriori* and *a priori* methods yield estimates which are evaluated online along the computed closed–loop trajectories and only use numerical information which is readily available in the scheme.

1 Introduction

Receding horizon control (RHC), often also termed model predictive control (MPC), is by now a well established method for the optimal control of linear and nonlinear systems [1, 2, 14]. The method approximates the solution to a infinite horizon optimal control problem which is computationally intractable in general by a sequence of finite horizon optimal control problems. Then the first element of the resulting control sequence is implemented in each time step which generates a closed–loop static state feedback.
The approximation of the infinite horizon problem naturally leads to the question about the suboptimality of the resulting MPC feedback. Hence our main task is to give estimates of the degree of suboptimality — and implicitly for stability — of the MPC feedback with respect to the original infinite horizon cost functional. This matter was treated in a number of papers, see e.g. [4, 6–8, 11, 17]. Here we deal with discrete–time nonlinear systems on arbitrary metric spaces and use finite horizon optimal control problems without terminal costs or terminal constraints. For these schemes, we present techniques for estimating the degree of suboptimality online along the closed–loop trajectory. The techniques rely on the computation of a "characteristic value" α at each time instant n along the closed–loop trajectory $x(n)$ and the actual estimate can then be computed from the collection of all these α-values. Like in [6] or [8], our approach is based on relaxed dynamic programming techniques.
The motivation for this work is twofold: on the one hand, we expect trajectory based estimates to be less conservative than the global estimates derived, e.g., in [6], [8] or [17], because in these references the worst case over the whole state space is estimated while here we only use those points of the state space which are actually visited by the closed–loop trajectory. On the other hand, we expect that our trajectory based estimates can be used as a building block for MPC schemes in which the optimization horizon is tuned adaptively, similar to adaptive step size control in numerical schemes for differential equations. In this context, the computational cost for evaluating our estimates is a crucial point and this is where the two techniques we present differ. While the first estimation technique yields a sharper estimate, it can only be evaluated *a posteriori*, i.e., the value α for time n can only be computed at time $n + 1$. In contrast to this, the second technique leads to a more conservative estimate of α but is computable with small effort from values which are known at time n.
The paper is organized as follows. In Section 2 we describe the problem setup and give the basic relaxed dynamic programming inequality which leads to our first estimation method. In the following Section 3 we state our main theorem which leads to an alternative estimation method. In Section 4 we illustrate both methods by means of a numerical simulation. The final Section 5 concludes the paper.

2 Problem formulation

Throughout this paper the nonlinear discrete–time system

$$x(n + 1) = f(x(n), u(n)), \quad x(0) = x_0 \tag{2.1}$$

with $x(n) \in X$ and $u(n) \in U$ for $n \in \mathbb{N}$ will the basis of this analysis. Here the state space X is an arbitrary metric space and we denote the space of control sequences $u : \mathbb{N}_0 \to U$ by $\mathcal{U}$.

Remark 2.1. *In particular our setting includes discrete–time dynamics induced by a sampled infinite dimensional system, cf. [9] for a continuous–time analysis of this setting and [8] for a numerical example.*

For this control system we want to find a static state feedback $u = \mu(x) \in \mathcal{U}$ which minimizes the infinite horizon cost functional $J_\infty(x_0, u) = \sum_{n=0}^{\infty} l(x_u(n), u(n))$ with stage cost $l : X \times U \to \mathbb{R}_0^+$ and optimal value function $V_\infty(x_0) = \inf_{u \in \mathcal{U}} J_\infty(x_0, u)$. Here and in the following we will assume that the minimum with respect to $u \in \mathcal{U}$ is attained for reasons of simplicity.
In order to avoid the problem of solving an infinite horizon optimal control problem which necessarily involves the solution of a Hamilton–Jacobi–Bellman partial differential equation we will use a receding horizon approach and replace the previously stated problem by a sequence of finite horizon optimal control problems. For this purpose we minimize the truncated cost functional $J_N(x_0, u) = \sum_{n=0}^{N-1} l(x_u(n), u(n))$ and denote the associated optimal value function by $V_N(x_0) = \min_{u \in \mathcal{U}} J_N(x_0, u)$. Moreover we will use the abbreviation

