Chapter 7
Variants and Extensions

The results developed so far in this book can be extended in many ways. In this chapter we present a selection of possible variants and extensions. Some of these introduce new combinations of techniques developed in the previous chapters, others relax some of the previous assumptions in order to obtain more general results or strengthen assumptions in order to derive stronger results. Several sections contain algorithmic ideas which can be added on top of the basic NMPC schemes from the previous chapters. Parts of this chapter contain results which are somewhat preliminary and are thus subject to further research. Some sections have a survey like style and, in contrast to the other chapters of this book, proofs are occasionally only sketched with appropriate references to the literature.

7.1 Mixed Constrained–Unconstrained Schemes

The previous Chaps. 5 and 6 have featured two extreme cases, namely NMPC schemes with terminal constraints $\mathbb{X}_0$ *and* costs F on the one hand and schemes without both $\mathbb{X}_0$ and F on the other hand. However, it appears natural to consider also intermediate or mixed cases, namely schemes in which (nonequilibrium) terminal constraint sets $\mathbb{X}_0$ but no terminal costs F are used and schemes in which terminal costs F but no terminal constraints sets $\mathbb{X}_0$ are used.

Schemes with terminal constraints $\mathbb{X}_0$ but without terminal costs F appear as a special case of Algorithm 3.10 (or its time varying counterpart 3.11) with ($\mathrm{OCP}_{\mathrm{N,e}}$) = (5.15) and $F \equiv 0$. For this setting, it is not reasonable to expect that Assumption 5.9(ii) holds. Consequently, the argument used in the proof of Theorem 5.13 does not apply; in fact, we are not aware of results in the literature analyzing such schemes with the techniques from Chap. 5.

Fortunately, the stability analysis in Chap. 6 provides a remedy to this problem. Observe that the main structural assumption on the control sequences from Assumption 6.4 needed in the fundamental Lemmas 6.9 and 6.10 in Chap. 6 is that each admissible control sequence $u \in \mathbb{U}^N(x)$ can be extended to an admissible control

L. Grüne, J. Pannek, *Nonlinear Model Predictive Control*,
Communications and Control Engineering,
DOI 10.1007/978-0-85729-501-9_7, © Springer-Verlag London Limited 2011

sequence $\hat{u} \in \mathbb{U}^{N+K}(x)$ for each $K \geq 1$. Since Lemma 5.2(i) ensures this property for $\mathbb{U}^N_{\mathbb{X}_0}(x)$ provided $\mathbb{X}_0$ is viable, we can incorporate the terminal constraint set $\mathbb{X}_0$ into the analysis from Chap. 6.

As a consequence, replacing $\mathbb{U}^N(x)$ by $\mathbb{U}^N_{\mathbb{X}_0}(x)$ in Assumption 6.4 and assuming Assumption 5.9(i), i.e., viability of $\mathbb{X}_0$, all results in Chap. 6 carry over to the scheme with terminal constraint set. In particular, the stability results Theorem 6.18, Corollary 6.19, Theorem 6.21 and Theorem 6.33 remain valid. However, like in Theorem 5.13 the resulting controller μ_N is only defined on the feasible set $\mathbb{X}_N$ from Definition 3.9.

This combined scheme inherits certain advantages and disadvantages from both schemes. From the terminal constrained scheme we inherit that the resulting controller μ_N is only defined on the feasible set $\mathbb{X}_N$. On the other hand, as discussed before Lemma 5.3, we do not need to assume viability of $\mathbb{X}$ but only for the terminal constraint set $\mathbb{X}_0$ (further methods to avoid the viability assumption on $\mathbb{X}$ will be discussed in Sects. 8.1–8.3).

From the unconstrained scheme we inherit the advantage that no terminal cost satisfying Assumption 5.9(ii) needs to be constructed. On the other hand, we need to ensure that the assumptions of one of the mentioned stability results from Chap. 6 hold whose rigorous verification may be involved, cf. also Sect. 6.6. For a more comprehensive discussion on advantages and disadvantages of different NMPC schemes we refer to Sect. 8.4.

Another way of imposing terminal constraints without terminal costs which can be found in the literature is via so-called *contractive constraints*. Here the terminal constraint set depends on the initial value x_0 of the optimal control problem ($\mathrm{OCP_{N,e}}$) via

$$\mathbb{X}_0 = \left\{ x \in \mathbb{X} \,\middle|\, |x|_{x_*} \leq \gamma |x_0|_{x_*} \right\}$$

for some constant $\gamma \in (0,1)$; see, e.g., the book of Alamir [1] or the works of de Oliveira Kothare and Morari [28] and De Nicolao, Magni and Scattolini [5]. However, for these constraints stability is only guaranteed if either the whole optimal control sequence (as opposed to only the first element) is applied or if the optimization horizon is treated as an optimization variable and the contractivity condition is incorporated into the optimization objective [1, Chap. 4]. Since these approaches do not conform with the MPC paradigm used throughout this book, we do not discuss their analysis in detail.

Schemes with terminal cost F but without terminal constraint $\mathbb{X}_0$ have been investigated in several places in the literature, for instance in Grimm, Messina, Tuna and Teel [13] and Jadbabaie and Hauser [22] (for more information on these references see also the discussions at the end of Sect. 6.1 and in Sect. 6.9). In both references stability results for such schemes are derived in which only positive definiteness of F is assumed. Roughly speaking, these references show that the addition of F does not destroy stability. While the authors emphasize the potential positive effects of adding such costs, they do not rigorously analyze these positive effects. In contrast to this, in the work of Parisini and Zoppoli [30] the specific properties of the terminal cost described in Remark 5.15 were exploited in order to show stability.

The proof in [30] uses that under suitable conditions and for sufficiently large optimization horizon N for all initial values from a given region the open-loop optimal trajectories end up in the terminal constraint set without actually imposing this as a condition. The same proof idea has been generalized later by Limón, Alamo, Salas and Camacho [24] for a more general terminal cost.

Here we outline an approach from Grüne and Rantzer [17] which we combine with the analysis technique from Chap. 6. This approach rigorously shows the positive effect of adding a terminal cost also in the absence of stabilizing terminal constraints. In contrast to [30] or [24] the stability property is not restricted to sets of initial values for which the open-loop optimal trajectories end up in a terminal constraint set. However, the fact that this happens for a set of initial values around the origin will be used in our proof. We start from a terminal cost function F satisfying Assumption 5.9(ii) with a forward invariant neighborhood $\mathbb{X}_0$ of x_*, however, we will not use $\mathbb{X}_0$ as a terminal constraint set. Instead, we assume that $F \equiv c > 0$ holds on the boundary $\partial\mathbb{X}_0$ with $c \geq \sup_{x\in\mathbb{X}_0} F(x)$. This is, for instance, satisfied if F is constructed from a linearization via linear–quadratic techniques according to Remark 5.15 and $\mathbb{X}_0$ is a sublevel set of F. Then we may extend F continuously to the whole set $\mathbb{X}$ by setting $F(x) := c$ for all $x \in \mathbb{X} \setminus \mathbb{X}_0$.

With this setting we obtain the following theorem.

Theorem 7.1 *Let the assumptions of Theorem* 6.33 *be satisfied for the NMPC Algorithm* 3.1 *without terminal cost. Let* $F : \mathbb{X} \to \mathbb{R}_0^+$ *and assume that Assumption* 5.9 *holds for some set* $\mathbb{X}_0$ *containing a ball* $\mathcal{B}_\eta(x_*)$ *for some* $\eta > 0$. *Assume, furthermore, that* $F \equiv c$ *holds outside* $\mathbb{X}_0$ *with* $c \geq \sup_{x\in\mathbb{X}_0} F(x)$ *and that* $F(x) \leq \tilde{\alpha}_2(|x|_{x_*})$ *holds for all* $x \in \mathbb{X}_0$ *and some* $\tilde{\alpha}_2 \in \mathcal{K}_\infty$. *Consider the NMPC Algorithm* 3.10 *with* $(\mathrm{OCP_{N,e}}) = (5.15)$ *for this* F *but without terminal constraints, i.e., with* $\mathbb{X}_0 = \mathbb{X}$ *in* (5.15).

Then the nominal NMPC closed-loop system (3.5) *with NMPC feedback law* μ_N *is semiglobally asymptotically stable on* $\mathbb{X}$ *with respect to the parameter* N *in the sense of Definition* 6.28(i).

Proof We consider the following three optimal control problems

(a) (5.15) with $\mathbb{X}_0 = \mathbb{X}$, which generates μ_N in this theorem
(b) (5.15) with $\mathbb{X}_0$ from Assumption 5.9 for F, which generates μ_N in Theorem 5.5
(c) $(\mathrm{OCP_N})$, which generates μ_N in Theorem 6.18

and denote the respective optimal value functions by $V_N^{(a)}$, $V_N^{(b)}$ and $V_N^{(c)}$. For each $x \in \mathbb{X}$ we obtain the inequalities $V_N^{(c)}(x) \leq V_N^{(a)}(x) \leq V_N^{(c)}(x) + c$ and, for $x \in \mathbb{X}_N$ (where $\mathbb{X}_N$ denotes the feasible set from Definition 3.9 for Problem (b)), we have $V_N^{(a)}(x) \leq V_N^{(b)}(x)$.

In order to show semiglobal asymptotic stability, i.e., Definition 6.28(i), we fix $\Delta > 0$. For an arbitrary $x \in \mathbb{X}$ we consider the optimal control $u^\star$ for Problem (a) (which implies $\mu_N(x) = u^\star(0)$ for μ_N from this theorem) and distinguish two cases:

(i) $x_{u^\star}(N,x) \in \mathbb{X}_0$: This implies $u^\star \in \mathbb{U}^N_{\mathbb{X}_0}(x)$ and hence $x \in \mathbb{X}_N$ and $V_N^{(a)}(x) = V_N^{(b)}(x)$. Using $x_{u^\star}(1,x) = f(x,\mu_N(x)) \in \mathbb{X}_N$ and $V_N^{(a)} \le V_N^{(b)}$ on $\mathbb{X}_N$, the proof of Theorem 5.5 yields

$$\begin{aligned} V_N^{(a)}(x) = V_N^{(b)}(x) &\ge \ell\bigl(x,\mu_N(x)\bigr) + V_N^{(b)}\bigl(f\bigl(x,\mu_N(x)\bigr)\bigr) \\ &\ge \ell\bigl(x,\mu_N(x)\bigr) + V_N^{(a)}\bigl(f\bigl(x,\mu_N(x)\bigr)\bigr). \end{aligned} \tag{7.1}$$

This inequality will be used below in order to conclude asymptotic stability. Before we turn to case (ii) we show that case (i) applies to all points $x \in \mathcal{B}_\delta(x_*)$ for some $\delta > 0$:

Since (5.20) shows $V_N^{(b)}(x) \le F(x)$ on $\mathbb{X}_0$, we obtain $V_N^{(a)}(x) \le V_N^{(b)}(x) \le \tilde{\alpha}_2(|x|_{x_*})$ for $x \in \mathcal{B}_\eta(x_*) \subseteq \mathbb{X}_0$. For $\delta = \min\{\eta, \tilde{\alpha}_2^{-1}(c/2)\}$ this implies $V_N^{(a)}(x) \le c/2$ for all $x \in \mathcal{B}_\delta(x_*)$. On the other hand, $x_{u^\star}(N,x) \notin \mathbb{X}_0$ implies $F(x_{u^\star}(N,x)) = c$ and thus $V_N^{(a)}(x) \ge c$. Hence, case (i) occurs for all $x \in \mathcal{B}_\delta(x_*)$.

(ii) $x_{u^\star}(N,x) \notin \mathbb{X}_0$: This implies $F(x_{u^\star}(N,x)) = c$ and thus $V_N^{(a)}(x) = V_N^{(c)}(x) + c$. This implies that $u^\star$ is an optimal control for $V_N^{(c)}(x)$ and from the proof of Theorem 6.33 we obtain that (5.1), i.e.,

$$V_N^{(c)}(x) \ge \alpha\ell\bigl(x,\mu_N(x)\bigr) + V_N^{(c)}\bigl(f\bigl(x,\mu_N(x)\bigr)\bigr)$$

holds for all $x \in Y = S \setminus P$ with S and P chosen as in the proof of Theorem 6.33. The sets S and P are forward invariant and by choosing $N \in \mathbb{N}$ sufficiently large we obtain $\alpha > 0$, $\overline{\mathcal{B}}_\Delta(x_*) \subseteq S$ and $P \subset \mathcal{B}_\delta(x_*)$ for Δ fixed above and δ defined at the end of case (i). Since $V_N^{(a)}(x) = V_N^{(c)}(x) + c$ and $V_N^{(a)}(f(x,\mu_N(x))) \le V_N^{(c)}(f(x,\mu_N(x))) + c$ we obtain

$$V_N^{(a)}(x) \ge \alpha\ell\bigl(x,\mu_N(x)\bigr) + V_N^{(a)}\bigl(f\bigl(x,\mu_N(x)\bigr)\bigr) \tag{7.2}$$

for all $y \in Y$ and some $\alpha > 0$.

Now, the choice of N and P implies that for $x \in S \setminus \mathcal{B}_\delta(x_*)$ Inequality (7.2) holds while for $x \in \mathcal{B}_\delta(x_*)$ Inequality (7.1) holds. This implies that Theorem 4.14 is applicable with $S(n) = S$ which yields semiglobal practical stability using Lemma 6.29(i). □

Comparing Theorem 7.1 with Theorem 6.33, one sees that the benefit of including the terminal cost F is that here we obtain semiglobal asymptotic stability while without F we can only guarantee semiglobal *practical* asymptotic stability. Loosely speaking, the unconstrained scheme guarantees stability up to the neighborhood $\mathcal{B}_\delta(x_*)$, while F ensures asymptotic stability inside this neighborhood.

7.2 Unconstrained NMPC with Terminal Weights

Our next extension analyzes the effect of inclusion of terminal weights in ($\mathrm{OCP_N}$), i.e., in NMPC schemes without stabilizing terminal constraints and costs. Both in numerical simulations and in practice one can observe that adding terminal weights can improve the stability behavior of the NMPC closed loop. Formally, adding terminal weights can be achieved by replacing the optimization criterion in ($\mathrm{OCP_N}$) by

$$J_N\big(x_0, u(\cdot)\big) := \sum_{k=0}^{N-2} \ell\big(x_u(k, x_0), u(k)\big) + \omega\ell\big(x_u(N-1, x_0), u(N-1)\big) \quad (7.3)$$

for some $\omega \geq 1$. For $\omega = 1$ we thus obtain the original problem ($\mathrm{OCP_N}$). This extension is a special case of ($\mathrm{OCP_{N,e}}$) in which we specify $\mathbb{X}_0 = \mathbb{X}$, $F \equiv 0$, $\omega_1 = \omega$ and $\omega_2 = \omega_3 = \cdots = \omega_N = 1$. In a similar way, such a terminal weight can be added to the respective time variant problem ($\mathrm{OCP_N^n}$) leading to a special case of ($\mathrm{OCP_{N,e}^n}$). Thus, all results developed in Chap. 3 apply to this problem. Given that the optimal control value $u(N-1)$ in (7.3) will minimize $\ell(x_u(N-1, x_0), u(N-1))$, this approach is identical to choosing $F(x) = \omega\ell^*(x)$ and $N = N-1$ in the terminal cost approach discussed in the previous section, with ℓ^* from (6.2). However, the specific structure of the terminal cost allows for applying different and more powerful analysis techniques which we explain now.

The terminal weight leads to an increased penalization of $\ell(x_u(N-1, x_0), u(N-1))$ in J_N and thus to an increased penalization of the distance of $x_u(N-1, x_0)$ to x_*. Thus, for $\omega > 1$ the optimizer selects a finite time optimal trajectory whose terminal state $x_{u^\star}(N-1, x_0)$ has a smaller distance to x_*. Since our goal is that the NMPC-feedback law μ_N steers the trajectory to x_*, this would intuitively explain better stability behavior.

Formally, however, the analysis is not that easy because in closed loop we never actually apply $u^\star(1), \ldots, u^\star(N-1)$ and the effect of ω on $u^\star(0)$ is not that obvious. Hence, we extend the technique developed in Chap. 6 in order to analyze the effect of ω. To this end, we change the definition (6.8) of B_N to

$$B_N(r) := \sum_{n=0}^{N-2} \beta(r, n) + \omega\beta(r, N-1).$$

With this definition, all results in Sect. 6.3 remain valid for the extended problem. Proposition 6.12 remains valid, too, if we change (6.11) to

$$\sum_{n=k}^{N-2} \lambda_n + \omega\lambda_{N-1} \leq B_{N-k}(\lambda_k), \quad k = 0, \ldots, N-2.$$

If, furthermore, in the subsequent statements we replace $\sum_{n=0}^{N-1} \lambda_n$ by $\sum_{n=0}^{N-2} \lambda_n + \omega\lambda_{N-1}$, then it can be shown that Proposition 6.17 remains valid if we replace (6.19) by

$$\underline{\alpha}_N^\omega := 1 - \frac{(\gamma_N - 1)(\gamma_2 - \omega)\prod_{i=3}^N(\gamma_i - 1)}{\prod_{i=2}^N \gamma_i - (\gamma_2 - \omega)\prod_{i=3}^N(\gamma_i - 1)}. \quad (7.4)$$

The proof is similar to the proof of Proposition 6.17 and can be found in Grüne, Pannek, Seehafer and Worthmann [20].

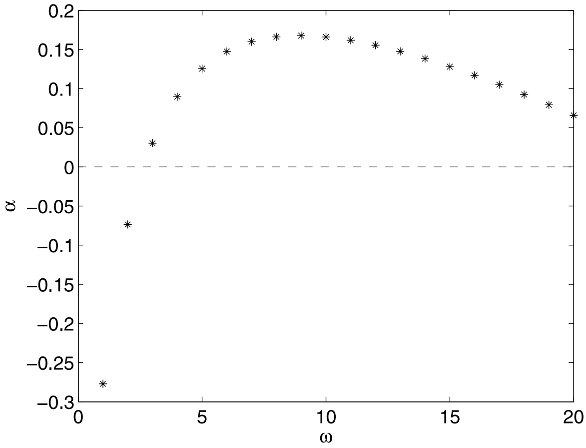


Fig. 7.1 Suboptimality index α depending on terminal weight ω

With this expression, Theorem 6.18 and its corollaries remain valid, except for the inequalities $V_N(x)/\alpha \le V_\infty(x)/\alpha$ and $CV_N(x) \le CV_\infty(x)$, which do in general no longer hold because of the additional weight which is present in V_N but not in V_∞.

Figure 7.1 shows the values from (7.4) for an exponential β of type (6.3) with $C = 2$ and $\sigma = 0.55$, optimization horizon $N = 5$ and terminal weights $\omega = 1, 2, \ldots, 20$. The figure illustrates that our analysis reflects the positive effect the terminal weight has on the stability: while for $\omega = 1, 2$ we obtain negative values for α and thus stability cannot be ensured, for $\omega \ge 3$ stability is guaranteed. However, one also sees that for $\omega \ge 10$ the value of α is decreasing, again. For more examples for the effect of terminal weights we refer to [20] and Example 7.14, below.

