Chapter 11
Numerical Discretization

This chapter is particularly devoted to sampled data systems, which need to be discretized in order to be able to solve the optimal control problem within the NMPC algorithm numerically. We present suitable methods, discuss the convergence theory for one step methods and give an introduction into step size control algorithms. Furthermore,we explain how these methods can be integrated into NMPC algorithms, investigate how the numerical errors affect the stability of the NMPC controller derived from the numerical model and show which kind of robustness is needed in order to ensure a practical kind of stability.

11.1 Basic Solution Methods

In order to define the setting, we start by summarizing the main concepts from Sect. 2.2. As already mentioned there, in most applications the discrete time system (2.1) is obtained from sampling a continuous time system

$$\dot{x}(t) = f_c(x(t), v(t)) \tag{2.6}$$

with $x(t) \in \mathbb{R}^d$ and $v(t) \in \mathbb{R}^m$. More precisely, given a subset $U \subseteq L^\infty([0, T], \mathbb{R}^m)$, i.e., each $u \in U$ is a continuous time control function defined on the sampling interval $[0, T]$, we define the discrete time dynamics f in (2.1) by

$$x^+ = f(x, u) := \varphi(T, 0, x, u) \tag{2.8}$$

where $\varphi(T, 0, x, u)$ is the solution of (2.6) with $v = u$ satisfying the initial condition $\varphi(0, 0, x, u) = x$. Here, we tacitly assume that for all admissible initial values $x \in \mathbb{X} \subseteq X = \mathbb{R}^d$ and all admissible controls $u \in \mathbb{U}(x) \subseteq U$ the solution $\varphi(t, 0, x, u)$

L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*,
Communications and Control Engineering, DOI 10.1007/978-3-319-46024-6_11

exists for $t \in [0, T]$. This way we obtain a discrete time system (2.1) whose solutions for each control sequence $u(\cdot) \in \mathbb{U}^N(x)$ satisfy

$$\varphi(t_n, t_0, x_0, v) = x_u(n, x_0), \quad n = 0, 1, 2, \dots, N, \tag{2.7}$$

for all sampling times $t_n = nT$, $n = 0, \dots, N$, and the continuous time control function v given by

$$v(t) = u(n)(t - t_n) \quad \text{for almost all } t \in [t_n, t_{n+1}] \text{ and all } n = 0, \dots, N-1, \tag{2.13}$$

cf. Theorem 2.7.

Since a closed formula for f defined in (2.8) will only be available in exceptional cases, it is in general necessary to use numerical schemes in order to compute a numerical approximation of f. This way, instead of an analytical formula we obtain an algorithm which can be used in order to compute the predictions needed in the optimal control problem and its variants. For the exposition in this chapter we restrict ourselves to sampling with zero order hold in which each element $u(n)$ of the control sequence is a constant function from $[0, T]$ to $\mathbb{R}^m$. This amounts to defining

$$U := \{u : [0, T] \to \mathbb{R}^m \mid \text{there exists } u_0 \in \mathbb{R}^m \text{ with } u(t) = u_0 \text{ for all } t \in [0, T]\}.$$

Observe that each element in U is uniquely defined by the value $u_0 \in \mathbb{R}^m$. Accordingly, we identify U with $\mathbb{R}^m$ and regard each $u \in U$ as a value in $\mathbb{R}^m$. Henceforth, we will again use the symbol u (instead of u_0) for this value. The resulting continuous time control function v in (2.7) is then piecewise constant on the sampling intervals, cf. also Fig. 2.3 and the discussion after Theorem 2.7. Recall from Remark 2.8 that the overlap of the sampling intervals at the sampling times t_n does not pose a problem in the definition of v in (2.13).

In the following, we give an introduction into numerical methods for ordinary differential equations and their analysis. In particular, we give details on so-called one step methods and show convergence results and requirements. Moreover, we sketch the basic idea of the very useful step size control algorithms. These algorithms allow us to externally define an error tolerance level for the solution and produce an adaptive time grid which is computationally much more efficient than using a sufficiently fine uniform grid, a requirement that is frequently found in the sampled data literature. For references to textbooks which cover the material presented here more comprehensively and in more detail we refer to Sect. 11.6.