$$u_N(x_0, \cdot) = \text{argmin}_{u \in \mathcal{U}} J_N(x_0, u) \tag{2.2}$$

for the minimizing open–loop control sequence of the reduced cost functional. This control gives us the optimal open–loop solution

$$x_{u_N}(n+1, x_0) = f\left(x_{u_N}(n, x_0), u_N(x_{u_N}(0, x_0), n)\right), \quad x_{u_N}(0, x_0) = x_0,\ n = 0, \ldots, N-1 \tag{2.3}$$

where $u_N(x_0, n)$ represents the n-th control value within the open–loop control sequence corresponding to the initial value x_0.
In order to obtain an infinite control sequence from this setting we define a feedback law μ_N by implementing only the first element of the optimal control sequence u_N. This is equivalent to defining μ_N via Bellman's optimality principle for the optimal value function V_N, i.e.

$$\mu_N(x(n)) := \text{argmin}_{u \in U} \left\{ V_{N-1}(x(n+1)) + l(x(n), u) \right\}. \tag{2.4}$$

In the literature this setup is usually called nonlinear model predictive control (NMPC) or receding horizon control (RHC). The resulting closed–loop trajectory will be denoted by

$$x(n+1) = f\left(x(n), \mu_N(x(n))\right), \quad x(0) = x_0,\ n \in \mathbb{N}_0. \tag{2.5}$$

Our intention is to give an estimate on the degree of suboptimality of the feedback μ_N for the infinite horizon problem which can be evaluated online along the closed–loop trajectory (2.5) without significant additional computational costs. More precisely, if we define the infinite horizon cost corresponding to μ_N by $V_\infty^{\mu_N}(x_0) := \sum_{n=0}^{\infty} l\left(x(n), \mu_N(x(n))\right)$, then we are interested in upper bounds for this infinite horizon value, either in terms of the finite horizon optimal value function V_N or in terms of the infinite horizon optimal value function V_∞. In particular, the latter will give us estimates about the degree of suboptimality of the controller μ_N in the actual step of the NMPC process.
The main tool we are going to use for this purpose is a rather straightforward and easily proved "relaxed" version of the dynamic programming principle. This fact has been used implicitly in many papers on dynamic programming techniques during the last decades. Recently, it has been studied by Lincoln and Rantzer in [12, 16].

Proposition 2.2. *Consider the MPC feedback law $\mu_N : X \to U$ from (2.4) and its associated trajectory $x(\cdot)$ according to (2.5) with initial value $x(0) = x_0 \in X$. If the inequality*

$$V_N(x(n)) \geq V_N(x(n+1)) + \alpha l(x(n), \mu_N(x(n))) \tag{2.6}$$

holds for some $\alpha \in (0, 1]$ and all $n \in \mathbb{N}_0$ then $\alpha V_\infty(x(n)) \leq \alpha V_\infty^{\tilde{\mu}}(x(n)) \leq V_N(x(n)) \leq V_\infty(x(n))$ holds for all $n \in \mathbb{N}_0$.

Proof. The proof is similar to that of [16, Proposition 3] and [6, Proposition 2.2]. Rearranging (2.6) and summing over n we obtain the upper bound

$$\alpha \sum_{j=n}^{K-1} l(x(j), \mu_N(x(j))) \leq V_N(x(n)) - V_N(x(K)) \leq V_N(x(n)).$$

Hence, taking $K \to \infty$ gives us our assertion since the final inequality $V_N \leq V_\infty$ is obvious. □

Remark 2.3. *Note that in this formulation α only depends on the points $x(n)$, while in [6, Proposition 2.2] it depends on all $x \in X$. Hence we expect a less conservative approximation of the degree of suboptimality.*

Since all values in (2.6) are available at runtime, the value α can be easily computed online along the closed–loop trajectory and thus (2.6) yields a computationally feasible and numerically cheap way to estimate the suboptimality of the trajectory.
Under suitable controllability assumptions, one can show that $\alpha \to 1$ as $N \to \infty$, cf. [6, 8]. Hence, the knowledge of α can in principle be used to adapt the optimization horizon N online by increasing N if the computed α is too small.