7.3 Nonpositive Definite Running Cost

In many regulator problems one is not interested in driving the whole state to a reference trajectory or point. Rather, often one is only interested in certain output quantities. The following example illustrates such a situation.

Example 7.2 We reconsider Example 2.2, i.e.,

$$\begin{pmatrix} x_1^+ \\ x_2^+ \end{pmatrix} = \begin{pmatrix} x_1 + x_2 + u/2 \\ x_2 + u \end{pmatrix} =: f(x,u)$$

with running cost

$$\ell(x,u) = x_1^2 + u^2.$$

In contrast to our standing assumption (3.2), no matter how we choose $x_* \in \mathbb{R}^2$, this function does not satisfy $\ell(x,u) > 0$ for all $x \in X$ and $u \in U$ with $x \neq x_*$. Instead,

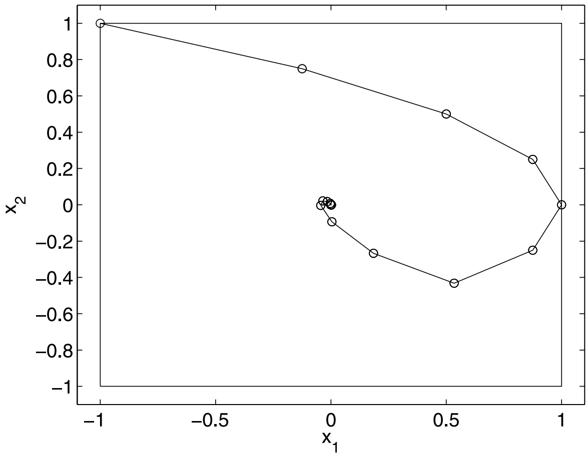


Fig. 7.2 MPC closed-loop trajectory with $N = 5$

following the interpretation of x_1 and x_2 as position and velocity of a vehicle in a plane, the running cost only penalizes the distance of the position x_1 from 0 but not the velocity.

However, the only way to put the system at rest with $x_1 = 0$ is to set $x_2 = 0$. Hence, one may expect that the NMPC controller will "automatically" steer x_2 to 0, too. The numerical simulation shown in Fig. 7.2 (performed with optimization horizon $N = 5$ without stabilizing terminal constraints and with state constraints $\mathbb{X} = [-1, 1]^2$ and control constraints $\mathbb{U} = [-1/4, 1/4]$) confirms that this is exactly what happens: the system is perfectly stabilized at $x_* = 0$ even though the running cost does not "tell" the optimization problem to steer x_2 to 0.

How can this behavior be explained theoretically? The decisive difference of ℓ from this example to ℓ used in the theorems in the previous chapters is that the lower bound $\ell(x, u) \geq \alpha_3(|x|_{x_*})$ imposed in all our results is no longer valid. In other words, the running cost is no longer positive definite.

For NMPC schemes with stabilizing terminal constraints and costs satisfying Assumption 5.9, the notion of input/output-to-state stability (IOSS) provides a way to deal with this setting. IOSS can be seen as a nonlinear detectability condition which ensures that the state converges to x_* if both the output and the input converge to their steady state values, which can in turn be guaranteed by suitable bounds on ℓ. We sketch this approach for time invariant reference $x^{\text{ref}} \equiv x_*$ with corresponding control value u_* satisfying $f(x_*, u_*) = x_*$.

To this end, we relax the assumptions of Theorem 5.13 as follows: instead of assuming (5.2) we consider an output function $h : X \to Y$ for another metric space Y. In Example 7.2 we have $X = \mathbb{R}^2$, $Y = \mathbb{R}$ and $h(x) = x_1$.

Now we change (5.2) to

$$\begin{aligned} &\alpha_1\big(\big|h(x)\big|_{y_*}\big) \leq V_N(x) \leq \alpha_2\big(|x|_{x_*}\big) \quad \text{and} \\ &\ell(x, u) \geq \alpha_3\big(\big|h(x)\big|_{y_*}\big) + \alpha_3\big(|u|_{u_*}\big) \end{aligned} \tag{7.5}$$

with $y_* = h(x_*)$ and $|h(x)|_{y_*} = d_Y(h(x), y_*)$, where $d_Y(\cdot,\cdot)$ is the metric on Y. Furthermore, we assume that the system with output $y = h(x)$ is IOSS in the following sense: There exist $\beta \in \mathcal{KL}$ and $\gamma_1, \gamma_2 \in \mathcal{K}_\infty$ such that for each $x \in \mathbb{X}$ and each admissible control $u \in \mathbb{U}^\infty(x)$ the inequality

$$\big|x_u(n,x)\big|_{x_*} \le \max\Big\{\beta\big(|x|_{x_*}, n\big), \gamma_1\Big(\max_{k=0,\dots,n-1}\big|u(k)\big|_{u_*}\Big), \gamma_2\Big(\max_{k=0,\dots,n-1}\big|y(k)\big|_{y_*}\Big)\Big\}$$

holds for all $n \in \mathbb{N}_0$ with $y(k) = h(x_u(k,x))$.

With these changed assumptions, the assertion of Theorem 5.13 remains valid. The proof relies on the fact that the function V_N still satisfies

$$V_N(x) \ge \ell\big(x, \mu_N(x)\big) + V_N\big(f\big(x, \mu_N(x)\big)\big).$$

This implies that $V_N(x_{\mu_N}(n,x))$ is monotone decreasing in n and since it is bounded from below by 0 it converges to some value as $n \to \infty$, although not necessarily to 0. However, the convergence of $V_N(x_{\mu_N}(n,x))$ implies convergence of $\ell(x_{\mu_N}(n,x), \mu_N(x_{\mu_N}(n,x))) \to 0$ which by means of the last inequality in (7.5) yields $h(x_{\mu_N}(n,x)) \to 0$ and $\mu_N(x_{\mu_N}(n,x)) \to 0$. Now the IOSS property can be used to conclude asymptotic stability of the closed loop. For more details of this approach, we refer to the book of Rawlings and Mayne [31, Sect. 2.7 and the references therein].

While the approach just sketched relies on stabilizing terminal constraints, the simulation in Example 7.2 shows that asymptotic stability can also be expected without such constraints. For this setting, a stability proof was given in the work of Grimm, Messina, Tuna and Teel [13] and the main result in this reference extends Theorem 6.33. Again, a detectability condition is used, but this time it is formulated via a suitable auxiliary function W: we assume the existence of a function $W : X \to \mathbb{R}_0^+$ which satisfies the inequalities

$$\begin{aligned} W(x) &\le \overline{\alpha}_W\big(|x|_{x_*}\big),\\ W\big(f(x,u)\big) - W(x) &\le -\alpha_W\big(|x|_{x_*}\big) + \gamma_W\big(\ell(x,u)\big) \end{aligned} \tag{7.6}$$

for all $x \in \mathbb{X}$, $u \in \mathbb{U}(x)$ and suitable functions $\overline{\alpha}_W, \alpha_W, \gamma_W \in \mathcal{K}_\infty$. In turn, we remove the lower bound $\alpha_3(|x|_{x_*}) \le \ell^*(x)$ for ℓ^* from (6.2) from the assumptions of Theorem 6.33. Observe that whenever this lower bound holds, the detectability condition is trivially satisfied with $W \equiv 0$, $\gamma_W(r) = r$ and $\alpha_W = \alpha_3$.

Under these modified assumptions, it is shown in [13, Theorem 1] that the semiglobal practical stability assertion of Theorem 6.33 remains valid. Furthermore, [13, Corollary 2 and Corollary 3] provide counterparts to Theorems 6.31 and 6.21 which prove semiglobal and "real" asymptotic stability, respectively. In contrast to the IOSS-based result for stabilizing terminal constraints, the proof of [13, Theorem 1] yields a Lyapunov function constructed from the optimal value function V_N and the function W from the detectability condition. In the simplest case, which occurs under suitable bounds on the involved $\mathcal{K}_\infty$-functions, this Lyapunov function is given by $V_N + W$. In general, a weighted sum has to be used.

In Example 7.2, numerical evaluation suggests that the detectability condition is satisfied for $W(x) = \max\{-|x_1 x_2| + x_2^2, 0\}/2$ and $\gamma_W(r) = r$. Plots of the difference $W(x) - W(f(x,u)) + \ell(x,u)$ in MAPLE indicate that this expression is positive definite and can hence be bounded from below by some function $\alpha_W(|x|_{x_*})$; a rigorous proof of this property is, however, missing up to now.

As discussed in Sect. 6.9, the analysis in [13] uses a condition of the form $V_N(x) \le \alpha_V(r)$ in order to show stability, which compared to our Assumptions 6.4 or 6.30 has the drawback to yield fewer information for the design of "good" running costs ℓ. Furthermore, suboptimality estimates are not easily available. It would hence be desirable to extend the statement and proof of Theorem 6.18 to the case of nonpositive definite running costs. A first attempt in this direction is the following: suppose that we are able to find a function $W: X \to \mathbb{R}_0^+$ satisfying (7.6) with $\gamma_W(r) = r$. Then the function

$$\ell_W(x,u) := W(x) - W\big(f(x,u)\big) + \ell(x,u)$$

satisfies a lower bound of the form

$$\ell_W^*(x) := \min_{u \in U} \ell_W(x,u) \ge \alpha_W\big(|x|_{x_*}\big)$$

for all $x \in X$. Let $u^\star$ be an optimal control for $V_N(x)$, i.e.,

$$V_N(x) = J_N\big(x, u^\star\big) = \sum_{k=0}^{N-1} \ell\big(x_{u^\star}(k,x), u^\star(k)\big)$$

and define

$$\widetilde{V}_N(x) := \sum_{k=0}^{N-1} \ell_W\big(x_{u^\star}(k,x), u^\star(k)\big).$$

The definition of ℓ_W then implies

$$\widetilde{V}_N(x) = W(x) - W\big(x_{u^\star}(N,x)\big) + V_N(x).$$

Changing the inequality in Assumption 6.30 to

$$V_k(x) \le B_k\big(\ell_W^*(x)\big) - W(x)$$

then implies

$$\widetilde{V}_k(x) \le B_k\big(\ell_W^*(x)\big).$$

Using this inequality, it should be possible to carry over all results in Sect. 6.3 to $\widetilde{V}_N$ using ℓ_W in place of ℓ. A rigorous investigation of this approach as well as possible extensions will be the topic of future research.

In this context we would like to emphasize once again that even if the running cost ℓ only depends on an output y, the resulting NMPC-feedback law is still a state feedback law because the full state information is needed in order to compute the prediction $x_u(\cdot, x_0)$ for $x_0 = x(n)$.

7.4 Multistep NMPC-Feedback Laws

Next we investigate what happens if instead of only the first control value $u^\star(0)$ we implement the first m values $u^\star(0), \dots, u^\star(m-1)$ before optimizing again. Formally, we can write this NMPC variant as a multistep feedback law

$$\mu_N(x,k) := u^\star(k), \quad k = 0, \dots, m-1,$$

where $u^\star$ is an optimal control sequence for problem $(\mathrm{OCP}_{\mathrm{N,e}})$ (or one of its variants) with initial value $x_0 = x$. The resulting generalized closed-loop system then reads

$$x(n+1) = f\big(x(n), \mu_N\big(x\big([n]_m\big), n - [n]_m\big)\big), \tag{7.7}$$

where $[n]_m$ denotes the largest product km, $k \in \mathbb{N}_0$, with $km \le n$. The value $m \in \{1, \dots, N-1\}$ is called the *control horizon*.

When using stabilizing terminal constraints, the respective stability proofs from Chap. 5 are easily extended to this setting which we illustrate for Theorem 5.13. Indeed, from $V_N(x) \le V_{N-1}(x)$ one immediately gets the inequality $V_N(x) \le V_{N-m}(x)$ for each $m \in \{1, \dots, N-1\}$ and each $x \in \mathbb{X}_{N-m}$. Proceeding as in the proof of Theorem 5.13 using Equality (3.20) inductively for $N, N-1, \dots, N-m+1$ and $V_N(x) \le V_{N-m}(x)$ one obtains

$$V_N(x) \ge \sum_{k=0}^{m-1} \ell\big(x_{\mu_N}(k,x), \mu_N(k,x)\big) + V_N\big(x_{\mu_N}(m,x)\big).$$

This shows that V_N is a Lyapunov function for the closed-loop system at the times $0, m, 2m, \dots$. Since a similar argument shows that $V_N(x_{\mu_N}(k,x))$ is bounded by $V_N(x)$ for $k = 1, \dots, m-1$, this proves asymptotic stability of the closed loop.

Without stabilizing terminal constraints, our analysis can be adjusted to the multistep setting, too, by extending Proposition 6.17 as well as the subsequent stability results, accordingly. The respective extension of Formulas (6.19) and (7.4) (including both control horizons $m \ge 1$ and terminal weights $\omega \ge 1$) is given by

$$\underline{\alpha}^{\omega}_{N,m} = 1 - \frac{(\gamma_{m+1}-\omega)\prod_{i=m+2}^{N}(\gamma_i-1)\prod_{i=N-m+1}^{N}(\gamma_i-1)}{(\prod_{i=m+1}^{N}\gamma_i-(\gamma_{m+1}-\omega)\prod_{i=m+2}^{N}(\gamma_i-1))(\prod_{i=N-m+1}^{N}\gamma_i-\prod_{i=N-m+1}^{N}(\gamma_i-1))}.$$

Again, the proof proceeds along the lines of the proof of Proposition 6.17 but becomes considerably more involved, cf. the paper by Grüne, Pannek, Seehafer and Worthmann [20].

It is worth noting that these extended stability and performance results remain valid if m is time varying, i.e., if the control horizon is changed dynamically, e.g., by a network induced perturbation. This has interesting applications in NMPC for networked control systems, cf. the work of Grüne, Pannek and Worthmann [18].

Figure 7.3 shows how $\alpha = \underline{\alpha}^{\omega}_{N,m}$ depends on m for an exponential β of type (6.3) with $C = 2$ and $\sigma = 0.75$, optimization horizon $N = 11$, terminal weight $\omega = 1$ and control horizons $m = 1, \dots, 10$. Here one observes two facts: first, the α-values are symmetric, i.e., $\underline{\alpha}^{\omega}_{N,m} = \underline{\alpha}^{\omega}_{N,N-m}$ and second, the values increase until $m = (N-1)/2$ and then decrease, again. This is not a particular feature of this

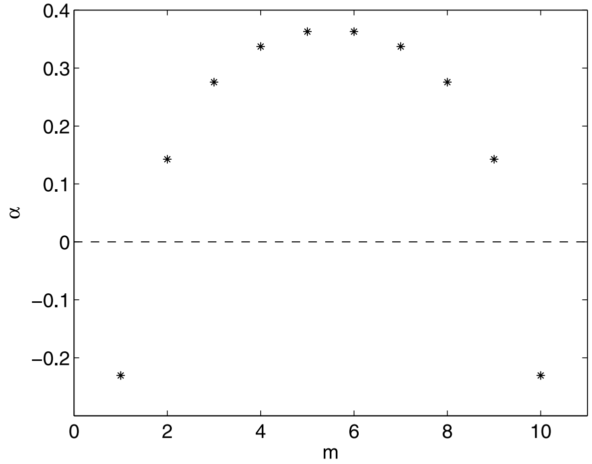


Fig. 7.3 Suboptimality index α depending on control horizon m

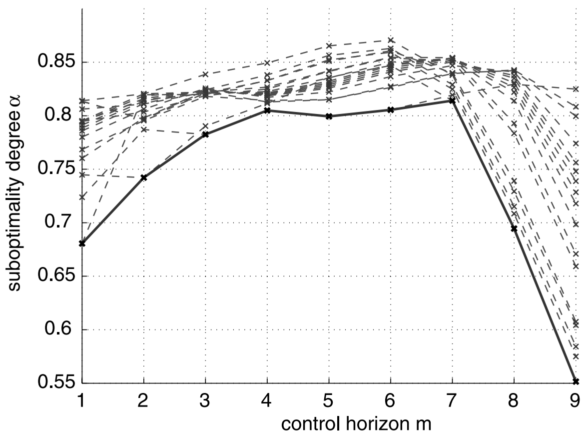


Fig. 7.4 Numerically measured values for α for a linear inverted pendulum and various initial values. The *thick line* represents the minimum

example. In fact, it can be rigorously proved for a general class of $\beta \in \mathcal{KL}_0$; see [20] for details.

It is interesting to compare Fig. 7.3 with α-values which have been obtained numerically from an NMPC simulation for the linear inverted pendulum, cf. Example 2.10 and Sect. A.2 or [18] for the precise description of the problem. Figure 7.4 shows the resulting values for a set of different initial values. These values have been computed by Algorithm 7.8 described in Sect. 7.7, below.

While the monotonicity is—at least approximately—visible in this example, the perfect symmetry from Fig. 7.3 is not reflected in Fig. 7.4. A qualitatively similar behavior can be observed for the nonlinear inverted pendulum; see Example 7.14, below. In fact, so far we have not been able to find an example for which the symmetry could be observed in simulations. This may be due to the fact that our stability estimate is tight not for a single system but rather for the whole class of systems satisfying Assumption 6.4, cf. Theorem 6.23. Our numerical findings suggest that the conservativity induced by this "worst case approach" is higher for small m than for large m. This is also supported by Monte Carlo simulations performed by Grüne in [14].

7.5 Fast Sampling

Let us now turn to the special case of sampled data systems. In this case, according to (2.12) the discrete time solution $x_u(n, x_0)$ represents the continuous time solution $\varphi(t, 0, x_0, v)$ at sampling times $t = nT$. In this setting, it is natural to define the optimization horizon not in terms of the discrete time variable n but in terms of the continuous time t. Fixing an optimization horizon $T_{\mathrm{opt}} > 0$ and picking a sampling period $T > 0$ where we assume for simplicity of exposition that T_{opt} is an integer multiple of T, the discrete time optimization horizon becomes $N = T_{\mathrm{opt}}/T$, cf. also Sect. 3.5.

Having introduced this notation, an interesting question is what happens to stability and performance of the NMPC closed loop if we keep T_{opt} fixed but vary the sampling period T. In particular, it is interesting to see what happens if we sample faster and faster, i.e., if we let $T \to 0$. Clearly, in a practical NMPC implementation we cannot arbitrarily reduce T because we need some time for solving the optimal control problem ($\mathrm{OCP_N}$) or its variants online. Still, in particular in the case of zero order hold it is often desirable to sample as fast as possible in order to approximate the ideal continuous time control signal as good as possible, cf., e.g., the paper of Nešić and Teel [26], and thus one would like to make sure that this does not have negative effects on the stability and performance of the closed loop.

In the case of equilibrium endpoint constraint from Sect. 5.2 it is immediately clear that the stability result itself does not depend on T, however, the feasible set $\mathbb{X}_N$ may change with T. In the case of zero order hold, i.e., when the continuous time control function v is constant on each sampling interval $[nT, (n+1)T)$, cf. the discussion after Theorem 2.7, it is easily seen that each trajectory for sampling period T is also a trajectory for each sampling period T/k for each $k \in \mathbb{N}$. Hence, the feasible set $\mathbb{X}_{kN}$ for sampling period T/k always contains the feasible set $\mathbb{X}_N$ for sampling period T, i.e., the feasible set cannot shrink for $k \to \infty$ and hence for sampling period T/k we obtain at least the same stability properties as for sampling period T.