For computing f in (2.8) it is sufficient to solve (2.6) on the interval $[0, T]$ on which $u \in U$ in (2.8) is constant. Hence, the right-hand side in (2.8) does not depend on t and—more importantly—does not exhibit discontinuities on the interval $[0, T]$. For this reason, standard numerical techniques can be applied. Still, the solution depends on the constant control value u which will be reflected in the subsequent notation.

Before we can develop solution methods for ordinary differential equations, we need to define some general concepts. As we have pointed out before, the fundamental idea of almost all numerical solution methods is to replace the analytic solution $\varphi(t, 0, x, u)$ for $t \in [0, T]$ by an approximation. Throughout the rest of this chapter, we denote this approximation by $\tilde{\varphi}(t, 0, x, u)$. The following definition states for which t such an app roximation is defined and what convergence of such an approximation means.

Definition 11.1 (i) A set $\mathscr{G} = \{\tau_0, \tau_1, \dots, \tau_M\}$ of time instants with $0 = \tau_0 < \tau_1 < \dots < \tau_M = T$ is called a *time grid* on the interval $[0, T]$. The values $h_i := \tau_{i+1} - \tau_i$ and $\overline{h} := \max_{i=0,\dots,M-1} h_i$ are called *step sizes* and *maximal step size*, respectively.

(ii) A function $\tilde{\varphi} : \mathscr{G} \times \mathscr{G} \times \mathbb{R}^d \times U \to \mathbb{R}^d$ is called *grid function*.

(iii) Assume that the solution $\varphi(t; \tau_0, x_0, u)$ of (2.6) exists for $t \in [0, T]$. Then a family of grid functions $\tilde{\varphi}_j$, $j \in \mathbb{N}$, on time grids $\mathscr{G}_j$ on the interval $[0, T]$ with maximal step sizes $\overline{h}_j$ is called *(discrete) approximation* of $\varphi(t; \tau_0, x_0, u)$ (2.6), if it is *convergent*, i.e.,

$$\max_{\tau_i \in \mathscr{G}_j} \|\tilde{\varphi}_j(\tau_i; \tau_0, x_0, u) - \varphi(\tau_i; \tau_0, x_0, u)\| \to 0 \quad \text{as } \overline{h}_j \to 0.$$

The convergence of the approximation is said to be of *order* $p > 0$ if for all compact sets $K \subset \mathbb{R}^d$, $Q \subset U$ there exists a constant $M > 0$ such that

$$\max_{\tau_i \in \mathscr{G}_j} \|\tilde{\varphi}_j(\tau_i; \tau_0, x_0, u) - \varphi(\tau_i; \tau_0, x_0, u)\| \leq M\overline{h}_j^p \tag{11.1}$$

holds for all $x_0 \in K$, all $u \in Q$ and all sufficiently fine grids $\mathscr{G}_j$ on $[0, T]$.

Less technically speaking, an approximation $\tilde{\varphi}(\tau_i, 0, x, u)$ is a grid function defined on $\mathscr{G}$ which approximates the values of the true solution at the grid points and becomes the more accurate the finer the grid becomes. Moreover, the larger the order of convergence p is, the faster the approximation will converge toward the exact solution for $\overline{h} \to 0$.

The most simple class of numerical methods to compute a discrete approximation satisfying Definition 11.1 are so-called one step methods. Although simple to design, these methods are nonetheless well suited even for rather complicated problems. One step methods compute the grid function $\tilde{\varphi}$ iteratively via

$$\tilde{\varphi}(\tau_0; \tau_0, x_0, u) := x_0, \quad \tilde{\varphi}(\tau_{i+1}; \tau_0, x_0, u) := \Phi(\tilde{\varphi}(\tau_i; \tau_0, x_0, u), u, h_i) \tag{11.2}$$

for $i = 0, \dots, M - 1$ starting from the given initial value x_0. Here Φ is a mapping

$$\Phi : \mathbb{R}^d \times \mathbb{U} \times \mathbb{R} \to \mathbb{R}^d$$

which should be easy to implement and cheap to evaluate on a computer and, of course, provide a convergent approximation in the sense of Definition 11.1(iii).