However, using (2.6), for the computation of α for the state $x(n)$ we need to know $V_N(x(n+1))$. At time n, this value can in principle be obtained by solving an additional optimal control problem. Proceeding this way, however, essentially doubles the computational effort and may thus not be feasible in real time applications. If we want to use only those numerical information which is readily available at time n then we will have to wait until time $n+1$ before the quality of the MPC feedback value $\mu_N(x(n))$ can be evaluated. In other words, (2.6) yields an *a posteriori* estimator, which is an obvious disadvantage if α is to be used for an online adaptation of N at time n. In the next section we present an alternative way in order to estimate α.

3 An estimation method for α

This section aims at reducing the amount of information necessary to give an estimate of the degree of suboptimality of the trajectory (2.4), (2.5) under consideration. Here we are interested in avoiding the use of future information, i.e., of $V_N(x(n+1))$, in our calculations. Of course, this will in general yield a more conservative estimate.
The following estimates are similar to certain results in [6], where, however, they were defined and used globally for all $x \in X$. In order to make those results computable without using a discretization of the state space X, here we formulate and prove alternative versions of these results which can be used along trajectories.

Lemma 3.1. *Consider $N \in \mathbb{N}$, a receding horizon feedback law μ_N and its associated closed–loop solution $x(\cdot)$ according to* (2.5) *with initial value $x(0) = x_0$. If*

$$V_N(x(n+1)) - V_{N-1}(x(n+1)) \leq (1-\alpha)l(x(n), \mu_N(x(n))) \tag{3.1}$$

holds for some $\alpha \in (0,1]$ and all $n \in N$, then V_N and μ_N satisfy (2.6) *and $\alpha V_\infty(x(n)) \leq \alpha V_\infty^{\mu_N}(x(n)) \leq V_N(x(n)) \leq V_\infty(x(n))$ holds for all $n \in \mathbb{N}_0$.*

Proof. Using the principle of optimality we obtain

$$V_N(x(n)) = l(x(n), \mu_N(x(n))) + V_{N-1}(x(n+1)) \overset{(3.1)}{\geq} V_N(x(n+1)) + \alpha l(x(n), \mu_N(x(n)))$$

Hence (2.6) holds with $\widetilde{V} = V_N$, $\tilde{\mu} = \mu_N$ and Proposition 2.2 guarantees the assertion. □

The following assumption contains the main ingredients for our result.

Assumption 3.2. *For given N, $N_0 \in \mathbb{N}$, $N \geq N_0 \geq 2$ there exists $\gamma > 0$ such that the inequalities*

$$\frac{V_{N_0}(x_{u_N}(N-N_0, x(n)))}{\gamma+1} \leq \max_{j=2,\ldots,N_0} l(x_{u_N}(N-j, x(n)), \mu_{j-1}(x_{u_N}(N-j, x(n)))) \tag{3.2}$$

$$\frac{V_k(x_{u_N}(N-k, x(n)))}{\gamma+1} \leq l(x_{u_N}(N-k, x(n)), \mu_k(x_{u_N}(N-k, x(n)))) \tag{3.3}$$

hold for all $k \in \{N_0+1, \ldots, N\}$ and all $n \in \mathbb{N}_0$ where $x_{u_N}(\cdot, x(n))$ is the optimal open–loop solution from (2.3) *starting in $x(n)$ and $x(\cdot)$ is the MPC closed–loop solution from* (2.5).