In the case of Lyapunov function terminal costs F as discussed in Sect. 5.3 either the terminal costs or the running costs need to be adjusted to the sampling period T in order to ensure that Assumption 5.9 remains valid. One way to achieve this is to choose a running cost in integral form (3.4) and the terminal cost F such that the following condition holds: for each $x \in \mathbb{X}_0$ and some $T_0 > 0$ there exists a continuous time control v satisfying $\varphi(t, 0, x, v) \in \mathbb{X}_0$ and

$$V\bigl(\varphi(t, 0, x, v)\bigr) - V(x) \le -\int_0^t L\bigl(\varphi(\tau, 0, x, v), v(\tau)\bigr)\, d\tau \tag{7.8}$$

for all $t \in [0, T]$, cf. also Findeisen [9, Sect. 4.4.2]. Under this condition one easily checks that Assumption 5.9 holds for ℓ from (3.4) and all $T \le T_0$, provided the control function v in (7.8) is of the form $v|_{[nT,(n+1)T)}(t) = u(n)(t)$ for an admissible discrete time control sequence $u(\cdot)$ with $u(n) \in U$. If $U = L^\infty([0, T], \mathbb{R}^m)$ then this last condition is not a restriction but if we use some smaller space for U (as in the case of zero order hold, cf. the discussion after Theorem 2.7), then this may be more difficult to achieve; see also [9, Remark 4.7].

Since the schemes from Chap. 6 do not use stabilizing terminal constraints $\mathbb{X}_0$ and terminal costs F, the difficulties just discussed vanish. However, the price to pay for this simplification is that the analysis of the effect of small sampling periods which we present in the remainder of this section is somewhat more complicated.

Fixing T_{opt} and letting $T \to 0$ we obtain that $N = T_{\text{opt}}/T \to \infty$. Looking at Theorem 6.21, this is obviously a good feature, because this theorem states that the larger N becomes, the better the performance will be. However, we cannot directly apply this theorem because we have to take into account that β in the Controllability Assumption 6.4 will also depend on T.

In order to facilitate the analysis, let us assume that in our discrete time NMPC formulation we use a running cost ℓ that only takes the states $\varphi(nT, 0, x_0, v)$ at the sampling instants and the respective control values into account.[1] For the continuous time system, the controllability assumption can be formulated in discrete time. We denote the set of admissible continuous time control functions (in analogy to the discrete time notation) by $\mathbb{V}^\tau(x)$. More precisely, for the admissible discrete time control values $\mathbb{U}(x) \subseteq U \subseteq L^\infty([0, T], \mathbb{R}^m)$ (recall that these "values" are actually functions on $[0, T]$, cf. the discussion after Theorem 2.7) and any $\tau > 0$ we define

$$\begin{aligned}\mathbb{V}^\tau(x) := \Big\{ v \in L^\infty\big([0,\tau], \mathbb{R}^m\big) \;\Big|\; &\text{there exists } u \in \mathbb{U}^N(x) \text{ with } N \ge \tau/T + 1 \\ &\text{such that } u(n) = v|_{[nT,(n+1)T]}(\cdot + nT) \\ &\text{holds for all } n \in \mathbb{N}_0 \text{ with } nT < \tau \Big\}.\end{aligned}$$

Then, the respective assumption reads as follows.

Assumption 7.3 We assume that the continuous time system is asymptotically controllable with respect to ℓ with rate $\beta \in \mathcal{KL}_0$, i.e., for each $x \in \mathbb{X}$ and each $\tau > 0$ there exists an admissible control function $v_x \in \mathbb{V}^\tau(x)$ satisfying

$$\ell\big(\varphi(t, 0, x, v_x), v_x(t)\big) \le \beta\big(\ell^*(x), t\big)$$

for all $t \in [0, \tau]$.

For the discrete time system (2.8) satisfying (2.12) the Controllability Assumption 7.3 translates to the discrete time Assumption 6.4 as

$$\ell\big(x_{u_x}(n, x), u_x(n)\big) \le \beta\big(\ell^*(x), nT\big).$$

[1] Integral costs (3.4) can be treated, too, but this is somewhat more technical, cf. Grüne, von Lossow, Pannek and Worthmann [21, Sect. 4.2].

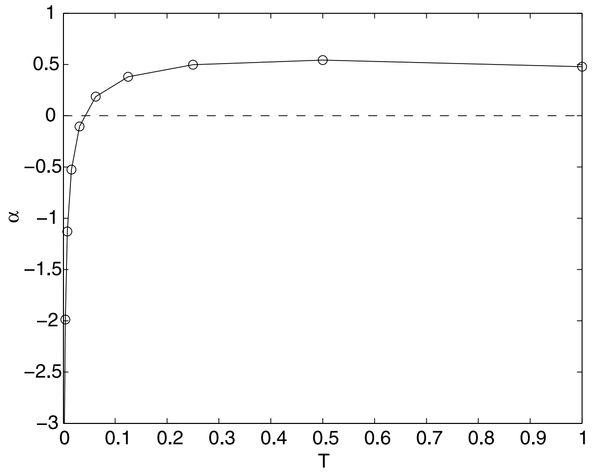


Fig. 7.5 Suboptimality index α from (6.19) for fixed T_{opt} and varying sampling period T

In the special case of exponential controllability, β in Assumption 7.3 is of the form

$$\beta(r,t) = Ce^{-\lambda t} r \tag{7.9}$$

for $C \geq 1$ and $\lambda > 0$. Thus, for the discrete time system, the Controllability Assumption 6.4 becomes

$$\ell\big(x_{u_x}(n,x), u_x(n)\big) \leq Ce^{-\lambda n T} \ell^*(x) = C\big(e^{-\lambda T}\big)^n \ell^*(x)$$

and we obtain a $\mathcal{KL}_0$-function of type (6.3) with C from (7.9) and $\sigma = e^{-\lambda T}$.

Summarizing, if we change the sampling period T, then not only the discrete time optimization horizon N but also the decay rate σ in the exponential controllability property will change, more precisely we have $\sigma \to 1$ as $T \to 0$. When evaluating (6.19) with the resulting values

$$\gamma_k = \sum_{j=0}^{k-1} Ce^{-\lambda j T},$$

it turns out that the convergence $\sigma \to 1$ counteracts the positive effect of the growing optimization horizons $N \to \infty$. In fact, the negative effect of $\sigma \to 1$ is so strong that α diverges to $-\infty$ as $T \to 0$. Figure 7.5 illustrates this fact (which can also be proven rigorously, cf. [21]) for $C = 2$, $\lambda = 1$ and $T_{\text{opt}} = 5$.

This means that whenever we choose the sampling period $T > 0$ too small, then performance may deteriorate and eventually instability may occur. This predicted behavior is not consistent with observations in numerical examples. How can this be explained?

The answer lies in the fact that our stability and performance estimate is only tight for one particular system in the class of systems satisfying Assumption 6.4, cf. Theorem 6.23 and the discussion preceding this theorem, and not for the whole class. In particular, the subclass of sampled data systems satisfying Assumption 6.4 may well behave better than general systems. Thus, we may try to identify the decisive

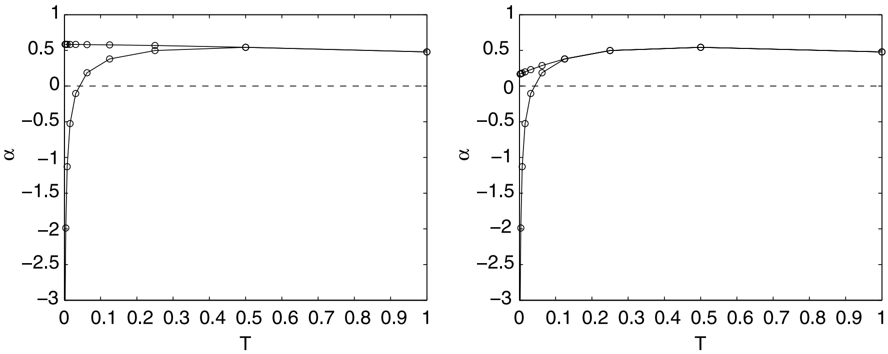


Fig. 7.6 α for fixed T_{opt} and varying sampling period T without Assumption 7.4 (*lower graphs*) and with Assumption 7.4 (*upper graphs*) with $L = 2$ (*left*) and $L = 10$ (*right*)

property which makes sampled data systems behave better and try to incorporate this property into our computation of α.

To this end, note that so far we have not imposed any continuity properties of f in (2.1). Sampled data systems, however, are governed by differential equations (2.6) for which we have assumed Lipschitz continuity in Assumption 2.4. Let us assume for simplicity of exposition that the Lipschitz constant in this assumption is independent of r. Then, for a large class of running costs ℓ the following property for the continuous time system can be concluded from Gronwall's Lemma; see [21] for details.

Assumption 7.4 There exists a constant $L > 0$ such that for each $x \in \mathbb{X}$ and each $\tau > 0$ there exists an admissible control function $v_x \in \mathbb{V}^\tau(x)$ satisfying

$$\ell\big(\varphi(t, 0, x, v_x), v_x(t)\big) \le e^{Lt} \ell^*(x)$$

for all $t \in [0, \tau]$.

The estimates on ℓ induced by this assumption can now be incorporated into the analysis in Chap. 6. As a result, the values γ_k in Formula (6.19) change to

$$\gamma_k = \min\left\{ \sum_{j=0}^{k-1} C e^{-\lambda j T}, \sum_{j=0}^{k-1} e^{LjT} \right\}.$$

The effect of this change is clearly visible in Fig. 7.6. The α-values from (6.19) no longer diverge to $-\infty$ but rather converge to a finite—and for the chosen parameters also positive—value as $T \to 0$. Again, this convergence behavior can be rigorously proved; for details we refer to [21].

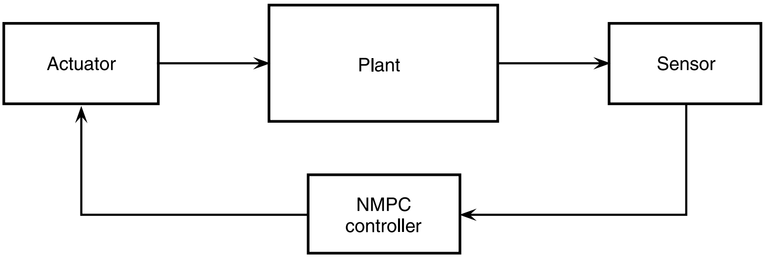


Fig. 7.7 Scheme of the NMPC closed-loop components

7.6 Compensation of Computation Times

Throughout the previous chapters we assumed that the solution of the optimal control problems ($\text{OCP}_{\text{N,e}}$) and its variants in Step (2) of Algorithms 3.1, 3.7, 3.10 and 3.11 can be obtained instantaneously, i.e., with negligible computation time. Clearly, this is not possible in general, as the algorithms for solving such problems, cf. Chap. 10 for details, need some time to compute a solution. If this time is large compared to the sampling period T, the computational delay caused by Step (2) is not negligible and needs to be considered. One way for handling these delays would be to interpret them as perturbations and use techniques similar to the robustness analysis in Sects. 8.5–8.9. In this section we pursue another idea in which a delay compensation mechanism is added to the NMPC scheme.

Taking a look at the structure of the NMPC algorithm from Chap. 3, we see that Steps (1)–(3) correspond to different physical tasks: measuring, computing and applying the control. These tasks are operated by individual components as shown schematically in Fig. 7.7. Note that in the following actuator, sensor and controller are not required to be physically decomposed, however, this case is also not excluded.

While it is a necessity to consider different clocks in a decomposed setting, it may not be the case if the components are physically connected. Here, we assume that every single component possesses its own clock and, for simplicity of exposition, that these clocks are synchronized (see the work of Varutti and Findeisen [34, Sect. III.C] for a possible way to relax this assumption). To indicate that a time instant n is considered with respect to a certain clock, we indicate this by adding indices s for the sensor, c for the NMPC controller and a for the actuator.

The idea behind the compensation approach is to run the NMPC controller component with a predefined time offset. This offset causes the controller to compute a control ahead of time, such that the computed control value is readily available at the time it is supposed to be applied, cf. Fig. 7.8. In this figure, τ_c denotes the actual computational delay and $\tau_c^{\max}$ denotes the predefined offset. In order to be operable, this offset needs to be chosen such that it is larger than the maximal computing time required to solve the optimal control problem in Step (2) of the considered NMPC algorithm. At time n_c this optimal control problem is solved with a prediction $\tilde{x}(n_a)$ of the initial value $x(n_a)$ based on the available measurement $x(n_c) = x(n_s)$. This

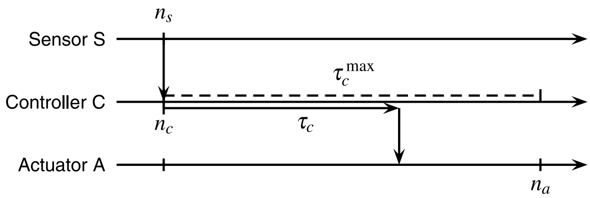


Fig. 7.8 Operation of time decoupled NMPC scheme

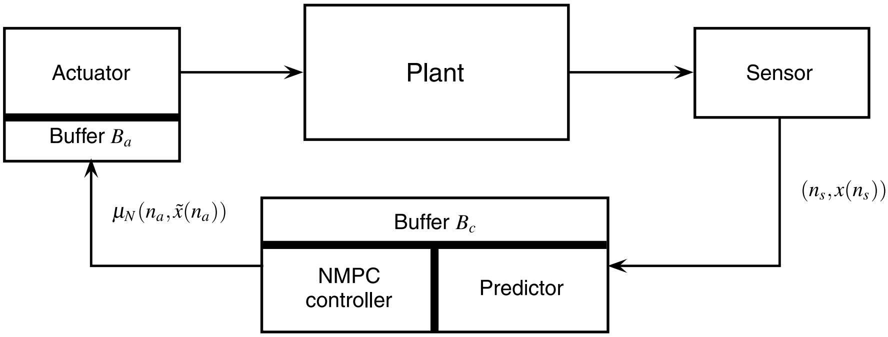


Fig. 7.9 Scheme of the time decoupled NMPC closed-loop components

prediction is performed using the same model which is used for the NMPC prediction in ($OCP_{N,e}$) or its variants, i.e., using (2.1).

In order to perform this prediction, the control values $\mu_N(n, x(n))$, $n \in \{n_s, \ldots, n_a\}$ which are to be applied at the plant during the time interval $[n_s, n_a]$ and which have been computed before by the NMPC controller are needed and are therefore buffered. Thus, we extend the scheme given in Fig. 7.7 by adding the required predictor to the controller. The structure of the resulting scheme is shown in Fig. 7.9.

Observe that in this scheme we buffer the control values twice: within the predictor, but also at the actuator since the computation of $\mu(n_a, x(n_a))$ will be finished ahead of time if $\tau_c < \tau_c^{\max}$, which is the typical case. Alternatively, one could use only one buffer at the controller and send each control value "just in time". Using two buffers has the advantage that further delays induced, e.g., by network delays between the controller and the actuator can be compensated; see also the discussion at the end of this section.

The corresponding algorithm has the following form. Since all NMPC algorithms stated in Chap. 3 can be modified in a similar manner, we only show the algorithm for the most general form given in Algorithm 3.11:

Algorithm 7.5 (Time decoupled NMPC algorithm for time varying reference) At each sampling time t_n, $n = 0, 1, 2, \ldots$:

(1) Measure the state $x(n_s) := x(n) \in X$ of the system and send pair $(n_s, x(n_s))$ to controller.

(2a) Delete pair $(n_c - 1, \mu_N(n_c - 1, x(n_c - 1)))$ from buffer B_c and compute the predicted state $\tilde{x}(n_c + \tau_c^{\max})$ from the measured state $x(n_c)$.

(2b) Set $\tilde{n} := n_c + \tau_c^{\max}$, $x_0 = \tilde{x}(\tilde{n})$ and solve the optimal control problem

$$\begin{array}{|l|}\hline \text{minimize} \quad J_N\big(\tilde{n}, x_0, u(\cdot)\big) := \sum_{k=0}^{N-1} \omega_{N-k}\ell\big(\tilde{n}+k, x_u(k, x_0), u(k)\big) \\ \qquad\qquad\qquad\qquad\qquad + F\big(\tilde{n} + N, x_u(N, x_0)\big) \\ \text{with respect to} \quad u(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(\tilde{n}, x_0), \quad \text{subject to} \\ x_u(0, x_0) = x_0, \qquad x_u(k+1, x_0) = f\big(x_u(k, x_0), u(k)\big) \\ \hline \end{array}$$

$$(\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N,e}})$$

and denote the obtained optimal control sequence by $u^\star(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(\tilde{n}, x_0)$.

(2c) Add pair $(\tilde{n}, \mu_N(\tilde{n}, \tilde{x}(\tilde{n}))) := (\tilde{n}, u^\star(0))$ to Buffer B_c and send it to actuator.

(3a) Delete pair $(n_a - \tau_c^{\max} - 1, \mu_N(n_a - \tau_c^{\max} - 1, \tilde{x}(n_a - \tau_c^{\max} - 1)))$ and add received pair $(n_a, \mu_N(n_a, \tilde{x}(n_a)))$ to buffer B_a.

(3b) Use $\mu_N(n_a - \tau_c^{\max}, \tilde{x}(n_a - \tau_c^{\max}))$ in the next sampling period.

At a first glance, writing this algorithm using three different clocks and sending time stamped information in Steps (1) and (2c) may be considered as overly complicated, given that n_s in Step (1) is always equal to n_c in Step (2a) and n_c in Step (2c) always equals n_a in Step (3a). However, this way of writing the algorithm allows us to easily separate the components—sensor, predictor/controller and actuator—of the NMPC scheme and to assume that the "sending" in Steps (1) and (2c) is performed via a digital network. Then, we can assign Step (1) to the sensor, Steps (2a)–(2c) to the controller and Steps (3a) and (3b) to the actuator. Assuming that all transmissions between the components can be done with negligible delay, we can run these three steps as separate algorithms in parallel. Denoting the real time by n, the resulting scheduling structure is sketched in Fig. 7.10 for $\tau_c^{\max} = 2$. For comparison, the structure of the NMPC Algorithm 3.11 without prediction is indicated by the dashed lines.

Since the algorithm is already applicable to work in parallel, it can be extended to a more complex networked control context in which transmission delays and packet loss may occur. To this end, such delays have to be considered in the prediction and an appropriate error handling must be added for handling dropouts; see, e.g., the paper by Grüne, Pannek and Worthmann [19]. In the presence of transmission delays and dropouts, we cannot expect that all control values are actually available at the actuator when they are supposed to be applied. Using NMPC, this can be compensated easily using the multistep feedback concept and the respective stability results from Sect. 7.4 as presented by Grüne et al. in [18].