In order to design such a map Φ, we use that the solution of the differential equation (2.6) for two consecutive grid points τ_i and τ_{i+1} satisfies the integral equation

$$\varphi(\tau_{i+1}; \tau_0, x_0, u) = \varphi(\tau_i; \tau_0, x_0, u) + \int\limits_{\tau_i}^{\tau_{i+1}} f_c(\varphi(t; \tau_0, x_0, u), u)dt.$$

Approximating the integral expression by the rectangle rule we obtain

$$\int\limits_{\tau_i}^{\tau_{i+1}} f_c(\varphi(t; \tau_0, x_0, u), u)dt \approx (\tau_{i+1} - \tau_i) f_c(\varphi(\tau_i; \tau_0, x_0, u), u) = h_i f_c(\varphi(\tau_i; \tau_0, x_0, u), u)$$

Inserting this approximation into the above integral equation then yields

$$\varphi(\tau_{i+1}; \tau_0, x_0, u) \approx \varphi(\tau_i; \tau_0, x_0, u) + h_i f_c(\varphi(\tau_i; \tau_0, x_0, u), u).$$

Now we define an approximate solution $\tilde{\varphi}$ by requiring that it exactly solves this approximate equation, i.e.,

$$\tilde{\varphi}(\tau_{i+1}; \tau_0, x_0, u) = \tilde{\varphi}(\tau_i; \tau_0, x_0, u) + h_i f_c(\tilde{\varphi}(t; 0, x_0, u), u). \qquad (11.3)$$

This is exactly the iteration in (11.2) with

$$\Phi(x, u, h) := x + hf_c(x, u).$$

This one step method is called the Euler scheme. Now, if we assume $\tilde{\varphi}(\tau_i; \tau_0, x_0, u) \approx \varphi(\tau_i; \tau_0, x_0, u)$, then we see that

$$\begin{aligned}\tilde{\varphi}(\tau_{i+1}; \tau_0, x_0, u) &\approx \varphi(\tau_i; 0, x_0, u) + h_i f_c(\varphi(\tau_i; \tau_0, x_0, u), u)\\ &\approx \varphi(\tau_i; \tau_0, x_0, u) + \int\limits_{\tau_i}^{\tau_{i+1}} f_c(\varphi(t; \tau_0, x_0, u), u)dt = \varphi(\tau_{i+1}; \tau_0, x_0, u)\end{aligned}$$

which suggests that this method yields an approximation in the sense of Definition 11.1(iii). Formally, we will prove this property for general one step methods in Theorem 11.5 below. Before we turn to the convergence analysis, we present an important class of solution methods which follow from a generalization of the Euler approximation idea to solve the integral equation.

The idea to generalize the Euler method is to use a higher order approximation for the integral. For example, one can approximate the integral by the trapezoidal rule instead of the rectangle rule, which leads to the approximation

$$\begin{aligned}\varphi(\tau_{i+1};\tau_0,x_0,u) &\approx \varphi(\tau_i;\tau_0,x_0,u)\\ &\quad + \frac{h_i}{2}\Big(f_c(\varphi(\tau_i;\tau_0,x_0,u),u)+f_c(\varphi(\tau_{i+1};\tau_0,x_0,u),u)\Big).\end{aligned}$$

When trying to use this approximation in order to define $\tilde{\varphi}$ analogous to (11.3), above, we run into the problem that the unknown value $\tilde{\varphi}(\tau_{i+1};\tau_0,x_0,u)$ appears on the right hand side. We can avoid this if we use the Euler scheme in order to approximate

$$f_c(\varphi(\tau_{i+1};\tau_0,x_0,u),u) \approx f_c(\varphi(\tau_i;\tau_0,x_0,u)+h_i f_c(\varphi(\tau_i;\tau_0,x_0,u))).$$

Proceeding this way we end up with the so-called Heun method

$$\Phi(x,v,h) := x + \frac{h}{2}\big(f_c(x,u)+f_c(x+hf_c(x,u),u)\big).$$

Observe that in this formula the value $f_c(x,u)$ appears twice and that the scheme uses nested evaluations of the vector field f_c. The formalism of Runge–Kutta methods now gives a systematic way to formalize this nested structure. We first illustrate this formalism using the Heun method, for which it reads

$$\begin{aligned}k_1 &:= f_c(x,u)\\ k_2 &:= f_c(x+hk_1,u)\\ \Phi(x,u,h) &:= x + h\left(\frac{1}{2}k_1+\frac{1}{2}k_2\right)\end{aligned}$$

The advantage of this formalism is that one can easily add new function evaluations or modify the weighted combination. This leads to the following general form.