Remark 3.3. *(i) Assumption 3.2 generalizes [6, Assumption 4.6] in which $N_0 = 2$ was used. In the numerical example, below, we will see that a judicious choice of N_0 can considerably improve our estimates.*
(ii) Assumption 3.2 involves both the state of the closed–loop trajectory $x(\cdot)$ from (2.5) *at time n and the open–loop trajectory $x_{u_N}(\cdot, x(n))$ from* (2.3) *starting in $x(n)$. Note that typically these two trajectories do not coincide. However, both are available in the MPC scheme at time n once the finite horizon optimization problem with initial value $x(n)$ is solved. From these, the optimal value functions on the left hand sides of the inequalities* (3.2) *and* (3.3) *are easily computed, since by Bellman's optimality principle they can be obtained by simply summing up the running cost along the "tails" of the optimal trajectory $x_{u_N}(\cdot, x(n))$. The only data which is not immediately available is the control value $\mu_{j-1}(x_{u_N}(N-j, x(n)))$ in* (3.2) *which needs to be determined by solving an optimal control problem with horizon $j-1 \leq N_0 - 1$. Since typically N_0 is considerable smaller than N, this can be done with much less effort than computing $V_N(x(n+1))$. Furthermore, if l is independent of u (as in our numerical example) then the control value is not needed at all and thus γ can be computed directly from the data available at time n.*

Proposition 3.4. *Consider $N \geq N_0 \geq 2$ and assume that Assumption 3.2 holds for these constants. Then*

$$\frac{(\gamma+1)^{N-N_0}}{(\gamma+1)^{N-N_0} + \gamma^{N-N_0+1}} V_N(x(n)) \leq V_{N-1}(x(n))$$

holds for all $n \in \mathbb{N}_0$.

Proof. In the following we use the abbreviation $x_{u_N}(j) := x_{u_N}(j, x(n))$, $j = 0, \ldots, N$, since all our calculations using the open–loop trajectory defined by (2.2), (2.3) refer to the fixed initial value $x(n)$.
Set $\tilde{n} := N - k$. First we will prove

$$V_{k-1}(f(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n})))) \leq \gamma l(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n}))) \tag{3.4}$$

for all $k \in \{N_0, \ldots, N\}$ and all $n \in \mathbb{N}$. Using the principle of optimality and Assumption 3.2 we obtain

$$V_{k-1}(f(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n})))) = V_k(x_{u_N}(\tilde{n})) - l(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n}))) \overset{(3.3)}{\leq} \gamma l(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n})))$$

Now we will prove the main assertion by induction over $k = N_0, \ldots, N$. For notational reason we will use the abbreviation $\eta_k = \frac{(\gamma+1)^{k-N_0}}{(\gamma+1)^{k-N_0}+\gamma^{k-N_0+1}}$ to prove $\eta_k V_k(x_{u_N}(\tilde{n})) \leq V_{k-1}(x_{u_N}(\tilde{n}))$ for $k = N_0, \ldots, N$. For $k = N_0$ we obtain this via

$$\begin{aligned} V_{N_0}(x_{u_N}(N - N_0)) &\overset{(3.2)}{\leq} (\gamma + 1) \max_{j=2,\ldots,N_0} l(x_{u_N}(N - j), \mu_{j-1}(x_{u_N}(N - j))) \\ &\leq (\gamma + 1) \sum_{j=2}^{N_0} l(x_{u_N}(N - j), \mu_{j-1}(x_{u_N}(N - j))) = \frac{1}{\eta_{N_0}} V_{N_0-1}(x_{u_N}(N - N_0)). \end{aligned}$$

For the induction step $k \to k + 1$ the following holds

$$\begin{aligned} V_k(x_{u_N}(\tilde{n})) &= V_{k-1}(f(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n})))) + l(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n}))) \\ &\overset{(3.4)}{\geq} \left(1 + \frac{1-\eta_k}{\gamma+\eta_k}\right) V_{k-1}(f(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n})))) + \left(1 - \gamma\frac{1-\eta_k}{\gamma+\eta_k}\right) l(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n}))) \\ &\overset{\text{IS}}{\geq} \eta_k \left(1 + \frac{1-\eta_k}{\gamma+\eta_k}\right) V_k(f(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n})))) + \left(1 - \gamma\frac{1-\eta_k}{\gamma+\eta_k}\right) l(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n}))) \\ &= \eta_k \frac{\gamma+1}{\gamma+\eta_k} \left\{V_k(f(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n})))) + l(x_{u_N}(\tilde{n}), \mu_k(x_{u_N}(\tilde{n})))\right\} = \eta_k \frac{\gamma+1}{\gamma+\eta_k} V_{k+1}(x_{u_N}(\tilde{n})) \end{aligned}$$

with

$$\eta_k \frac{\gamma+1}{\gamma+\eta_k} = \frac{(\gamma+1)^{k-2}}{(\gamma+1)^{k-2}+\gamma^{k-1}} \frac{\gamma+1}{\gamma + \frac{(\gamma+1)^{k-2}}{(\gamma+1)^{k-2}+\gamma^{k-1}}} = \frac{(\gamma+1)^{k-1}}{(\gamma+1)^{k-1}+\gamma^k} = \eta_{k+1}.$$