Besides [19], which forms the basis for the presentation in this section, model based prediction for compensating computational delay in NMPC schemes has been considered earlier, e.g., in the works of Chen, Ballance and O'Reilly [4] and Findeisen and Allgöwer [10]. Note that the use of the nominal model (2.1) for predicting

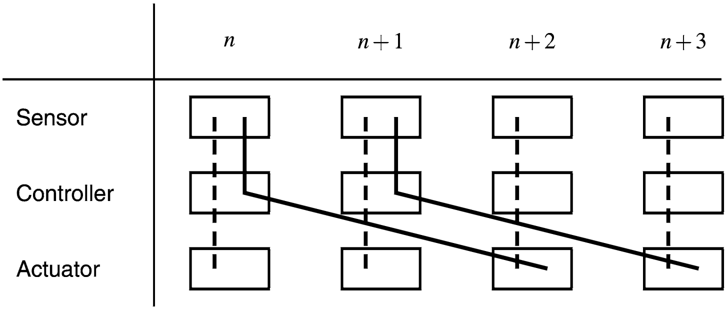


Fig. 7.10 Comparison of scheduling structure between NMPC Algorithms 3.11 (*dashed lines*) and 7.5 (*solid lines*) with $\tau_c^{\max} = 2T$

future states may lead to wrong predictions in case of model uncertainties, disturbances etc. In this case, the predicted state $\tilde{x}(\tilde{n})$ may differ from the actual state $x(n_a)$ at time $n_a = \tilde{n}$ and hence ($\mathrm{OCP}_{\mathrm{N,e}}$) is solved with a wrong initial value. In the paper of Zavala and Biegler [35] a method for correcting this mismatch based on NLP sensitivity techniques is presented, cf. also Sect. 10.5.

7.7 Online Measurement of α

In the analysis of NMPC schemes without stabilizing terminal constraints in Chap. 6, one of the central aims was to establish conditions to rigorously guarantee the existence of $\alpha \in (0, 1]$ such that the inequality

$$V_N(n, x) \geq \alpha \ell\big(n, x, \mu_N(n, x)\big) + V_N\big(n + 1, f\big(x, \mu_N(n, x)\big)\big) \qquad (5.1)$$

holds for all $x \in \mathbb{X}$ and $n \in \mathbb{N}_0$. While Theorem 6.14 and Proposition 6.17 provide computational methods for estimating α from the problem data, the assumptions needed for these computations—in particular Assumption 6.4—may be difficult to check.

In this section we present methods from Grüne and Pannek [15] and Pannek [29] which allow for the online computation or estimation of α along simulated NMPC closed-loop trajectories. There are several motivations for proceeding this way. First, as already mentioned, it may be difficult to check the assumptions needed for the computation of α using Theorem 6.14 or Proposition 6.17. Although a simulation based computation of α for a selection of closed-loop trajectories cannot rigorously guarantee stability and performance for all possible closed-loop trajectories, it may still give valuable insight into the performance of the controller. In particular, the information obtained from such simulations may be very useful in order to tune the controller parameters, in particular the optimization horizon N and the running cost ℓ.

Second, requiring (5.1) to hold for all $x \in \mathbb{X}$ may result in a rather conservative estimate for α. As we will see in Proposition 7.6, below, for assessing the performance of the controller along one closed-loop trajectory it is sufficient that (5.1) holds only for those points $x \in \mathbb{X}$ which are actually visited by this trajectory.

Finally, the knowledge of α may be used for an online adaptation of the optimization horizon N; some ideas in this direction are described in the subsequent Sect. 7.8.

Our first result shows that for assessing stability and performance of the NMPC controller along one specific closed-loop trajectory it is sufficient to find α such that (5.1) holds for the points actually visited by this trajectory.

Proposition 7.6 *Consider the feedback law* $\mu_N : \mathbb{N}_0 \times \mathbb{X} \to \mathbb{U}$ *computed from Algorithm* 3.7 *and the closed-loop trajectory* $x(\cdot) = x_{\mu_N}(\cdot)$ *of* (3.9) *with initial value* $x(0) \in \mathbb{X}$ *at initial time* 0. *If the optimal value function* $V_N : \mathbb{N}_0 \times \mathbb{X} \to \mathbb{R}_0^+$ *satisfies*

$$V_N\bigl(n, x(n)\bigr) \geq V_N\bigl(n+1, x(n+1)\bigr) + \alpha \ell\bigl(n, x(n), \mu_N\bigl(n, x(n)\bigr)\bigr) \tag{7.10}$$

for some $\alpha \in (0, 1]$ *and all* $n \in \mathbb{N}_0$, *then*

$$\alpha V_\infty\bigl(n, x(n)\bigr) \leq \alpha J_\infty\bigl(n, x(n), \mu_N\bigr) \leq V_N\bigl(n, x(n)\bigr) \leq V_\infty\bigl(n, x(n)\bigr) \tag{7.11}$$

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

If, in addition, there exist $\alpha_1, \alpha_2, \alpha_3 \in \mathcal{K}_\infty$ *such that* (5.2) *holds for all* $(n, x) \in \mathbb{N}_0 \times \mathbb{X}$ *with* $n \in \mathbb{N}_0$ *and* $x = x(n)$, *then there exists* $\beta \in \mathcal{KL}$ *which only depends on* $\alpha_1, \alpha_2, \alpha_3$ *and* α *such that the inequality*

$$\bigl|x(n)\bigr|_{x^{\mathrm{ref}}(n)} \leq \beta\bigl(\bigl|x(0)\bigr|_{x^{\mathrm{ref}}(0)}, n\bigr)$$

holds for all $n \in \mathbb{N}_0$, *i.e.*, x *behaves like a trajectory of an asymptotically stable system.*

Proof The proof of (7.11) is similar to the proof of Theorem 4.11.

The existence of β follows with the same construction as in the proof of Theorem 2.19, observing that the definition of β in this proof only depends on α_1, α_2 and $\alpha_V = \alpha\alpha_3$ and not on the specific form of $V = V_N$. □

Proposition 7.6 gives us a way to compute α from the data available at runtime and guarantees the performance estimate (7.11) as well as—under the additional assumption that (5.2) holds—asymptotic stability-like behavior for the considered closed-loop trajectory if $\alpha > 0$. Moreover, under this additional assumption (7.10) immediately implies that V_N strictly decreases along the trajectory, i.e., it behaves like a Lyapunov function.

Since the values of α for which (5.1) holds for all $x \in \mathbb{X}$ and for which (7.10) holds along a specific trajectory x_{μ_N} will be different in general, we introduce the following definition.

Definition 7.7

(1) We call $\alpha := \max\{\alpha \mid (5.1) \text{ holds for all } x \in \mathbb{X}\}$ the *global suboptimality degree*.
(2) For fixed $x \in \mathbb{X}$ the maximal value of α satisfying (5.1) for this x is called *local suboptimality degree* in x.

(3) Given a closed-loop trajectory $x_{\mu_N}(\cdot)$ of (3.9) with initial time 0 we call $\alpha := \max\{\alpha \mid (7.10) \text{ holds for all } n \in \mathbb{N}_0 \text{ with } x(\cdot) = x_{\mu_N}(\cdot)\}$ the *closed-loop suboptimality degree* along $x_{\mu_N}(\cdot)$.

An algorithm to evaluate α from (7.10) can easily be obtained and integrated into Algorithm 3.7:

Algorithm 7.8 (NMPC algorithm for time varying reference x^{ref} with a posteriori suboptimality estimate) Set $\alpha = 1$. At each sampling time t_n, $n = 0, 1, 2, \ldots$:

(1) Measure the state $x(n) \in X$ of the system.
(2) Set $x_0 = x(n)$ and solve the optimal control problem

$$\begin{array}{|l|}\hline \text{minimize} \quad J_N\big(n, x_0, u(\cdot)\big) := \sum_{k=0}^{N-1} \ell\big(n+k, x_u(k, x_0), u(k)\big) \\ \text{with respect to} \quad u(\cdot) \in \mathbb{U}^N(x_0), \quad \text{subject to} \\ x_u(0, x_0) = x_0, \qquad x_u(k+1, x_0) = f\big(x_u(k, x_0), u(k)\big) \\ \hline \end{array} \qquad (\text{OCP}_N^n)$$

and denote the obtained optimal control sequence by $u^\star(\cdot) \in \mathbb{U}^N(x_0)$.
(3) Define the NMPC-feedback value $\mu_N(n, x(n)) := u^\star(0) \in U$ and use this control value in the next sampling period.
(4) If $n \geq 1$ compute α via

$$\begin{array}{|l|}\hline \alpha_l = \dfrac{V_N(n-1, x(n-1)) - V_N(n, x(n))}{\ell(n-1, x(n-1), \mu_N(n-1, x(n-1)))}, \\ \alpha = \min\{\alpha, \alpha_l\}. \\ \hline \end{array}$$

Proposition 7.6 and Algorithm 7.8 are easily extended to the multistep NMPC case described in Sect. 7.4. In this case, (7.10) is replaced by

$$\begin{aligned} V_N\big(n, x(n)\big) \geq\; & V_N\big(n+m+1, x(n+m+1)\big) \\ & + \alpha \sum_{k=0}^{m} \ell\big(n+k, x_u\big(k, x(n)\big), u^\star\big(k, x(n)\big)\big) \end{aligned}$$

and the definition of α_l in Step (4) is changed, accordingly.

Note that in Step (4) of Algorithm 7.8, the computation of α_l does not provide the value of α in (7.10) for the current time instant n but for $n-1$. This is why we call α from Algorithm 7.8 an *a posteriori* estimate. The distinction between the current value of α_l and α in Step (4) is required in order to be consistent with Proposition 7.6 since α_l corresponds to the local suboptimality degree in $x(n-1)$ while the suboptimality degree according to Proposition 7.6 is the minimum over all α_l along the closed loop.

While Algorithm 7.8 is perfectly suited in order to evaluate the performance of an NMPC controller via numerical simulations, its a posteriori nature is not suitable if we want to use the estimated α in order to adjust the optimization horizon N. For instance, if we detect that at some time n the value of α in (7.10) is too small—or even negative—then we may want to increase N in order to increase α (see Sect. 7.8 for more details on such procedures). However, in Algorithm 7.8 the value of α in (7.10) only becomes available at time $n+1$, which is too late in order to adjust N.

A simple remedy for this problem is to solve at time n a second optimal control problem ($\mathrm{OCP}_\mathrm{N}^\mathrm{n}$) with initial value $x_u(1, x(n))$ and initial time $n := n+1$. However, since solving the problem ($\mathrm{OCP}_\mathrm{N}^\mathrm{n}$) is the computationally most expensive part of the NMPC algorithm, this solution would be rather inefficient.

In order to obtain an *a priori* estimate with reduced additional computing costs, a few more insights into the local NMPC problem structure are required. The main tool we are going to use is the following lemma.

Lemma 7.9 *Consider the feedback law* $\mu_N : \mathbb{N}_0 \times \mathbb{X} \to \mathbb{U}$ *computed from Algorithm* 3.7 *and the closed-loop trajectory* $x(\cdot) = x_{\mu_N}(\cdot)$ *of* (3.9) *with initial value* $x(0) = x_0 \in \mathbb{X}$ *at initial time* 0. *If*

$$\begin{aligned} &V_N\bigl(n+1, x(n+1)\bigr) - V_{N-1}\bigl(n+1, x(n+1)\bigr) \\ &\quad\le (1-\alpha)\ell\bigl(n, x(n), \mu_N\bigl(n, x(n)\bigr)\bigr) \end{aligned} \tag{7.12}$$

holds for some $\alpha \in [0, 1]$ *and some* $n \in \mathbb{N}_0$, *then* (7.11) *holds for this* n.

Proof Using the dynamic programming principle (3.16) with $K = 1$ we obtain

$$\begin{aligned} V_N\bigl(n, x(n)\bigr) &= \ell\bigl(n, x(n), \mu_N\bigl(n, x(n)\bigr)\bigr) + V_{N-1}\bigl(n+1, x(n+1)\bigr) \\ &\ge \ell\bigl(n, x(n), \mu_N\bigl(n, x(n)\bigr)\bigr) + V_N\bigl(n+1, x(n+1)\bigr) \\ &\quad - (1-\alpha)\ell\bigl(n, x(n), \mu_N\bigl(n, x(n)\bigr)\bigr) \\ &= V_N\bigl(n+1, x(n+1)\bigr) + \alpha\ell\bigl(n, x(n), \mu_N\bigl(n, x(n)\bigr)\bigr). \end{aligned}$$

Hence, (7.10) holds and Proposition 7.6 guarantees the assertion. □

Now, we would not gain much if we tried to compute α using (7.12) directly, since we would again need the future information $V_N(n+1, x(n+1))$, i.e., the solution of another optimal control problem (in contrast to that $V_{N-1}(n+1, x(n+1))$ is readily available at time n since by the dynamic programming principle it can be computed from $V_N(n, x(n))$ and $\ell(x(n), \mu_N(x(n))))$. There is, however, a way to reduce the size of the additional optimal control problem that needs to be solved. To this end, we introduce the following assumption which will later be checked numerically in our algorithm.

Assumption 7.10 For given $N, N_0 \in \mathbb{N}$, $N \ge N_0 \ge 2$, there exists a constant $\gamma > 0$ such that for the optimal open-loop solution $x_{u^\star}(\cdot, x(n))$ of ($\mathrm{OCP}_\mathrm{N}^\mathrm{n}$) in Algorithm 3.7 the inequalities

$$\frac{V_{N_0}(n+N-N_0, x_{u^\star}(N-N_0, x(n)))}{\gamma+1}$$

$$\leq \max_{j=N-N_0,\ldots,N-2} \ell\big(n+j, x_u\big(n+j, x(n)\big), \mu_{N-j-1}\big(n+j, x_{u^\star}\big(j, x(n)\big)\big)\big),$$

$$\frac{V_k(n+N-k, x_{u^\star}(N-k, x(n)))}{\gamma+1}$$

$$\leq \ell\big(n+N-k, x_{u^\star}\big(N-k, x(n)\big), \mu_k\big(n+N-k, x_{u^\star}\big(N-k, x(n)\big)\big)\big)$$

hold for all $k \in \{N_0+1, \ldots, N\}$ and all $n \in \mathbb{N}_0$.

Note that computing γ for which this assumption holds requires only the computation of μ_j for $j = 1, \ldots, N_0 - 1$ in the first inequality, since μ_k in the second inequality can be obtained from $u^\star$ via (3.23). This corresponds to solving $N_0 - 2$ additional optimal control problems which may look like a step backward, but since these optimal control problems are defined on a significantly smaller horizon, the computing costs are actually reduced. In fact, in the special case that ℓ does not depend on u, no additional computations have to be performed, at all. In this assumption, the value N_0 is a design parameter which affects the computational effort for checking Assumption 7.10 as well as the accuracy of the estimate for α obtained from this assumption.

Under Assumption 7.10 we can relate the minimal values of two optimal control problems with different horizon lengths.

Proposition 7.11 *Suppose that Assumption 7.10 holds for $N \geq N_0 \geq 2$. Then*

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

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

Proof In the following we use the abbreviation $x_u(j) := x_u(j, x(n))$, $j = 0, \ldots, N$, since all our calculations use the open-loop trajectory with fixed initial value $x(n)$.

Set $\tilde{n} := N - k$. Using the principle of optimality and Assumption 7.10 we obtain

$$\begin{aligned} &V_{k-1}\big(n+\tilde{n}+1, f\big(x_u(\tilde{n}), \mu_k\big(n+\tilde{n}, x_u(\tilde{n})\big)\big)\big) \\ &\quad \leq \gamma \ell\big(n+\tilde{n}, x_u(\tilde{n}), \mu_k\big(n+\tilde{n}, x_u(\tilde{n})\big)\big) \end{aligned} \tag{7.13}$$

for all $k \in \{N_0+1, \ldots, N\}$ and all $n \in \mathbb{N}_0$.

We abbreviate $\eta_k = \frac{(\gamma+1)^{k-N_0}}{(\gamma+1)^{k-N_0}+\gamma^{k-N_0+1}}$ and prove the main assertion $\eta_k V_k(n+\tilde{n}, x_u(\tilde{n})) \leq V_{k-1}(n+\tilde{n}, x_u(\tilde{n}))$ by induction over $k = N_0, \ldots, N$. By choosing
$x_u(0) = x(n)$ with n being arbitrary but fixed we obtain

$$\begin{aligned} &V_{N_0}\big(n+N-N_0, x_u(N-N_0)\big) \\ &\quad \leq (\gamma+1) \max_{j=2,\ldots,N_0} \ell\big(n+N-j, x_u(N-j), \mu_{j-1}\big(n+N-j, x_u(N-j)\big)\big) \\ &\quad \leq (\gamma+1) \sum_{j=2}^{N_0} \ell\big(n+N-j, x_u(N-j), \mu_{j-1}\big(n+N-j, x_u(N-j)\big)\big) \end{aligned}$$

$$= \frac{1}{\eta_{N_0}} V_{N_0-1}\big(n+N-N_0, x_u(N-N_0)\big).$$

For the induction step $k \to k+1$ the following holds, using (7.13) and the induction assumption:

$$\begin{aligned}
&V_k\big(n+\tilde{n}, x_u(\tilde{n})\big)\\
&\quad = V_{k-1}\big(n+\tilde{n}+1, f\big(x_u(\tilde{n}), \mu_k\big(n+\tilde{n}, x_u(\tilde{n})\big)\big)\big)\\
&\qquad + \ell\big(n+\tilde{n}, x_u(\tilde{n}), \mu_k\big(n+\tilde{n}, x_u(\tilde{n})\big)\big)\\
&\quad \ge \eta_k\left(1+\frac{1-\eta_k}{\gamma+\eta_k}\right) V_k\big(n+\tilde{n}+1, f\big(x_u(\tilde{n}), \mu_k\big(n+\tilde{n}, x_u(\tilde{n})\big)\big)\big)\\
&\qquad + \left(1-\gamma\frac{1-\eta_k}{\gamma+\eta_k}\right)\ell\big(n+\tilde{n}, x_u(\tilde{n}), \mu_k\big(n+\tilde{n}, x_u(\tilde{n})\big)\big)\\
&\quad = \eta_k\frac{\gamma+1}{\gamma+\eta_k}\big(V_k\big(n+\tilde{n}+1, f\big(x_u(\tilde{n}), \mu_k\big(n+\tilde{n}, x_u(\tilde{n})\big)\big)\big)\\
&\qquad + \ell\big(n+\tilde{n}, x_u(\tilde{n}), \mu_k\big(n+\tilde{n}, x_u(\tilde{n})\big)\big)\big)
\end{aligned}$$

using the dynamic programming principle (3.16) with $K=1$ in the last step. Hence, we obtain $V_k(n+\tilde{n}, x_u(\tilde{n})) \ge \eta_k \frac{\gamma+1}{\gamma+\eta_k} V_{k+1}(n+\tilde{n}, x_u(\tilde{n}))$ 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 choose $k=N$ then we get $\tilde{n}=0$. Inserting this into our induction result we can use $x_u(0)=x_u(0,x(n))=x(n)$ and the assertion holds. □

Finally, we can now use Proposition 7.11 within the NMPC closed loop. This allows us to verify Condition (7.12) and to estimate α directly from Assumption 7.10.