Definition 11.2 An s-stage (explicit) Runge–Kutta method is given by

$$\begin{aligned}k_i &:= f\left(x + h\sum_{j=1}^{i-1} a_{ij}k_j\right) \quad \text{for } i=1,\dots,s\\ \Phi(x,u,h) &:= x + h\sum_{i=1}^{s} b_i k_i.\end{aligned}$$

The value $k_i = k_i(x,u,h)$ is called the ith stage of the method.

The methods thus defined depend on the parameters a_{ij} and b_i. If the vector field explicitly depends on t—which is not the case in our setting—then additional

Table 11.1 Butcher tableaus for the Euler, Heun, and classical Runge–Kutta method (left to right)

$$
\begin{array}{c|c}
0 & \\
\hline
 & 1
\end{array}
\qquad
\begin{array}{c|cc}
0 & & \\
1 & 1 & \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}
\qquad
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\hline
 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6}
\end{array}
$$

parameters c_i are used in the definition. More compactly, these parameters are written as so-called *Butcher tableaus* of the form

$$
\begin{array}{c|ccccc}
c_1 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s\,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

Table 11.1 shows Butcher tableaus corresponding to the Euler scheme (left), the Heun scheme (middle) and the so-called classical Runge–Kutta scheme with $s = 4$ stages proposed by Carl Runge and Martin Kutta in 1895 (right).

Remark 11.3 Models based on partial differential equations, like the one discussed in Example 6.32, require discretization techniques different from the one discussed here. In particular, apart from the discretization in time also a discretization in space has to be performed. Popular techniques for this purpose are finite difference or finite element methods and the interested reader is referred to the large amount of textbooks on this topic, like, e.g., the books by LeVeque [9] or Braess [1] respectively.

11.2 Convergence Theory

Having defined one step methods we now show that the resulting approximations actually converge toward the solution. To this end, we define the error at time $\tau_i \in \mathscr{G}$ as

$$e(\tau_i) := \|\tilde{\varphi}(\tau_i; \tau_0, x, u) - \varphi(\tau_i; \tau_0, x, u)\|.$$

The main idea to show convergence is to use the triangle inequality in order to separate the error sources in the iteration (11.2) into the error caused by the previously accumulated error (a) and the local error (b). Abbreviating $\varphi(\tau_i) = \varphi(\tau_i; \tau_0, x_0, u)$ and $\tilde{\varphi}(\tau_i) = \tilde{\varphi}(\tau_i; \tau_0, x_0, u)$, this leads to the estimate

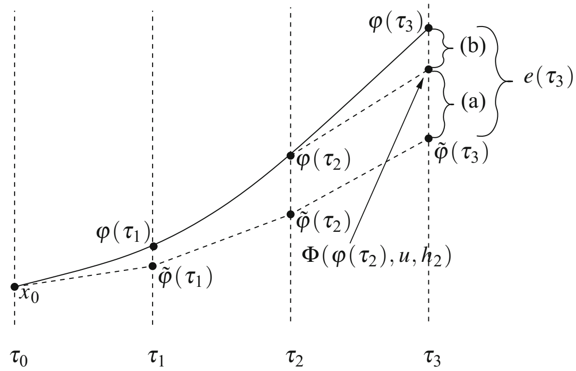
$\varphi(\tau_3)$
(b)
$e(\tau_3)$
(a)
$\tilde{\varphi}(\tau_3)$
$\varphi(\tau_2)$
$\tilde{\varphi}(\tau_2)$
$\Phi(\varphi(\tau_2), u, h_2)$
$\varphi(\tau_1)$
$\tilde{\varphi}(\tau_1)$
x_0
τ_0
τ_1
τ_2
τ_3

Fig. 11.1 Illustration of the separation of errors

$$\begin{aligned} e(\tau_{i+1}) = \|\tilde{\varphi}(\tau_{i+1}) - \varphi(\tau_{i+1})\| &= \|\Phi(\tilde{\varphi}(\tau_i), u, h_i) - \varphi(\tau_{i+1})\| \\ &\leq \underbrace{\|\Phi(\tilde{\varphi}(\tau_i), u, h_i) - \Phi(\varphi(\tau_i), u, h_i)\|}_{\text{accumulated error (a)}} + \underbrace{\|\Phi(\varphi(\tau_i), u, h_i) - \varphi(\tau_{i+1})\|}_{\text{local error (b)}}. \end{aligned} \tag{11.4}$$

The idea is sketched in Fig. 11.1 for $i = 2$.