If we now insert $k = N$, i.e. $\tilde{n} = 0$, we obtain the desired inequality for $x_{u_N}(0) = x_{u_N}(0, x(n)) = x(n)$. Since n was arbitrary, this yields the assertion. □

Theorem 3.5. *Consider $\gamma > 0$ and N, $N_0 \in \mathbb{N}$, $N \geq N_0$ such that $(\gamma + 1)^{N-N_0} > \gamma^{N-N_0+2}$ holds. If Assumption 3.2 is fulfilled for these γ, N and N_0, then the estimate*

$$\alpha V_\infty^{\mu_N}(x(n)) \leq V_N(x(n)) \leq V_\infty(x(n)) \quad \textit{with} \quad \alpha = \frac{(\gamma+1)^{N-N_0} - \gamma^{N-N_0+2}}{(\gamma+1)^{N-N_0}} \tag{3.5}$$

holds for all $n \in \mathbb{N}$.

Proof. Using Proposition 3.4 we get

$$V_N(x(n)) - V_{N-1}(x(n)) \leq \left(\frac{(\gamma+1)^{N-N_0} + \gamma^{N-N_0+1}}{(\gamma+1)^{N-N_0}} - 1\right) V_{N-1}(x(n)) = \frac{\gamma^{N-N_0+1}}{(\gamma+1)^{N-N_0}} V_{N-1}(x(n)).$$

Considering $j = n - 1$ we obtain the open–loop expression

$$V_N(x(j+1)) - V_{N-1}(x(j+1)) \leq \frac{\gamma^{N-N_0+1}}{(\gamma+1)^{N-N_0}} V_{N-1}(f(x_{u_N}(0, x(j)), \mu_N(x_{u_N}(0, x(j))))).$$

Now we can use (3.4) with $k = N$ and get

$$V_N(x(j+1)) - V_{N-1}(x(j+1)) \leq \frac{\gamma^{N-N_0+2}}{(\gamma+1)^{N-N_0}} l(x_{u_N}(0, x(j)), \mu_N(x_{u_N}(0, x(j)))) = \frac{\gamma^{N-N_0+2}}{(\gamma+1)^{N-N_0}} l(x(j), \mu_N(x(j))).$$

Hence the assumptions of Lemma 3.1 are fulfilled with $\alpha = 1 - \frac{\gamma^{N-N_0+1}}{(\gamma+1)^{N-N_0}} = \frac{(\gamma+1)^{N-N_0} - \gamma^{N-N_0+2}}{(\gamma+1)^{N-N_0}}$. □

Theorem 3.5 immediately leads to our second suboptimality estimate: at each time instant n we can compute γ from the inequalities (3.2) and (3.3) (cf. Remark 3.3) and then compute α according to (3.5). In contrast to computing α directly from (2.6), we obtain a criterion for the quality of $\mu_N(x(n))$ which is computable with small computational effort from the data available at time n (cf. Remark 3.3(ii)), i.e., we obtain an *a priori* estimate which is available before the current step is actually carried out.

Remark 3.6. *(i) Asymptotic stability can be concluded from our suboptimality results if the running cost l is positive definite, for details see [8]. Furthermore our results can be extended to practical optimality and stability similar to [6].*
(ii) Another way of numerically computing suboptimality estimates was presented in [17] for linear finite dimensional system. The main difference to our approach is that the condition in [17] has to be verified by computing numerical approximations to the optimal value functions, which is feasible only for low dimensional linear systems but infeasible in our nonlinear setting on arbitrary metric spaces.