Theorem 7.12 *Consider $\gamma>0$ and $N, N_0 \in \mathbb{N}$, $N \ge N_0$ such that $(\gamma+1)^{N-N_0} > \gamma^{N-N_0+2}$ holds. If Assumption 7.10 is fulfilled for these γ, N and N_0, then the estimate (7.11) holds for all $n \in \mathbb{N}_0$ where*

$$\alpha := \frac{(\gamma+1)^{N-N_0}-\gamma^{N-N_0+2}}{(\gamma+1)^{N-N_0}}. \tag{7.14}$$

Proof From Proposition 7.11 we know

$$V_N\big(n,x(n)\big) - V_{N-1}\big(n,x(n)\big) \le \frac{\gamma^{N-N_0+1}}{(\gamma+1)^{N-N_0}} V_{N-1}\big(n,x(n)\big).$$

Setting $j=n-1$, we can reformulate this and obtain

$$\begin{aligned}
&V_N\big(j+1, x(j+1)\big) - V_{N-1}\big(j+1, x(j+1)\big)\\
&\quad \le \frac{\gamma^{N-N_0+1}}{(\gamma+1)^{N-N_0}} V_{N-1}\big(j+1, f\big(x_u\big(0,x(j)\big), \mu_N\big(j, x_u\big(0,x(j)\big)\big)\big)\big)
\end{aligned}$$

using the dynamics of the optimal open-loop solution. Now, we can use (7.13) with $k = N$ and get

$$V_N\big(j+1, x(j+1)\big) - V_{N-1}\big(j+1, x(j+1)\big) \leq \frac{\gamma^{N-N_0+2}}{(\gamma+1)^{N-N_0}} \ell\big(j, x(j), \mu_N\big(j, x(j)\big)\big).$$

Hence, the assumptions of Lemma 7.9 are fulfilled with

$$\alpha = 1 - \frac{\gamma^{N-N_0+2}}{(\gamma+1)^{N-N_0}} = \frac{(\gamma+1)^{N-N_0} - \gamma^{N-N_0+2}}{(\gamma+1)^{N-N_0}}$$

and the assertion follows. □

Similar to Proposition 7.6, the required values of γ and α are easily computable and allow us to extend Algorithm 3.7 in a similar manner as we did in Algorithm 7.8.

Algorithm 7.13 (NMPC algorithm for time varying reference x^{ref} with a priori suboptimality estimate) Set $\alpha = 1$. At each sampling time t_n, $n = 0, 1, 2, \ldots$:

(1) Measure the state $x(n) \in X$ of the system.
(2) Set $x_0 = x(n)$ and solve the optimal control problem

$$\begin{array}{|l|}\hline \text{minimize} \quad J_N\big(n, x_0, u(\cdot)\big) := \sum_{k=0}^{N-1} \ell\big(n+k, x_u(k, x_0), u(k)\big) \\ \text{with respect to} \quad u(\cdot) \in \mathbb{U}^N(x_0), \quad \text{subject to} \\ x_u(0, x_0) = x_0, \qquad x_u(k+1, x_0) = f\big(x_u(k, x_0), u(k)\big) \\ \hline \end{array} \qquad (\text{OCP}_\text{N}^\text{n})$$

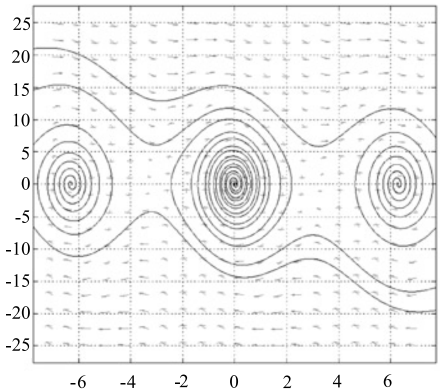
and denote the obtained optimal control sequence by $u^\star(\cdot) \in \mathbb{U}^N(x_0)$.
(3) Define the NMPC-feedback value $\mu_N(n, x(n)) := u^\star(0) \in U$ and use this control value in the next sampling period.
(4) Compute α via

$$\begin{array}{|l|}\hline \text{Find the minimal } \gamma \text{ which satisfies the inequalities} \\ \text{in Assumption 7.10 for the current } n \text{ and set} \\ \alpha = \min\left\{\alpha, \dfrac{(\gamma+1)^{N-N_0} - \gamma^{N-N_0+2}}{(\gamma+1)^{N-N_0}}\right\}. \\ \hline \end{array}$$

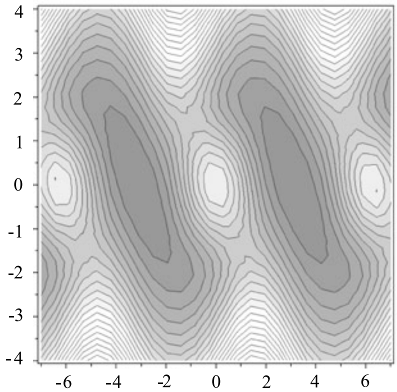
Note that checking the additional condition $(\gamma+1)^{N-N_0} > \gamma^{N-N_0+2}$ from Theorem 7.12 is unnecessary, since a violation would lead to a negative α in which case asymptotic stability cannot be guaranteed by means of Theorem 7.12, anyway.

Similar to Proposition 7.6, the results from Theorem 7.12 are easily carried over to the multistep NMPC case described in Sect. 7.4 by extending Assumption 7.10.

Example 7.14 To illustrate these results, we consider the inverted pendulum on a cart problem from Example 2.10 with parameters $g = 9.81$, $l = 10$ and $k_R = k_L =$



(a) Vector field of the inverted pendulum on a cart example 2.10. Here, the first two components are displayed which represent the pendulum.



(b) Contour plot of the cost function ℓ. Similar to Figure 7.11a, only the dependency on the first two components is displayed.

Fig. 7.11 Vector field and cost function

0.01 and control constraint set $\mathbb{U} = [-15, 15]$. Our aim is to stabilize one of the upright positions $x \in \mathcal{S} := \{((k+1)\pi, 0, 0, 0)^\top \mid k \in 2\mathbb{Z}\}$. For this example we will provide online measurements of α using Algorithm 7.8 for one fixed initial value and varying terminal weights ω, cf. Sect. 7.2, and control horizons, cf. Sect. 7.4. For a comparison of Algorithms 7.8 and 7.13 we refer to [15] and [29].

In order to obtain a suitable cost function, we follow the guidelines from Sect. 6.6 and construct a cost function for which—at least in the first two components—the overshoot of ℓ along a typical stable trajectory becomes small. To this end, we have used the geometry of the vector field of the first two differential equations representing the pendulum, see Fig. 7.11(a), and shaped the cost function such that it exhibits local maxima at the downward equilibria and "valleys" along the stable manifolds of the upright equilibria to be stabilized. The resulting cost function ℓ is of the integral type (3.4) with

$$\begin{aligned} L(x,u) := 10^{-4}u^2 + \big(&3.51\sin(x_1-\pi)^2 + 4.82\sin(x_1-\pi)x_2 \\ &+ 2.31x_2^2 + 0.1\big(\big(1-\cos(x_1-\pi)\big)\cdot\big(1+\cos(x_2)^2\big)\big)^2 \\ &+ 0.01x_3^2 + 0.1x_4^2\big)^2, \end{aligned}$$

cf. Fig. 7.11(b). Using the terminal weights from Sect. 7.2, the cost functional becomes

$$J_N(x_0,u) = \sum_{i=0}^{N-2} \ell\big(x(i),u(i)\big) + \omega\ell\big(x(N-1),u(N-1)\big).$$

This way of adjusting the cost function to the dynamics allows us to considerably reduce the length of the optimization horizon for obtaining stability in the NMPC scheme without stabilizing terminal constraints compared to simpler choices of ℓ.

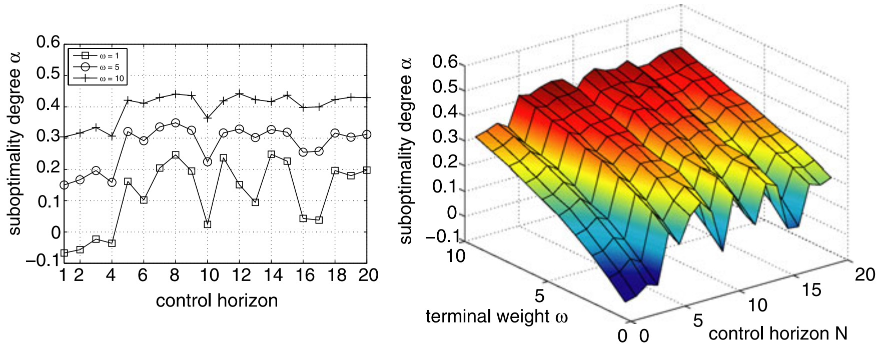


Fig. 7.12 Computed value for $\alpha^{\omega}_{70,m}$ for the nonlinear inverted pendulum example 2.10 with control horizons $m \in \{1, \ldots, 20\}$ and terminal weights $\omega \in \{1, \ldots, 10\}$

However, for the initial value $x_0 = (2\pi + 1.5, 0, 0, 0)$ and sampling period $T = 0.05$, which have been used in the subsequent computations, we still need a rather large optimization horizon of $N = 70$ to obtain stability of the closed loop.

Since the cost function is 2π-periodic it does not penalize the distance to a specific equilibrium in $\mathcal{S}$; rather, it penalizes the distance to the whole set. For a better comparison of the solutions for different parameters we want to force the algorithm to stabilize one specific upright position in $\mathcal{S}$. To this end, we add box-constraints to $\mathbb{X}$ limiting the x_1-component to the interval $[-\pi + 0.01, 3\pi - 0.01]$. The tolerances of the optimization routine and the differential equation solver are set to 10^{-6} and 10^{-7}, respectively. The NMPC closed-loop trajectories displayed in Fig. 7.12 are simulated for terminal weights $\omega = 1, \ldots, 10$, cf. Sect. 7.2, and control horizons $m = 1, \ldots, 10$, cf. Sect. 7.4. The resulting α-values from Algorithm 7.8, denoted by $\alpha^{\omega}_{N,m}$, are shown in Fig. 7.12.

Note that for $\omega = 1$ the α values are negative for control horizons $m = 1, \ldots, 4$. Still, larger control horizons exhibit a positive α value such that stability is guaranteed. This is in accordance with the theoretical results from Sect. 7.4, even though these simulation based results do not share the monotonicity of the theoretical bounds from Fig. 7.3. Additionally, an increase of α can be observed for all control horizons m if ω is increased. This confirms the stabilizing effect of terminal costs shown theoretically in Sect. 7.2; cf. Fig. 7.1.

Summarizing, these results show that the online measurement of α yields valuable insights into the performance analysis of NMPC schemes without terminal constraints and thus nicely complements the theoretical results from Chap. 6 and Sects. 7.2 and 7.4.

7.8 Adaptive Optimization Horizon

In the previous Sect. 7.7 we have shown how the suboptimality degree α can be computed at runtime of the NMPC scheme without stabilizing terminal constraints.

If the horizon length N is not chosen adequately, then it is likely that during runtime a value $\alpha < 0$ is obtained. In this case, stability of the closed loop cannot be guaranteed by Proposition 7.6 or Theorem 7.12. However, the ability to compute α for each point $x(n)$ on the closed-loop trajectory using the techniques from Sect. 7.7 naturally leads to the idea of adapting the optimization horizon N at each time n such that stability and desired performance can be guaranteed. In this section, we will show some algorithms for this purpose, taken from Pannek [29]. Here we restrict ourselves to the basic idea and refer to [29] for more sophisticated approaches.

The fundamental idea of such an adaptive algorithm is rather simple: introducing a stability and suboptimality threshold $\overline{\alpha} > 0$, at each sampling instant n we prolong the optimization horizon if α for the current horizon is smaller than $\overline{\alpha}$. If $\alpha > \overline{\alpha}$ holds, then we may reduce N in order to save computational time. This leads to the following algorithm.

Algorithm 7.15 (Adaptive horizon NMPC algorithm for time varying reference) Set $N_0 > 0$ and $\overline{\alpha} > 0$. At each sampling time t_n, $n = 0, 1, 2, \ldots$:

(1) Measure the state $x(n) \in X$ of the system and set $\alpha = 0$.
(2) While $\overline{\alpha} > \alpha$
  (a) Set $x_0 = x(n)$, $N = N_n$ and solve the optimal control problem

$$\boxed{\begin{array}{l} \text{minimize} \quad J_N\big(n, x_0, u(\cdot)\big) := \displaystyle\sum_{k=0}^{N-1} \ell\big(n+k, x_u(k, x_0), u(k)\big) \\ \text{with respect to} \quad u(\cdot) \in \mathbb{U}^N(x_0), \quad \text{subject to} \\ x_u(0, x_0) = x_0, \qquad x_u(k+1, x_0) = f\big(x_u(k, x_0), u(k)\big). \end{array}} \qquad (\mathrm{OCP}_\mathrm{N}^\mathrm{n})$$

  Denote the obtained optimal control sequence by $u^\star(\cdot) \in \mathbb{U}^N(x_0)$.
  (b) Compute α via Proposition 7.6 or Theorem 7.12.
  (c) If $\alpha > \overline{\alpha}$ call reducing strategy for N_n, else call increasing strategy for N_n; obtain $u^\star(\cdot)$ for the new $N = N_n$ and an initial guess for N_{n+1}.
(3) Define the NMPC-feedback value $\mu_N(n, x(n)) := u^\star(0) \in U$ and use this control value in the next sampling period.

Here, the initial guess N_{n+1} in Step (2c) will typically be $N_{n+1} = N_n$, however, as we will see below, in the case of reducing N_n the choice $N_{n+1} = N_n - 1$ is more efficient, cf. the discussion after Proposition 7.18.

If this algorithm is successful in ensuring $\alpha \geq \overline{\alpha}$ for each n, then the assumptions of Proposition 7.6 or Theorem 7.12 are satisfied. However, these results require the optimization horizon N to be fixed and hence do not apply to Algorithm 7.15 in which N_n changes with time.

To cope with this issue, we generalize Proposition 7.6 to varying optimization horizons. To this end, for each $x \in \mathbb{X}$ and $N \in \mathbb{N}$ we denote the maximal α from (7.10) by $\alpha(N)$. We then introduce the following assumption, which guarantees that for any horizon N satisfying $\alpha(N) \geq \overline{\alpha}$ the controller shows a bounded guaranteed performance if the horizon length is increased.

Assumption 7.16 Given $n \in \mathbb{N}_0$, $x \in \mathbb{X}$, $N < \infty$ and a value $\overline{\alpha} \in (0, 1)$ with $\alpha(N) \geq \overline{\alpha}$, we assume that there exist constants $C_l, C_\alpha > 0$ such that the inequalities

$$\begin{aligned} &C_l \ell\bigl(n, x, \mu_N(n, x)\bigr) \\ &\quad \leq \ell\bigl(n, x, \mu_{\widetilde{N}}(n, x)\bigr) \frac{V_{\widetilde{N}}(n, x) - V_{\widetilde{N}}(n+1, f(x, \mu_N(n, x)))}{V_{\widetilde{N}}(n, x) - V_{\widetilde{N}}(n+1, f(x, \mu_{\widetilde{N}}(n, x)))}, \end{aligned} \tag{7.15}$$

$$C_\alpha \alpha(N) \leq \alpha(\widetilde{N}) \tag{7.16}$$

hold for all $\widetilde{N} \geq N$.

The reason for Assumption 7.16 is that it is possible that the performance of the controller μ_N may not improve monotonically as N increases; see Di Palma and Magni [6]. Consequently, we cannot expect $\alpha(\widetilde{N}) \geq \alpha(N)$ for $\widetilde{N} > N$. Still, we need to ensure that $\alpha(\widetilde{N})$ does not become too small compared to $\alpha(N)$, in particular, $\alpha(\widetilde{N})$ should not drop below zero if the horizon length is increased; this is ensured by (7.16). Furthermore, we need an estimate for the dependence of $\ell(n, x, \mu_N(n, x))$ on N which is given by (7.15). Unfortunately, for both inequalities so far we were not able to provide sufficient conditions in terms of the problem data, like, e.g., a controllability condition similar to Assumption 6.4. Still, numerical evaluation for several examples showed that these inequalities are satisfied and that C_l and C_α attain reasonable values.

Using Assumption 7.16, we obtain a stability and performance estimate of the closed loop in the context of changing horizon lengths similar to Proposition 7.6. Since the closed-loop control resulting from Algorithm 7.15 now depends on a sequence of horizons $(N_n)_{n\in\mathbb{N}_0}$ we obtain a sequence of control laws $(\mu_{N_n})_{n\in\mathbb{N}_0}$. The closed-loop trajectory generated by this algorithm is then given by

$$x(n+1) = f\bigl(x(n), \mu_{N_n}\bigl(n, x(n)\bigr)\bigr). \tag{7.17}$$

Theorem 7.17 *Consider the sequence of feedback laws (μ_{N_n}) computed from Algorithm 7.15 and the corresponding closed-loop trajectory $x(\cdot)$ from (7.17). Assume that for optimal value functions $V_{N_n} : \mathbb{N}_0 \times \mathbb{X} \to \mathbb{R}_0^+$ of ($\mathrm{OCP}_\mathrm{N}^\mathrm{n}$) with $N = N_n$ the inequality*

$$V_{N_n}\bigl(n, x(n)\bigr) \geq V_{N_n}\bigl(n+1, x(n+1)\bigr) + \overline{\alpha}\ell\bigl(n, x(n), \mu_{N_n}\bigl(n, x(n)\bigr)\bigr) \tag{7.18}$$

holds for all $n \in \mathbb{N}_0$ and that Assumption 7.16 is satisfied for all triplets $(n, x, N) = (n, x(n), N_n)$, $n \in \mathbb{N}_0$, with constants $C_l^{(n)}$, $C_\alpha^{(n)}$. Then

$$\alpha_C V_\infty\bigl(n, x(n)\bigr) \leq \alpha_C J_\infty\bigl(n, x(n), \mu_{(N_n)}\bigr) \leq V_{N^\star}\bigl(n, x(n)\bigr) \leq V_\infty\bigl(n, x(n)\bigr) \tag{7.19}$$

holds for all $n \in \mathbb{N}_0$ where $\alpha_C := \min_{i\in\mathbb{N}_{\geq n}} C_\alpha^{(i)} C_l^{(i)} \overline{\alpha}$.