4 Numerical Experiments

In order to illustrate our results we consider a digital redesign problem (cf. [15]) of an arm/rotor/platform (ARP) model:

$$\begin{aligned}
\dot{x}_1 &= x_2 + x_6 x_3 & \dot{x}_5 &= x_6\\
\dot{x}_2 &= -\tfrac{k_1}{M}x_1 - \tfrac{b_1}{M}x_2 + x_6 x_4 - \tfrac{mrb_1}{M^2}x_6 & \dot{x}_6 &= -a_1 x_5 - a_2 x_6 + a_1 x_7 + a_3 x_8 - p_1 x_1 - p_2 x_2\\
\dot{x}_3 &= -x_6 x_1 + x_4 & \dot{x}_7 &= x_8\\
\dot{x}_4 &= -x_6 x_2 - \tfrac{k_1}{M}x_3 - \tfrac{b_1}{M}x_4 + \tfrac{mrk_1}{M^2} & \dot{x}_8 &= a_4 x_5 + a_5 x_6 - a_4 x_7 - (a_5 + a_6)x_8 + \tfrac{1}{J}u
\end{aligned}$$

For this system a continuous-time full-state feedback u_0 was designed via backstepping such that the output $\zeta := x_5 - \frac{a_3}{a_1 - a_2 a_3}[x_6 - a_3 x_7]$ is close to x_5 and tracks a given reference signal $\zeta_{\text{ref}}(t) = \sin(t)$, see [3, Chapter 7.3.2] for details on the backstepping design and the specification of the model parameters. In the MPC redesign we now use the trajectory of the continuously controlled system as a reference trajectory for the MPC scheme in order to compute a sampled-data feedback which tracks the continuous time behavior. To this end we denote the reference solution generated by the continuous time system $x_{\text{ref}}(\cdot)$.
We set the initial value to $x(t_0) = (0, 0, 0, 0, 10, 0, 0, 0)$, the absolute and relative tolerances for the solver of the differential equation as well as the accuracy of the optimization routine to 10^{-6}, the length of the open–loop horizon within the MPC–algorithm to $H = N \cdot T$ with $N = 5$ and sampling period $T = 0.3$. Moreover we use the cost functional $J(x, u) = \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} |x_5(t) - x_{5,\text{ref}}(t)| dt$.

		Prop. 2.2	Theorem 3.5, $N_0 = 2$		Theorem 3.5, $N_0 = 4$	
n	$V_N(x(n))$	α	α	γ	α	γ
1	0.55878228	0.99933183	0.41130899	1.59325108	0.99452337	0.18661250
2	0.07950916	0.99904250	0.24403255	1.72584196	0.98682604	0.25473550
3	0.01103795	0.99889255	-0.19707587	2.00666873	0.99126149	0.22009511
4	0.00200117	0.99951566	0.53980747	1.47468820	0.98390720	0.27367155
5	0.00031303	0.99795730	0.93745853	0.82379521	0.98955100	0.23452637
6	5.9986e-05	0.96752356	-3.8274e+02	2.1005e+01	0.99329164	0.20044056
7	1.1639e-05	0.98150222	-3.4011e+02	1.9880e+01	-4.22481242	2.67868057
8	2.9932e-06	0.63738607	-5.4845e+02	2.4868e+01	-2.80911012	2.33280061
9	2.4208e-06	-0.93289676	-9.6362e+02	3.2503e+01	-1.0239e+03	3.2503e+01
10	2.4882e-06	0.67495857	0.84640267	1.06102880	0.62356666	0.89323127

Table 1: Comparing results from Proposition 2.2 and Theorem 3.5 for $N = 5$ and various N_0