Proof Given $(i, x(i), N_i)$ for some $i \in \mathbb{N}_0$, Assumption 7.16 for $(n, x, N) = (i, x(i), N_i)$ guarantees $\alpha(N_i) \leq \alpha(\widetilde{N})/C_\alpha^{(i)}$ for $\widetilde{N} \geq N_i$. Choosing $\widetilde{N} = N^\star$, we obtain $\overline{\alpha} \leq \alpha(N_i) \leq \alpha(N^\star)/C_\alpha^{(i)}$ using the relaxed Lyapunov Inequality (7.18). Multiplying by the stage cost $\ell(i, x(i), \mu_{N_i}(i, x(i)))$, we can conclude

$$\begin{aligned}
&\overline{\alpha}\ell\big(i, x(i), \mu_{N_i}\big(i, x(i)\big)\big)\\
&\quad\le \frac{\alpha(N^\star)}{C_\alpha^{(i)}}\ell\big(i, x(i), \mu_{N_i}\big(i, x(i)\big)\big)\\
&\quad= \frac{V_{N^\star}(i, x(i)) - V_{N^\star}(i+1, f(x(i), \mu_{N^\star}(i, x(i))))}{C_\alpha^{(i)}\ell(i, x(i), \mu_{N^\star}(i, x(i)))}\ell\big(i, x(i), \mu_{N_i}\big(i, x(i)\big)\big)\\
&\quad\le \frac{V_{N^\star}(i, x(i)) - V_{N^\star}(i+1, f(x(i), \mu_{N_i}(i, x(i))))}{C_\alpha^{(i)}C_l^{(i)}}
\end{aligned}$$

using (7.18) and (7.15). In particular, the latter condition allows us to use an identical telescope sum argument as in the proof of Proposition 7.6 since it relates the closed-loop varying optimization horizon to a fixed one. Hence, summing the running costs along the closed-loop trajectory reveals

$$\alpha_C \sum_{i=n}^{K} \ell\big(i, x(i), \mu_{N_i}\big(i, x(i)\big)\big) \le V_{N^\star}\big(n, x(n)\big) - V_{N^\star}\big(K+1, x(K+1)\big)$$

where we defined $\alpha_C := \min_{i\in[n,\dots,K]} C_\alpha^{(i)} C_l^{(i)} \overline{\alpha}$. Since $V_{N^\star}(K+1, x(K+1)) \ge 0$ holds, we can neglect it in the last inequality. Taking K to infinity yields

$$\alpha_C V_\infty^{\mu_{(N_i)}}\big(n, x(n)\big) = \alpha_C \lim_{K\to\infty} \sum_{i=n}^{K} \ell\big(i, x(i), \mu_{N_i}\big(i, x(i)\big)\big) \le V_{N^\star}\big(n, x(n)\big).$$

Since the first and the last inequality of (7.19) hold by definition of V_N and V_∞, the assertion follows. □

If the conditions of this theorem hold, then stability-like behavior of the closed loop can be obtained analogously to Proposition 7.6.

Having shown the analytical background, we now present adaptation strategies which can be used for increasing or reducing the optimization horizon N in Step (2c) of Algorithm 7.15. For simplicity of exposition, we restrict ourselves to two simple strategies and consider a posteriori estimates based variants only. Despite their simplicity, these methods have shown to be reliable and fast in numerical simulations. A more detailed analysis, further methods and comparisons can be found in [29]. The following proposition yields the basis for a strategy for reducing N_n.

Proposition 7.18 *Consider the optimal control problem* ($\mathrm{OCP}_\mathrm{N}^\mathrm{n}$) *with initial value* $x_0 = x(n)$, $N_n \in \mathbb{N}$, *and denote the optimal control sequence by* $u^\star$. *For fixed* $\overline{\alpha} \in (0, 1)$, *suppose there exists an integer* $\overline{i} \in \mathbb{N}_0$, $0 \le \overline{i} < N$ *such that*

$$\begin{aligned}
&V_{N_n-i}\big(n+i+1, x_{u^\star}\big(i+1, x(n)\big)\big)\\
&\qquad + \overline{\alpha}\ell\big(n+i, x_{u^\star}\big(i, x(n)\big), \mu_{N_n-i}\big(n+i, x_{u^\star}\big(i, x(n)\big)\big)\big)\\
&\quad \le V_{N_n-i}\big(n+i, x_{u^\star}\big(i, x(n)\big)\big)
\end{aligned} \tag{7.20}$$

holds for all $0 \le i \le \overline{i}$. *Then, setting* $N_{n+i} = N_n - i$ *and* $\mu_{N_{n+i}}(n+i, x(n+i)) = u^\star(i)$ *for* $0 \le i \le \overline{i} - 1$, *Inequality* (7.18) *holds for* $n = n, \dots, n + \overline{i} - 1$.

Proof The proof follows immediately from the fact that for $\mu_{N_{n+i}}(n+i, x(n+i)) = u^\star(i)$ the closed-loop trajectory (7.17) satisfies $x(n+i) = x_{u^\star}(i, x(n))$. Hence, (7.18) follows from (7.20). □

Observe that Proposition 7.18 is quite similar to the results from Sect. 7.4, since $\mu_{N_{n+i}}(n+i, x(n+i))$ as defined in this theorem coincides with the multistep feedback law from Sect. 7.4. Thus, Proposition 7.18 guarantees that if $\bar{i} > 1$, then the multistep NMPC feedback from Sect. 7.4 can be applied with $m = \bar{i}$ steps such that the suboptimality threshold $\overline{\alpha}$ can be guaranteed. With the choice $N_{n+i} = N_n - i$, due to the principle of optimality we obtain that the optimal control problems within the next $\bar{i} - 1$ NMPC iterations are already solved since $\mu_{N_{n+i}}(n+i, x(n+i))$ can be obtained from the optimal control sequence $u^\star(\cdot) \in \mathbb{U}^N(x(n))$ computed at time n. This implies that the most efficient way for the reducing strategy in Step (2c) of Algorithm 7.15 is not to reduce N_n itself but rather to reduce the horizons N_{n+i} by i for the subsequent sampling instants $n+1, \ldots, n+\bar{i}$, i.e., we choose the initial guess in Step (2c) as $N_{n+1} = N_n - 1$. Still, if the a posteriori estimate is used, the evaluation of (7.20) requires the solution of an additional optimal control problem in each step in order to compute $V_{N_n - i}(n+i+1, x_{u^\star}(i+1, x(n)))$.

In contrast to this efficient and simple shortening strategy, it is quite difficult to obtain efficient methods for prolonging the optimization horizon N in Step (2c) of Algorithm 7.15. In order to understand why this is the case, we first introduce the basic idea behind any such prolongation strategy: at each sampling instant we iteratively increase the horizon N_n until (7.18) is satisfied and use this horizon for the next NMPC step. In order to ensure that iteratively increasing N_n will eventually lead to a horizon for which (7.18) holds, we make the following assumption.

Assumption 7.19 Given $\overline{\alpha} \in (0, 1)$, for all $x_0 \in \mathbb{X}$ and all $n \in \mathbb{N}_0$ there exists a finite horizon length $\overline{N} = \overline{N}(n, x_0) \in \mathbb{N}$ such that (7.18) holds with $\alpha(N_n) \geq \overline{\alpha}$ for $x(n) = x_0$ and $N_n \geq \overline{N}$.

Assumption 7.19 can be seen as a performance assumption which requires the existence of a horizon length N_n such that the predefined threshold $\overline{\alpha}$ can be satisfied. If no such horizon exists, no prolongation strategy can be designed which can guarantee closed-loop suboptimality degree $\alpha > \overline{\alpha}$. Assumption 7.19 is, for instance, satisfied if the conditions of Theorem 6.21 hold.

The following proposition shows that under this assumption any iterative strategy which increases the horizon will terminate after finitely many steps with a horizon length N for which the desired local suboptimality degree holds.

Proposition 7.20 *Consider the optimal control problem* ($\mathrm{OCP}^{\mathrm{n}}_{\mathrm{N}}$) *with initial value* $x_0 = x(n)$ *and* $N_n \in \mathbb{N}$. *For fixed* $\overline{\alpha} \in (0, 1)$ *suppose that Assumption 7.19 holds. Then, any algorithm which iteratively increases the optimization horizon* N_n *and terminates if* (7.18) *holds will terminate in finite time with an optimization horizon* N_n *for which* (7.18) *holds. In particular, Theorem 7.17 is applicable provided Assumption 7.16 holds.*

Proof The proof follows immediately from Assumption 7.19. □

Unfortunately, if (7.18) does not hold it is in general difficult to assess by how much N_n should be increased such that (7.18) holds for the increased N_n. The most simple strategy of increasing N_n by one in each iteration shows satisfactory results in practice, however, in the worst case it requires us to check (7.18) $\overline{N} - N_n + 1$ times at each sampling instant. In contrast to the shortening strategy, the principle of optimality cannot be used here to establish a relation between the optimal control problems for different N_n and, moreover, these problems may exhibit different solution structures which makes it a hard task to provide a suitable initial guess for the optimization algorithm; see also Sect. 10.5.

In order to come up with more efficient strategies, different methods have been developed [29] which utilize the structure of the suboptimality estimate itself to determine by how much N_n should be increased. Compared to these methods, however, the performance of the simple strategy of increasing N_n by one is still acceptable. In the following example we illustrate the performance of this strategy for Example 2.11.

Example 7.21 For the ARP system (2.19)–(2.26) we have already analytically derived a continuous time tracking feedback in (2.28). However, this feedback law performs poorly under sampling, in particular, for the sampling period $T = 0.2$ which we consider here we obtain an unstable closed-loop sampled data system.

In order to obtain a sampled data feedback law which shows better performance we use the digital redesign technique proposed by Nešić and Grüne in [27]: given a signal $v(t)$ to track, we numerically simulate the continuous time controlled system in order to generate the output x^{ref} which in turn will be used as the reference trajectory for an NMPC tracking problem. The advantage of proceeding this way compared to the direct formulation of an NMPC tracking problem lies in the fact that—according to our numerical experience—the resulting NMPC problem is much easier to solve and in particular requires considerably smaller optimization horizons in order to obtain a stable NMPC closed loop.

Specifically, we consider the piecewise constant reference function

$$v(t) = \begin{cases} 10, & t \in [0, 5) \cup [9, 10), \\ 0, & t \in [5, 9) \cup [10, 15) \end{cases}$$

for the x_5-component of the trajectory of the system. In order to obtain short transient times for the continuous time feedback, we set the design parameters c_i in (2.28) to $(c_0, c_1, c_2, c_3) = (10\,000, 3500, 500, 35)$. Then, we incorporate the resulting trajectory displayed in Fig. 7.13 as reference $x^{\text{ref}}(\cdot)$ in the NMPC algorithm. Since our goal is to track the reference with the x_5-component of the trajectory, we use the simple quadratic cost function

$$J(x_0, u) = \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \left| x_{5,u}(t, x_0) - x_{5,\text{ref}}(t) \right| dt$$

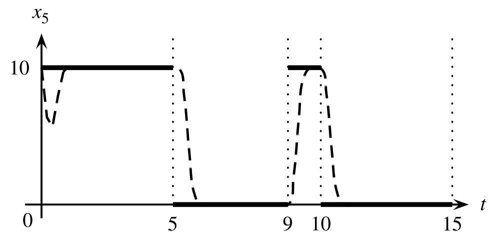


Fig. 7.13 Reference function for the continuous time feedback (*solid*) and state trajectory using the continuous time feedback (*dashed*). The latter will be used as reference function within the NMPC algorithm

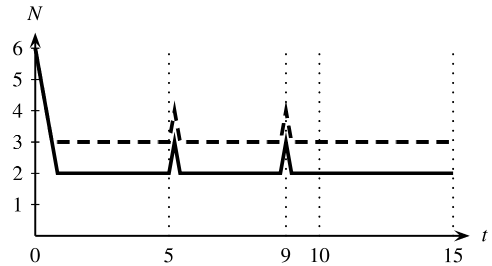


Fig. 7.14 Optimization horizons computed by the adaptive NMPC Algorithm 7.15 for the ARP problem using the a posteriori estimate (*solid*) and the a priori estimate (*dashed*)

within the adaptive horizon NMPC Algorithm 7.15. Moreover, we select the sampling period $T = 0.2$ and fix the initial value $x(0) = (0, 0, 0, 0, 10, 0, 0, 0)$ for both the continuously and the sampled-data controlled system.

Using the a posteriori and a priori estimation techniques within the adaptive NMPC Algorithm 7.15, we obtain the evolutions of horizons N_n along the closed loop for the suboptimality bound $\overline{\alpha} = 0.1$ as displayed in Fig. 7.14. Comparing the horizons chosen by the a priori and the a posteriori estimates, one sees that the a posteriori algorithms yields smaller optimization horizons which makes the resulting scheme computationally more efficient, however, at the expense that the evaluation of the a posteriori criterion itself is computationally more demanding; see also Fig. 7.15, below.

It is also interesting to compare these horizons to the standard NMPC Algorithm 3.7 with fixed N which needs a horizon of $N = 6$ in order to guarantee $\alpha \geq \overline{\alpha}$ along the closed loop. Here, one observes that the required horizon N_n for the adaptive NMPC approach is typically smaller than $N = 6$ for both the a posteriori and the a priori estimate based variant. One also observes that the horizon is increased at the jump points of the reference function $\upsilon(\cdot)$, which is the behavior one would expect in a "critical" situation and nicely reflects the ability of the adaptive horizon algorithm to adapt to the new situation.

Although the algorithm chooses to modify the horizon length throughout the run of the closed loop, one can barely see a difference between the resulting x_5 trajectories and the (dashed) reference trajectory given in Fig. 7.13. For this reason, we do not display the closed-loop solutions. Instead, we additionally plotted the computing times of the two adaptive NMPC variants in Fig. 7.15. Again, one can immediately see the spikes in the graph right at the points in which $\upsilon(\cdot)$ jumps. This figure also illustrates the disadvantage of the algorithm of having to solve multiple additional optimal control problems whenever N is increased, which clearly shows up in the

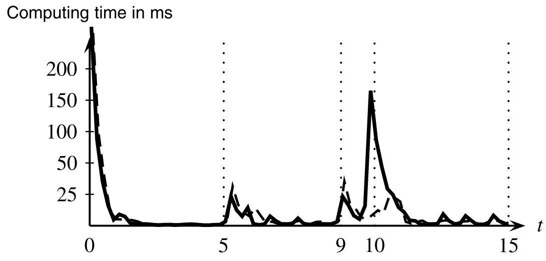


Fig. 7.15 Computing times of the adaptive NMPC Algorithm 7.15 for the ARP problem using the a posteriori estimate (*solid*) and the a priori estimate (*dashed*)

higher computation times at these points, in particular for the computationally more expensive a posteriori estimation.

While the adaptive optimization horizon algorithm produces good results in this example, we would like to mention that there are other examples—like, e.g., the swing-up of the inverted pendulum—for which the algorithm performs less convincing. We conjecture that a better understanding of Assumption 7.16 may provide the insight needed in order to tell the situations in which the adaptive algorithms provides good results from those in which it does not.

7.9 Nonoptimal NMPC

In the case of limited computational resources and/or fast sampling, the time available for solving the optimization problems ($\mathrm{OCP_N}$) or its variants may not be sufficient to obtain an arbitrary accurate solution. Typically, the algorithms for solving these problems, i.e., for obtaining $u^\star$ and thus $\mu_N(x(n)) = u^\star(0)$, work iteratively[2] and with limited computation time may we may be forced to terminate this algorithm prior to convergence to the optimal control sequence $u^\star$.

It is therefore interesting to derive conditions which ensure stability and performance estimates for the NMPC closed loop in this situation. To this end, we modify Algorithm 3.1 as follows.

Algorithm 7.22 We replace Steps (2) and (3) of Algorithm 3.1 (or its variants) by the following:

(2′) For initial value $x_0 = x(n)$, given an initial guess $u_n^0(\cdot) \in U^N$ we iteratively compute $u_n^j(\cdot) \in U^N$ by an iterative optimization algorithm such that

$$J_N\big(x_0, u_n^{j+1}(\cdot)\big) \le J_N\big(x_0, u_n^j(\cdot)\big).$$

We terminate this iteration after $j^* \in \mathbb{N}$ iterations, set $u_n(\cdot) := u_n^{j^*}(\cdot)$ and $\widetilde{V}_N(n) := J_N(x_0, u_n(\cdot))$.

[2]For more information on these algorithms see Chap. 10 and for numerical aspects of the theory in this chapter in particular Sect. 10.6.

(3′) Define the NMPC-feedback value $\mu_N(x(n)) := u_n(0) \in U$ and use this control value in the next sampling period.

One way to ensure proper operation of such an algorithm is by assuming that the sampling period is so small such that the optimal control from sampling instant $n-1$ is still "almost optimal" at time n. In this case, one iteration starting from $u_n^0 = u_{n-1}$, i.e., $j^* = 1$, may be enough in order to be sufficiently close to an optimal control, i.e., to ensure $J_N(x(n), u_n^1) \approx V_N(x(n))$. This procedure is, e.g., investigated by Diehl, Findeisen, Allgöwer, Bock and Schlöder in [8].

An alternative but conceptually similar idea is presented in work of Graichen and Kugi [11]. In this reference a sufficiently large number of iterations j^* is fixed and conditions are given under which the control sequences $u_n^{j^*}$ become more and more optimal as n increases, i.e., they satisfy $J_N(x(n), u_n^{j^*}) \approx V_N(u^\star)$ for sufficiently large n. Using suitable bounds during the transient phase in which this approximate optimality does not yet hold then allows the authors to conclude stability estimates.

While these results use that $u_n^{j^*}$ is close to $u^\star$ in an appropriate sense, here we investigate the case in which $u_n^{j^*}$ may be far away from the optimal solution. As we will see, asymptotic stability in the sense of Definition 2.14 is in general difficult to establish in this case. However, it will still be possible to prove the following weaker property.

Definition 7.23 Given a set $S \subseteq \mathbb{X}$, we say that the NMPC closed loop (2.5) is *attractive on S* if for each $x \in S$ the convergence

$$\lim_{k\to\infty} x_{\mu_N}(k, x) = x_*$$

holds.

Contrary to asymptotic stability, a merely attractive solution x_{μ_N} which starts close to the equilibrium x_* may deviate far from it before it eventually converges to x_*. In order to exclude this undesirable behavior, one may wish to require the following stability property in addition to attraction.

Definition 7.24 Given a set $S \subseteq \mathbb{X}$, we say that the NMPC closed loop (2.5) is *stable on S* if there exists $\alpha_S \in \mathcal{K}$ such that the inequality

$$\left|x_{\mu_N}(k, x)\right|_{x_*} \le \alpha_S\left(|x|_{x_*}\right)$$

holds for all $x \in S$ and all $k = 0, 1, 2, \ldots$.

It is well known that under suitable regularity conditions attractivity and stability imply asymptotic stability; see, e.g., the book of Khalil [23, Chap. 4]. Since this is not the topic of this book, we will not go into technical details here and rather work with the separate properties attractivity and stability in the remainder of this section.

The following variant of Proposition 7.6 will be used in order to ensure attractivity and stability.