Here the α–values shown in the table for Proposition 2.2 for each time n are computed from (2.6) for this n, i.e., these are the α–values which are computed from $V_N(x(n + 1))$ and thus become available at time $n + 1$.
In our simulation, the exact α-values from Proposition 2.2 are close to one for the first iteration steps indicating that the feedback is almost infinite horizon optimal. However, from iteration step 8 onwards the values become smaller and even negative which shows that optimality (even approximate optimality) is lost here. The reason for this is that here the values of V_N are close to the accuracy of the optimization routine and the tolerances of the solver of the differential equations, hence numerical errors become dominant. Nevertheless, the measured values of α in conjunction with the values of V_N show that the closed loop system behaves "almost optimal" until a very small neighborhood of the reference trajectory is reached.
The numerical results also reveal that for $N_0 = 4$ the estimated α-values from Theorem 3.5 yield good estimates for the exact values until the neighborhood of the reference trajectory is reached. While in this simulation $N_0 = 4$ is

clearly preferable, further numerical experiments have shown that one can not specify a “best” N_0 in advance. This is reasonable since α in (3.5) is monotone in γ and N_0 separately. However, one may use a simple optimization over N_0 in order to obtain the “best” possible estimate α in the sense of Assumption 3.2 and Theorem 3.5.

5 Conclusion

We have presented two methods for the online estimation of the suboptimality of MPC schemes along trajectories. In both methods the estimation is based exclusively on numerical values which are readily available in the scheme. While the first method produces tighter estimates, it only allows to assess the quality of the n-th step *a posteriori*, i.e., at time $n+1$. In contrast to this our second method allows for an estimation at time n with small computational effort. Future research will aim at the design of algorithms which adaptively choose suitable optimization horizons N based on these estimates.

References

[1] F. Allgöwer, A. Zheng, *Nonlinear model predictive control*, Progress in Systems and Control Theory, 26, Birkhäuser Verlag: Basel (2000).

[2] E. F. Camacho, C. Bordons, *Model predictive control*, Springer Verlag: London (2004).

[3] R. A. Freeman, P. V. Kokotović, *Robust nonlinear control design*, Birkhäuser: Boston (1996).

[4] F. Di Palma, L. Magni, *On optimality of nonlinear model predictive control*, Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems, Leuven, Belgium (2004), CD–ROM, Session MP8, Paper No. 238.

[5] G. Grimm, M. J. Messina, A. R. Teel, S. Tuna, *Model predictive control: for want of a local control Lyapunov function, all is not lost*, IEEE Trans. Automat. Contr., 50(5) (2005) pp. 546–558.

[6] L. Grüne, A. Rantzer, *On the infinite horizon performance of receding horizon controllers*, IEEE Trans. Automat. Control, (2008) to appear.

[7] L. Grüne, *Optimization based stabilization of nonlinear control systems*, in: I. Lirkov et al., eds., Large-Scale Scientific Computations (LSSC07), Springer LNCS 4818, to appear

[8] L. Grüne, *Analysis and design of unconstrained nonlinear MPC schemes for finite and infinite dimensional systems*, (2007) submitted.

[9] K. Ito, K. Kunisch, *Receding horizon optimal control for infinite dimensional systems*, ESAIM Control Optim. Calc. Var., 8 (2002) , pp. 741–760, A tribute to J. L. Lions.

[10] A. Jadbabaie, J. Hauser, *On the stability of receding horizon control with a general terminal cost*, IEEE Trans. Automat. Contr., 50 (2005), pp. 674–678.

[11] S. S. Keerthy, E. G. Gilbert, *Optimal infinite horizon feedback laws for a general class of constrained discrete–time systems*, J. Optimiz. Theory Appl., 57 (1988), pp. 265–293.

[12] B. Lincoln, A. Rantzer, *Relaxing dynamic programming*, IEEE Trans. Automat. Control, 51 (2006), pp. 1249–1260.

[13] L. Magni, R. Sepulchre, *Stability margins of nonlinear receding-horizon control via inverse optimality*, Systems Control Lett., 32 (1997), pp. 241–245.

[14] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, *Constrained model predictive control: stability and optimality*, Automatica, 36 (2000), pp. 789–814.

[15] D. Nešić, L. Grüne, *A receding horizon control approach to sampled-data implementation of continuous-time controllers*, Systems & Control Letters 55 (2006), pp. 660–672

[16] A. Rantzer, *Relaxed dynamic programming in switching systems*, IEE Proceedings — Control Theory and Applications, 153 (2006), pp. 567–574.

[17] J. S. Shamma, D. Xiong, *Linear nonquadratic optimal control*, IEEE Trans. Automat. Control, 42 (1997), pp. 875–879.