Proposition 7.25 *Consider the solution* $x(n) = x_{\mu_N}(n, x_0)$ *of the NMPC closed loop* (2.5), *a set* $S \subseteq \mathbb{X}$, *a value* $\alpha \in (0, 1]$. *Assume that* ℓ *satisfies*

$$\ell^*(x) \geq \alpha_3\big(|x|_{x_*}\big) \tag{7.21}$$

for some $\alpha_3 \in \mathcal{K}_\infty$ *and all* $x \in S$ *and that for each* $x_0 \in S$ *there exists a function* $\widetilde{V}_N : \mathbb{N}_0 \to \mathbb{R}_0^+$ *which for all* $n \in \mathbb{N}_0$ *satisfies*

$$\widetilde{V}_N(n) \geq \widetilde{V}_N(n+1) + \alpha\ell\big(x(n), \mu_N\big(x(n)\big)\big). \tag{7.22}$$

Then the closed loop (2.5) *is attractive on* S *and the inequality*

$$J_\infty(x_0, \mu_N) \leq \widetilde{V}_N(0) \tag{7.23}$$

holds for $J_\infty(x_0, \mu_N)$ *from Definition* 4.10.

If, in addition, there exists $\tilde{\alpha}_2 \in \mathcal{K}_\infty$ *independent of* x_0 *such that the functions* $\widetilde{V}_N$ *satisfy*

$$\widetilde{V}_N(0) \leq \tilde{\alpha}_2\big(\big|x(0)\big|_{x_*}\big), \tag{7.24}$$

then the closed loop (2.5) *is stable on* S.

Proof Iterating Inequality (7.22) for $n = 0, \ldots, k$ and using $\widetilde{V}_N(n) \geq 0$ yields

$$\sum_{n=0}^{k} \ell\big(x(n), \mu_N\big(x(n)\big)\big) \leq \widetilde{V}_N(0) - \widetilde{V}_N(k+1) \leq \widetilde{V}_N(0).$$

Letting $k \to \infty$ we obtain

$$J_\infty(x, \mu_N) = \lim_{k\to\infty} \sum_{n=0}^{k} \ell\big(x(n), \mu_N\big(x(n)\big)\big) \leq \widetilde{V}_N(0),$$

i.e., (7.23). Now nonnegativity of ℓ implies $\lim_{n\to\infty} \ell(x(n), \mu_N(x(n))) = 0$ and thus (7.21) implies $x(n) \to 0$, i.e., attractivity.

In order to prove stability under the additional assumption (7.24), observe that (7.22) together with the nonnegativity of $\widetilde{V}_N$ and (7.21) implies

$$\widetilde{V}_N(n) \geq \alpha\ell\big(x(n), \mu_N\big(x(n)\big)\big) \geq \alpha\,\alpha_3\big(\big|x(n)\big|_{x_*}\big) =: \tilde{\alpha}_1\big(\big|x(n)\big|_{x_*}\big).$$

Furthermore, (7.22) implies that $\widetilde{V}_N(n)$ is decreasing in n. Using these properties, stability immediately follows from

$$\big|x(n)\big|_{x_*} \leq \tilde{\alpha}_1^{-1}\big(\widetilde{V}_N(n)\big) \leq \tilde{\alpha}_1^{-1}\big(\widetilde{V}_N(0)\big) \leq \tilde{\alpha}_1^{-1}\big(\tilde{\alpha}_2\big(|x_0|_{x_*}\big)\big) =: \alpha_S\big(|x_0|_{x_*}\big). \qquad \square$$

The precise conditions on u_n^j and u_n in Algorithm 7.22 which ensure attractivity, stability and suboptimality estimates now depend on whether stabilizing terminal constraints are used or not. We first consider the case of stabilizing terminal constraints which was investigated, e.g., by Michalska and Mayne [25], Scokaert, Mayne and Rawlings [32] and Rawlings and Mayne [31, Sect. 2.8] which all use conceptually similar ideas. Here, we follow the latter reference.

The approach in [31, Sect. 2.8] can be written as a variant of Theorem 5.13. In particular, we assume that Assumption 5.9 is satisfied. In order to obtain a more convenient notation, on the terminal constraint set $\mathbb{X}_0$ we define a map $\kappa : \mathbb{X}_0 \to \mathbb{U}$ which assigns to each $x \in \mathbb{X}_0$ the control value $u_x \in \mathbb{U}(x)$ from Assumption 5.9(ii). With this notation, the corresponding theorem reads as follows.

Theorem 7.26 *Assume that the conditions of Theorem* 5.13 *are satisfied. Consider Algorithm* 3.10 *with Steps* (2) *and* (3) *replaced by Steps* (2′) *and* (3′) *of Algorithm* 7.22 *under the following assumptions for a set* $S \subseteq \mathbb{X}_N$.

(i) *For* $n = 0$, *we are able to find an admissible initial guess* $u_0^0(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(x_0)$ *for each initial value* $x_0 = x(0) \in S$.
(ii) *For* $n = 1, 2, \ldots,$ *the initial guess* $u_n^0(\cdot)$ *is chosen as* $u_n^0(k) = u_{n-1}(k+1)$, $k = 0, \ldots, N-2$ *and* $u_n^0(N-1) = \kappa(x_{u_n^0}(N-1, x_0))$.
(iii) *For all* $n = 0, 1, 2, \ldots$ *the control sequences* $u_n(\cdot) = u_n^{j^*}(\cdot)$ *satisfy* $u_n^{j^*}(\cdot) \in \mathbb{U}^N_{\mathbb{X}_0}(x_0)$, *i.e., they are admissible.*

Then the NMPC closed loop (2.5) *is attractive on S and the inequality*

$$J_\infty(x, \mu_N) \leq \widetilde{V}_N(0)$$

holds. If, in addition, there exists $\tilde{\alpha}_3 \in \mathcal{K}_\infty$ *such that the inequality* $J_N(x_0, u_0^0(\cdot)) \leq \tilde{\alpha}_3(|x|_{x_*})$ *holds for* $u_0^0(\cdot)$ *from* (i), *then* (2.5) *is also stable on S.*

Proof First note that (i) ensures that u_0^0 is admissible at time $n = 0$ and that (iii) ensures that u_n^0 in (ii) is admissible for $n = 1, 2, \ldots,$ cf. also Lemma 5.10(i).

We abbreviate $x(n) = x_{\mu_n}(n)$. Then, (ii) and the same computation as in the proof of Lemma 5.12 yield the inequality $J_N(x(n+1), u_{n+1}^0(\cdot)) \leq J_{N-1}(x(n+1), u_n(\cdot + 1))$ for each $n \geq 0$. On the other hand, the definition of J_N in Algorithm 3.10 implies

$$\widetilde{V}_N(n) = J_N\big(x(n), u_n(\cdot)\big) = \ell\big(x(n), u_n(0)\big) + J_{N-1}\big(f\big(x(n), u_n(x)\big), u_n(\cdot + 1)\big).$$

The identities $f(x(n), u_n(x)) = x(n+1)$, $u_n(0) = \mu_N(x(n))$ and the inequality $\widetilde{V}_N(n+1) \leq J_N(x(n+1), u_{n+1}^0(\cdot))$ then lead to

$$\widetilde{V}_N(n) \geq \ell\big(x(n), u_n(0)\big) + J_N\big(x_0, u_n^0(\cdot)\big) \geq \ell\big(x(n), \mu_N\big(x(n)\big)\big) + \widetilde{V}_N(n+1),$$

i.e., (7.22). Now all properties follow directly from Proposition 7.25. □

Remark 7.27

(i) If the assumptions of Proposition 5.14(ii) hold, then for $x_0 \in \mathbb{X}_0$, the additional stability condition $J_N(x_0, u_0^0(\cdot)) \leq \tilde{\alpha}_3(|x|_{x_*})$ can be guaranteed if we define $u_0^0(\cdot)$ by $u_0^0(k) := \kappa(x_{u_0^0}(k, x_0))$, $k = 0, \ldots, N-1$. From Assumption 5.9(ii) it follows that this choice implies $J_N(x_0, u_0^0(\cdot)) \leq F(x_0) \leq \tilde{\alpha}_2(|x|_{x_*})$ and thus the desired inequality follows with $\tilde{\alpha}_3 = \tilde{\alpha}_2$. Hence, this choice guarantees stability locally around x_*.

One may also apply this definition to u_n^0 in (ii) for those n in which $x(n) \in \mathbb{X}_0$ holds. This way, stability is ensured at least for the tail of the resulting closed-loop trajectory. If we use this choice of u_n^0 and do not perform the iterative optimization in Step (2′) of Algorithm 7.22, i.e., if we choose $j^* = 0$, then we obtain an algorithm similar to the so-called dual mode strategy from [25].

(ii) Iterative optimizations algorithms are usually designed such that the intermediate results satisfy the desired constraints as soon as the algorithm has succeeded in finding an admissible solution; see Sect. 10.6 for details. Since condition (ii) in Theorem 7.26 ensures that we already initialize the iterative optimization with an admissible solution, most common optimization algorithms will yield solutions $u_n^{j^*}(\cdot)$ satisfying condition (iii) of Theorem 7.26 regardless of how j^* is chosen.

(iii) Theorem 7.26 yields attractivity for arbitrary $j^* \in \mathbb{N}_0$. In particular, it applies to $j^* = 0$, i.e., to the case in which we do not optimize at all. This means that attractivity follows readily from the stabilizing terminal constraints and the particular construction of the initial guesses. An important consequence of this property is that we can fix j^* a priori, e.g., determined by the available computation time, which makes this approach suitable for real-time NMPC schemes.

Without stabilizing terminal constraints, stability is inherited from optimality and we can no longer expect attractivity or stability for arbitrary j^*. Instead, we need to make sure that $u_n^{j^*}$ is at least "good enough" to ensure (7.22). This is the idea of the following algorithm for determining j^* taken from Grüne and Pannek [16].

Algorithm 7.28 Given $\alpha \in (0, 1)$, in Step (2′) of Algorithm 7.22 we iterate over $u_n^j(\cdot) \in \mathbb{U}^N(x(n))$ for $j = 1, 2, \ldots$ until the termination criterion

$$J_N\big(x(n), u_n^{j^*}(\cdot)\big) \leq \widetilde{V}_N(n-1) - \alpha\ell\big(x(n-1), u_{n-1}(0)\big) \tag{7.25}$$

is satisfied.

The following theorem shows attractivity, suboptimality and stability for this algorithm.

Theorem 7.29 *Consider a set $S \subseteq \mathbb{X}_N$, $\alpha \in (0, 1]$ and Algorithm* 3.1 *with Steps* (2) *and* (3) *replaced by Steps* (2′) *and* (3′) *of Algorithm* 7.22. *Assume that Algorithm* 7.28 *is used in Step* (2′) *of Algorithm* 7.22 *and that* (7.25) *is feasible for each $n \in \mathbb{N}$, i.e., that for each $n \in \mathbb{N}$ there exists $u_n^{j^*} \in \mathbb{U}^N(x(n))$ such that* (7.25) *holds. Assume furthermore that* (7.21) *holds for the running cost ℓ.*

Then the NMPC closed loop (2.5) *is attractive on S and the inequality*

$$J_\infty(x, \mu_N) \leq \widetilde{V}_N(0)$$

holds. If, in addition, there exists $\tilde{\alpha}_3 \in \mathcal{K}_\infty$ such that the inequality $J_N(x_0, u_0^0(\cdot)) \leq \tilde{\alpha}_3(|x|_{x_*})$ holds for the initial guess $u_0^0(\cdot)$ in Step (2′) *of Algorithm* 7.22 *for each $x(0) \in S$, then* (2.5) *is also stable on S.*

Proof Under the stated assumptions, all properties follow directly from Proposition 7.25. □

Remark 7.30 In contrast to what was observed in Remark 7.27(iii) for the terminal constrained scheme, here we cannot in general fix j^* a priori. Indeed, the number of iterations of the optimization algorithm which are needed until (7.25) is satisfied depends on various factors—particularly on the choice of u_{n-1} and u_n^0—and is in general unknown before the optimization is started. We assume that for sufficiently small sampling periods similar techniques as developed by Diehl, Findeisen, Allgöwer, Bock and Schlöder [8] or Graichen and Kugi [11] can be used in order to bound the number of needed iterations when setting $u_n^0 = u_{n-1}$, but this has not yet been investigated rigorously.

In the general case, the feasibility assumption for (7.25) in Theorem 7.29 may not even be satisfied. Before we investigate this issue, we illustrate the performance of this algorithm by a numerical example.

Example 7.31 We consider the nonlinear pendulum from Example 2.10, where the task is now to stabilize the downward equilibrium $x_* = (0, 0, 0, 0)^T$. Figures 7.16 and 7.17 below show parts of the closed-loop trajectories of x_1 and x_3 using Algorithm 7.22 and Algorithm 7.28 in Step (2′) for varying α. The running cost is of type (3.4) with

$$L(x, u) = 100 \sin^2(0.5x_1) + x_2^2 + 10.0x_3^2 + x_4^2 + u^2,$$

and sampling period $T = 0.15$ and the NMPC algorithm was run with optimization horizon $N = 17$ and input constraints $\mathbb{U} = [-1, 1]$ using a recursive discretization and a line-search (SQP) method to solve the resulting optimization problem; see Chap. 10 for details on such methods.

One can see clearly from Figs. 7.16 and 7.17 that the closed-loop system is stable for all values of α. Moreover, one can nicely observe the improvement of the closed-loop behavior visible in the decreasing time until the system comes to rest for increasing values of α.

This is also reflected in the total closed-loop costs: While for $\alpha = 0.1$ the costs sum up to $V_\infty^{\tilde{\mu}_N}(x_0) \approx 2512.74$, we obtain a total cost of $V_\infty^{\tilde{\mu}_N}(x_0) \approx 2485.83$ for $\alpha = 0.95$. Note that the majority of the costs, i.e., approximately 2435, is accumulated on the interval $[0, 5]$ on which the trajectories for different α are almost identical and which is therefore not displayed in Figs. 7.16 and 7.17. However, the choice of α has a visible impact on the closed-loop performance in the remaining part of the interval.

Regarding the computational cost, the total number of (SQP) steps which are executed during the run of the NMPC procedure reduces from 455 for $\alpha = 0.95$ and 407 for $\alpha = 0.9$, to 267 and 246 for $\alpha = 0.5$ and $\alpha = 0.1$, respectively. Hence, we obtain an average of approximately 2.5–4.5 optimization iterations per MPC step over the entire interval $[0, 15]$, while using standard termination criteria 9.5 optimization iterations per NMPC step are required.

A closer look at the numerical simulation in this example reveals that for each α there were some sampling instants n at which it was not possible to satisfy the

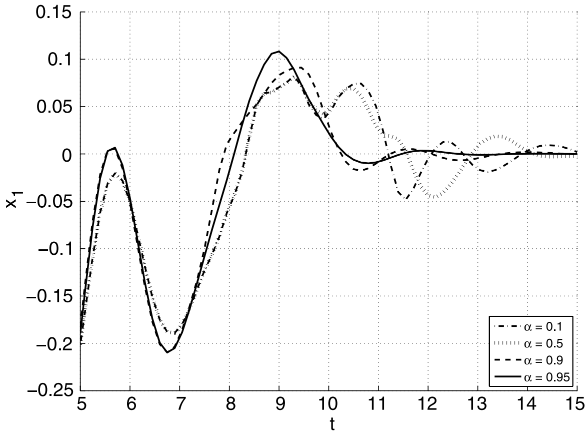


Fig. 7.16 Angle of the pendulum x_1 for varying α

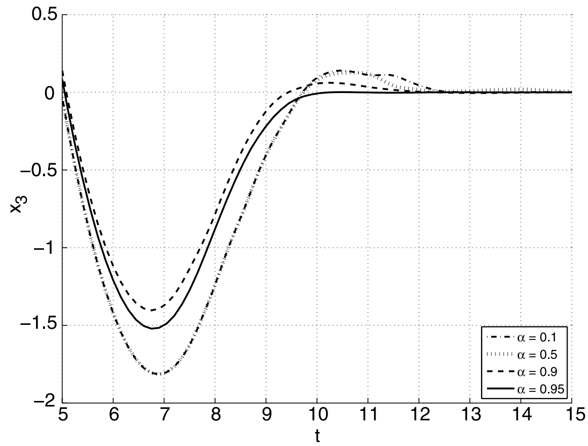


Fig. 7.17 Position of the cart x_3 for varying α

suboptimality based termination criterion (7.25). In this case we simply iterated the SQP optimization routine until convergence.

While this fact is not visible in Figs. 7.16 and 7.17 and obviously does not affect stability and performance in our example, this observation raises the question whether (7.25) is feasible, i.e., whether at time n we can ensure the existence of $u_n^{j^*}$ such that (7.25) is satisfied regardless of how u_{n-1} was chosen, before. In order to analyze this question, let us suppose that Assumption 6.4 holds. Then, observing that for optimal controls (7.25) coincides with (5.1), Theorem 6.14 yields that (7.25) is feasible if u_{n-1} is an optimal control sequence and α in (7.25) is smaller than α from (6.14). However, even with this choice of α in (7.25), condition (7.25) may not be feasible for nonoptimal control sequences u_{n-1}.

In order to understand why this is the case we investigate how Proposition 6.12—which provides the crucial ingredient for deriving (6.14)—changes if the optimal control sequence $u^\star$ in this proposition is replaced by a nonoptimal control sequence u_{n-1}. To this end, we fix $n \in \mathbb{N}$ and set $x = x_{\mu_N}(n)$ and $u = u_{n-1}$. Now, first observe that the inequalities in (6.12) remain valid regardless of the optimality of $u^\star$. All inequalities in (6.11), however, require optimality of the control sequence $u^\star$ generating the λ_n. In order to maintain at least some of these inequalities we can pick an optimal control sequence $u^\star$ for initial value $x_u(1,x)$ and horizon length $N-1$ and define a control sequence $\tilde{u}$ via $\tilde{u}(0)=u(0)$, $\tilde{u}(n)=u^\star(n-1)$, $n=1,\dots,N-1$. Then, abbreviating

$$\begin{aligned} \tilde{\lambda}_n &= \ell\bigl(x_{\tilde{u}}(n,x),\tilde{u}(n)\bigr), \quad n=0,\dots,N-1 \quad \text{and} \\ \tilde{\nu} &= V_N\bigl(x_u(1,x)\bigr)=V_N\bigl(x_{\tilde{u}}(1,x)\bigr), \end{aligned} \tag{7.26}$$

we arrive at the following version of Proposition 6.12.

Proposition 7.32 *Let Assumption* 6.4 *hold. Then the inequalities*

$$\sum_{n=k}^{N-1}\tilde{\lambda}_n \le B_{N-k}(\tilde{\lambda}_k) \quad \textit{and} \quad \tilde{\nu}\le\sum_{n=0}^{j-1}\tilde{\lambda}_{n+1}+B_{N-j}(\tilde{\lambda}_{j+1}) \tag{7.27}$$

hold for $k=1,\dots,N-2$ *and* $j=0,\dots,N-2$.

Proof Analogous to the proof of Proposition 6.12. □

The subtle but crucial difference of (7.27) to (6.11), (6.12) is that the left inequality in (7.27) is not valid for $k=0$. As a consequence, $\tilde{\lambda}_0$ does not appear in any of the inequalities, thus for any $\tilde{\lambda}_1,\dots,\tilde{\lambda}_n$ and $\tilde{\nu}$ satisfying (7.27) and any $\delta>0$ the values $\delta\tilde{\lambda}_1,\dots,\delta\tilde{\lambda}_n$ and $\delta\tilde{\nu}$ satisfy (7.27), too. Hence, unless (7.27) implies $\tilde{\nu}\le\sum_{n=0}^{N-1}\tilde{\lambda}_n$—which is a very particular case—replacing (6.11), (6.12) in (6.14) by (7.27) will lead to the optimal value $\alpha=-\infty$. Consequently, feasibility of (7.25) cannot be concluded for any positive α.

The following example shows that this undesirable result is not simply due to an insufficient estimate for α but that infeasibility of (7.25) can indeed happen.

Example 7.33 Consider the 1d system

$$x^+ = x/2+u \tag{7.28}$$

with $\ell(x,u)=|x|$, input constraint $u\ge 0$ and optimization horizon $N=3$. A simple computation using $u_x\equiv 0$ shows that for this system Assumption 6.4 is satisfied with $\beta(r,k)=C\sigma^k r$ with $C=1$ and $\sigma=1/2$. Hence, Corollary 6.19 applies and we can use (6.19) in order to compute that for $N=3$ Inequality (5.1) holds for $\alpha=7/8$. If u_{n-1} in the termination criterion (7.25) is chosen as the optimal control $u^\star$, then (7.25) implies that (5.1) is feasible for this α.

For $x(n-1)=0$, it is obvious that the control $u^\star\equiv 0$ is optimal. Using the nonoptimal control given by $u_{n-1}(0)=\varepsilon>0$ and $u_{n-1}(1)=u_{n-1}(2)=0$ yields the trajectory $x_{u_{n-1}}(0)=x(n-1)=0$, $x_{u_{n-1}}(k)=\varepsilon 2^{-k+1}$, $k=1,2$, which implies $x(n)=\varepsilon$ and

$$J_3\bigl(x(n-1),u_{n-1}\bigr)=\sum_{k=0}^{1}\varepsilon 2^{-k}=3\varepsilon/2.$$

On the other hand, for the initial value $x(n) = \varepsilon$ it is easily seen that for each control u_n the inequality

$$J_3\big(x(n), u_n\big) \geq \sum_{k=0}^{2} \varepsilon 2^{-k} = 7\varepsilon/4 > 3\varepsilon/2 = J_N\big(x(n-1), u_{n-1}\big)$$

holds. Hence, for this choice of u_{n-1} the Inequality (7.25) is not feasible for any $\alpha > 0$.

Clearly, in order to rigorously ensure attraction and guaranteed performance one should derive conditions which exclude these situations and we briefly discuss two possible approaches for this purpose.

One way to guarantee feasibility of (7.25) is to add the missing inequality in (7.27) (i.e., the left inequality for $k = 0$) as an additional constraint in the optimization. This guarantees feasibility of (7.25) for any α smaller than the value from (6.19). One drawback of this approach is that—similar to the terminal constraint case—an additional constraint in the optimization is needed which needs to be ensured for all $j \geq 1$ or at least for j^*. This makes the optimization more demanding, since in contrast to Remark 7.27(ii) here we do not have a canonical candidate for an admissible solution which can be used for initializing the iterative optimization. Another drawback is that the value $B_N(\tilde{\lambda}_0)$ depends on the in general unknown function β from Assumption 6.4 and thus needs to be determined either by an a priori analysis or by a try-and-error procedure.

Another way to guarantee feasibility is to choose ℓ in such a way that there exists $\gamma > 0$ for which

$$\gamma\ell(x,u) \geq \ell^*\big(f(x,u)\big) \tag{7.29}$$

holds for all $x \in X$ and all $u \in U$. Then from (7.29) and the controllability Assumption 6.4 for $x = f(x(n-1), \tilde{u}_{n-1}(0))$ we get

$$\sum_{k=0}^{N-1} \tilde{\lambda}_k \leq \tilde{\lambda}_0 + B_{N-1}\Big(\ell^*\Big(f\big(x(n-1), \tilde{u}_{n-1}(0)\big)\Big)\Big) \leq \tilde{\lambda}_0 + B_{N-1}(\gamma\tilde{\lambda}_0).$$

Replacing $\beta(r,0)$ by $\max\{\beta(r,t), \tilde{\beta}(r,t)\}$ with $\tilde{\beta}(r,0) = \beta(\gamma r, 0) + r$ and $\tilde{\beta}(r,k) = \beta(\gamma r, k)$ for $k \geq 1$, this right hand side is $\leq B_N(\tilde{\lambda}_0)$ which again yields the left inequality in (7.27) for $k = 0$ and thus feasibility of (7.25). Note that (7.29) holds for our example (7.28) if we change $\ell(x,u) = |x|$ to $\ell(x,u) = |x| + |u|/\gamma$. For this ℓ and the points and control sequences considered in the example, we obtain

$$J_3\big(x(n-1), u_{n-1}\big) = 3\varepsilon/2 + \varepsilon = 5\varepsilon/2$$

from which one computes that (7.25) is now feasible.

The advantage of this method is that no additional constraints have to be imposed in the optimization. Its disadvantages are that constructing ℓ satisfying (7.29) may be complicated for more involved dynamics and that the overshoot encoded in β will in general increase for the re-designed ℓ. As outlined in Sect. 6.6, this may lower the

NMPC closed-loop performance and cause the need for larger optimization horizons N in order to obtain stability.

An in depth study of these approaches and in particular their algorithmic implementation and numerical evaluation will be the topic of further research.

7.10 Beyond Stabilization and Tracking

All NMPC variants discussed so far have in common that the cost function ℓ penalizes the distance to some desired reference, either to an equilibrium x_* or to a time varying reference x^{ref}. These variants may hence be called *stabilizing NMPC*. There is, however, a large variety of optimal control problems where this is not the case. For instance, in economic applications one typically uses a running cost ℓ_e which reflects an economic cost rather than a distance to some reference, cf., e.g., Seierstad and Sydsæter [33]. In what follows we will refer to ℓ_e as the *economic* cost. In such problems, the desired limit behavior of the optimal trajectories is not given a priori in terms of a reference x_* or x^{ref} but is rather an outcome of the optimization itself. Even for rather simple nonlinear models, this limit behavior can be surprisingly complex, as, e.g., the examples in the book of Grass, Caulkins, Feichtinger, Tragler and Behrens [12]—for optimal control problems mainly motivated by social sciences—show.

One way to use stabilizing NMPC for such problems is as follows. In a first step, the optimal limit behavior for the economic running cost ℓ_e is identified. Assuming that this problem can be solved analytically or numerically we obtain an optimal reference solution x^{ref} which, however, does not need to be asymptotically stable. Hence, a stabilizing controller needs to be designed in order to stabilize the optimal reference. To this end, in a second step a cost function ℓ—which we will refer to as *stabilizing* cost—penalizing the distance to x^{ref} is designed which is suitable for running a stabilizing NMPC scheme in order to obtain a stable closed loop.

Proceeding this way guarantees asymptotic stability of the optimal equilibrium (e.g., under the various conditions on f, ℓ and the particular NMPC scheme discussed in this book) but the resulting closed-loop trajectories based on the optimization of the stabilizing cost ℓ may be very different from the optimal trajectories using the economic cost ℓ_e. In particular, they may be far from optimal when performance is measured via the economic cost function ℓ_e.

Due to the fact that for running the NMPC Algorithms 3.1 and its variants no particular conditions on ℓ are needed, it is a natural idea to try to run these algorithms using the economic cost ℓ_e in (OCP_N) and its variants instead of taking the detour via the stabilizing cost function ℓ. Formally, most usual NMPC algorithms (in particular those discussed in this book) are perfectly suited for doing so, however, the theoretical results ensuring stability and performance are in general not applicable, because the economic cost ℓ_e will not satisfy the conditions needed for these results. Hence, new conditions for ensuring stability and performance are needed.

Here we summarize some recent results in this direction. In [3] (see also the references in this paper for earlier research on this subject), Angeli, Amrit and Rawlings observe that if one adds the optimal limit behavior as a terminal constraint to the NMPC scheme, then performance estimates for the NMPC closed loop can be given. More precisely, assume that the optimal control problem exhibits an optimal equilibrium x_* with related control value u_*, i.e., $f(x_*, u_*) = x_*$ holds and $\ell_e(x_*, u_*)$ is minimal among all possible equilibria. Then, using the NMPC scheme from Sect. 5.2 with $\ell = \ell_e$ and $\mathbb{X}_0 = \{x_*\}$, for each $x \in \mathbb{X}_N$ one obtains the performance estimate

$$\overline{J}_\infty(x, \mu_N) \le \ell_e(x_*, u_*), \tag{7.30}$$

where $\overline{J}_\infty$ denotes the averaged infinite horizon cost functional

$$\overline{J}_\infty(x_0, \mu_N) := \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K} \ell_e\big(x_{\mu_N}(k, x_0), \mu\big(x_{\mu_N}(k)\big)\big). \tag{7.31}$$

Observe that $\overline{J}_\infty(x_0, \mu_N)$ is not simply $J_\infty(x_0, \mu_N)$ from (4.10) with ℓ replaced by ℓ_e. The important difference between J_∞ and $\overline{J}_\infty$ is that $\overline{J}_\infty$ contains the additional averaging term $1/K$. This term is necessary since in general for economic running costs ℓ_e we cannot expect the infinite sum in (4.10) to converge. This approach can be extended to periodic optimal trajectories x^{ref} instead of equilibria by using suitable periodic terminal constraint sets; for details see [3].

It is interesting to note that—at least in the case of an optimal equilibrium x_* with control value u_*—the estimate (7.30) may also hold for controllers μ_N from stabilizing NMPC schemes. To this end, we use a stabilizing running cost ℓ satisfying

$$\ell(x, u) \ge \alpha_1\big(|x|_{x_*} + |u|_{u_*}\big) \tag{7.32}$$

for some $\alpha_1 \in \mathcal{K}_\infty$ and assume that $J_\infty(x_0, \mu_N)$ is finite and that the economic cost ℓ_e is continuous. Then, since $J_\infty(x_0, \mu_N)$ is finite, $\ell_e(x_{\mu_N}(n), \mu_N(x_{\mu_N}(n)))$ converges to 0 as $n \to \infty$ and hence the lower bound (7.32) implies $x_{\mu_N}(n) \to x_*$ and $\mu_N(x_{\mu_N}(n)) \to u_*$ as $n \to \infty$. This, in turn, implies $\ell_e(x_{\mu_N}(n), \mu_N(x_{\mu_N}(n))) \to \ell_e(x_*, u_*)$ as $n \to \infty$ from which (7.30) follows. Hence, although it seems reasonable to expect that for NMPC with economic running cost ℓ_e one obtains a better performance of the closed-loop trajectories in terms of the economic objective ℓ_e, this is not reflected in the asymptotic estimate (7.30).

In the usual NMPC setting, a finite value of $J_\infty(x_0, \mu_N)$ from (4.10) together with positive definiteness of ℓ allows one to conclude that the closed-loop trajectory must converge to x_*, because otherwise $J_\infty(x_0, \mu_N)$ would be unbounded. This is not the case for the averaged functional $\overline{J}_\infty(x_0, \mu)$ from (7.31) and, indeed, one needs additional conditions in order to ensure that the closed-loop solution satisfying (7.30) does converge to x_*. Such a condition has been presented in Diehl, Amrit and Rawlings [7] for the case of an optimal steady state and finite-dimensional state space $X = \mathbb{R}^d$. The condition, called strong duality, demands the existence of a value $\lambda_* \in \mathbb{R}^d$ such that x_* and u_* minimize the expression

$$\ell_e(x, u) + \big[x - f(x, u)\big]^T \lambda_*$$

over all admissible states $x \in \mathbb{X}$ and control values $u \in \mathbb{U}(x)$. Furthermore, the existence of $\alpha_1 \in \mathcal{K}_\infty$ with

$$\ell_e(x,u) + \left[x - f(x,u)\right]^T \lambda_* - \ell_e(x_*,u_*) \geq \alpha_1\left(|x|_{x_*}\right)$$

is required. Under these conditions, a Lyapunov function can be constructed by adding suitable correction terms to the finite horizon optimal value function V_N (corresponding to the economic running cost ℓ_e). In [2], Angeli and Rawlings further observed that strong duality can be interpreted as a dissipativity condition, which links this condition to more classical concepts used in the stability analysis of control systems.

Summarizing, the results sketched in this section show that NMPC can be used for obtaining optimal feedback controllers also for optimal control problems different from the classical NMPC objectives stabilization and tracking. We conjecture that NMPC will prove valuable also for other types of optimization criteria, however, we are also convinced that there are problems which are not solvable using the receding horizon NMPC paradigm. An in depth analysis of the structural properties an optimal control problem needs to exhibit in order to be tractable with NMPC techniques would certainly be an interesting research project.

References

1. Alamir, M.: Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes. Lecture Notes in Control and Information Sciences, vol. 339. Springer, London (2006)
2. Angeli, D., Rawlings, J.B.: Receding horizon cost optimization and control for nonlinear plants. In: Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems – NOLCOS 2010, Bologna, Italy, pp. 1217–1223 (2010)
3. Angeli, D., Amrit, R., Rawlings, J.B.: Receding horizon cost optimization for overly constrained nonlinear plants. In: Proceedings of the 48th IEEE Conference on Decision and Control – CDC 2009, Shanghai, China, pp. 7972–7977 (2009)
4. Chen, W., Ballance, D.J., O'Reilly, J.: Model predictive control of nonlinear systems: Computational delay and stability. IEE Proc., Control Theory Appl. **147**(4), 387–394 (2000)
5. De Nicolao, G., Magni, L., Scattolini, R.: Stability and robustness of nonlinear receding horizon control. In: Nonlinear Predictive Control, pp. 3–23. Birkhäuser, Basel (2000)
6. Di Palma, F., Magni, L.: On optimality of nonlinear model predictive control. Systems Control Lett. **56**(1), 58–61 (2007)
7. Diehl, M., Amrit, R., Rawlings, J.B.: A Lyapunov function for economic optimizing model predictive control. IEEE Trans. Autom. Control (2010, in press). doi:10.1109/TAC.2010.2101291
8. Diehl, M., Findeisen, R., Allgöwer, F., Bock, H.G., Schlöder, J.P.: Nominal stability of the real-time iteration scheme for nonlinear model predictive control. IEE Proc., Control Theory Appl. **152**, 296–308 (2005)
9. Findeisen, R.: Nonlinear model predictive control: A sampled-data feedback perspective. PhD thesis, University of Stuttgart, VDI-Verlag, Düsseldorf (2004)
10. Findeisen, R., Allgöwer, F.: Computational delay in nonlinear model predictive control. In: Proceedings of the International Symposium on Advanced Control of Chemical Processes, Hong Kong, China, 2003. Paper No. 561
11. Graichen, K., Kugi, A.: Stability and incremental improvement of suboptimal MPC without terminal constraints. IEEE Trans. Automat. Control **55**, 2576–2580 (2010)

12. Grass, D., Caulkins, J.P., Feichtinger, G., Tragler, G., Behrens, D.A.: Optimal Control of Nonlinear Processes. With Applications in Drugs, Corruption, and Terror. Springer, Berlin (2008)
13. Grimm, G., Messina, M.J., Tuna, S.E., Teel, A.R.: Model predictive control: for want of a local control Lyapunov function, all is not lost. IEEE Trans. Automat. Control **50**(5), 546–558 (2005)
14. Grüne, L.: Worst case vs. average performance estimates for unconstrained NMPC schemes. PAMM **10**, 607–608 (2010)
15. Grüne, L., Pannek, J.: Practical NMPC suboptimality estimates along trajectories. Systems Control Lett. **58**(3), 161–168 (2009)
16. Grüne, L., Pannek, J.: Analysis of unconstrained NMPC schemes with incomplete optimization. In: Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems – NOLCOS 2010, Bologna, Italy, pp. 238–243 (2010)
17. Grüne, L., Rantzer, A.: On the infinite horizon performance of receding horizon controllers. IEEE Trans. Automat. Control **53**, 2100–2111 (2008)
18. Grüne, L., Pannek, J., Worthmann, K.: A networked unconstrained nonlinear MPC scheme. In: Proceedings of the European Control Conference – ECC 2009, Budapest, Hungary, pp. 91–96 (2009)
19. Grüne, L., Pannek, J., Worthmann, K.: A prediction based control scheme for networked systems with delays and packet dropouts. In: Proceedings of the 48th IEEE Conference on Decision and Control – CDC 2009, Shanghai, China, pp. 537–542 (2009)
20. Grüne, L., Pannek, J., Seehafer, M., Worthmann, K.: Analysis of unconstrained nonlinear MPC schemes with varying control horizon. SIAM J. Control Optim. **48**, 4938–4962 (2010)
21. Grüne, L., von Lossow, M., Pannek, J., Worthmann, K.: MPC: implications of a growth condition on exponentially controllable systems. In: Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems – NOLCOS 2010, Bologna, Italy, pp. 385–390 (2010)
22. Jadbabaie, A., Hauser, J.: On the stability of receding horizon control with a general terminal cost. IEEE Trans. Automat. Control **50**(5), 674–678 (2005)
23. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice Hall, Upper Saddle River (2002)
24. Limón, D., Alamo, T., Salas, F., Camacho, E.F.: On the stability of constrained MPC without terminal constraint. IEEE Trans. Automat. Control **51**(5), 832–836 (2006)
25. Michalska, H., Mayne, D.Q.: Robust receding horizon control of constrained nonlinear systems. IEEE Trans. Automat. Control **38**(11), 1623–1633 (1993)
26. Nešić, D., Teel, A.R.: A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. IEEE Trans. Automat. Control **49**(7), 1103–1122 (2004)
27. Nešić, D., Grüne, L.: A receding horizon control approach to sampled-data implementation of continuous-time controllers. Systems Control Lett. **55**, 660–672 (2006)
28. de Oliveira Kothare, S.L., Morari, M.: Contractive model predictive control for constrained nonlinear systems. IEEE Trans. Automat. Control **45**(6), 1053–1071 (2000)
29. Pannek, J.: Receding horizon control: A suboptimality-based approach. PhD thesis, University of Bayreuth, Germany (2009)
30. Parisini, T., Zoppoli, R.: A receding-horizon regulator for nonlinear systems and a neural approximation. Automatica **31**(10), 1443–1451 (1995)
31. Rawlings, J.B., Mayne, D.Q.: Model Predictive Control: Theory and Design. Nob Hill Publishing, Madison (2009)
32. Scokaert, P.O.M., Mayne, D.Q., Rawlings, J.B.: Suboptimal model predictive control (feasibility implies stability). IEEE Trans. Automat. Control **44**(3), 648–654 (1999)
33. Seierstad, A., Sydsæter, K.: Optimal Control Theory with Economic Applications. North-Holland, Amsterdam (1987)
34. Varutti, P., Findeisen, R.: Compensating network delays and information loss by predictive control methods. In: Proceedings of the European Control Conference – ECC 2009, Budapest, Hungary, pp. 1722–1727 (2009)
35. Zavala, V.M., Biegler, L.T.: The advanced-step NMPC controller: optimality, stability and robustness. Automatica **45**(1), 86–93 (2009)