In order to prove convergence we will use the following conditions which guarantee that both errors (a) and (b) remain small.

Definition 11.4 (i) A one step method satisfies the *Lipschitz condition* if for all compact subsets $K \subset \mathbb{R}^d$ and $Q \subset U$ there exists a constant $\Lambda > 0$ such that for all sufficiently small $h > 0$ the inequality

$$\|\Phi(x_1, u, h) - \Phi(x_2, u, h)\| \leq (1 + \Lambda h)\|x_1 - x_2\| \tag{11.5}$$

holds for all $x_1, x_2 \in K$ and all $u \in Q$.

(ii) A one step method is called *consistent* with *order of consistency* $p > 0$ if for all compact subsets $K \subset \mathbb{R}^d$ and $Q \subset U$ there exists a constant $C > 0$ such that for all sufficiently small $h > 0$ the inequality

$$\|\Phi(x, u, h) - \varphi(h; 0, x, u)\| \leq Ch^{p+1} \tag{11.6}$$

holds for all $x \in K$ and all $u \in Q$.

Inequality (11.5) guarantees that the propagation of previous errors within a one step method, i.e., term (a), stays bounded. The consistency condition (11.6), on the other hand, ensures that the local error (b) remains small.

One can easily show that the previously introduced Euler approximation as well as all explicit Runge–Kutta methods satisfy the Lipschitz condition (11.5) if the vector field f_c satisfies the Lipschitz condition from Assumption 2.4. The consistency condition (11.6), on the other hand, cannot be checked that easily in general. In order to verify that a method Φ exhibits an order of consistency $p \geq 1$, one utilizes the Taylor approximation of the method with respect to the step size h in $h = 0$, i.e.,

$$\Phi(x,u,h) = x + \sum_{i=1}^{p} \frac{h^i}{i!} \left.\frac{\partial^i}{\partial h^i}\right|_{h=0} \Phi(x,u,h) + O(h^{p+1}) \qquad (11.7)$$

and compares it to the Taylor approximation of the exact solution $\varphi(h; 0, x, u)$ with respect to h in $h = 0$. It turns out that this Taylor approximation can be computed without actually using the—in general unknown—solution φ. To this end, we use the higher order Lie derivative $L^i_{f_c}$, $i \in \mathbb{N}_0$, with respect to the vector field f_c which for arbitrary smooth vector fields $g : \mathbb{R}^d \times U \to \mathbb{R}^d$ is defined inductively by

$$L^0_{f_c} g(x,u) = g(x,u), \qquad L^i_{f_c} g(x,u) = \left(\frac{\partial}{\partial x} L^{i-1}_{f_c} g(x,u)\right) f_c(x,u).$$

Using the Lie derivative, the Taylor approximation of φ reads

$$\varphi(h; 0, x, u) = x + \sum_{i=1}^{p} \frac{h^i}{i!} L^{i-1}_{f_c} f_c(x,u) + O(h^{p+1}). \qquad (11.8)$$

Then, if the p summands in (11.7) and (11.8) coincide, the scheme is consistent with order p. In particular, if Φ can be written as

$$\Phi(x,u,h) = x + h\psi(x,u,h)$$

with a continuous function ψ satisfying $\psi(x,u,0) = f_c(x,u)$, then it follows that the order of consistency is at least $p = 1$.

Using this technique one can show that the order of consistency of the classical Runge–Kutta method is $p = 4$. More generally, the comparison of the summands can be used in order to derive conditions on the coefficients of arbitrary Runge–Kutta schemes for any consistency order $p \geq 1$. Unfortunately, the number of these condition grows exponentially with p, hence for $p \geq 10$ it is almost impossible to use them for constructing appropriate Runge–Kutta methods.

Note that in order to guarantee the order of consistency p the vector field f_c needs to be p times continuously differentiable with respect to x in order to ensure that approximation (11.8) holds. If the vector field depends on t, then it also needs to be p times continuously differentiable with respect to t on the interval $[\tau_i, \tau_{i+1}]$ if we want to apply (11.6) on this interval. This is no problem as long as $[\tau_i, \tau_{i+1}] \subseteq [0, T]$ which is always the case in this section. However, if we consider sampled data systems, i.e., (2.6) with v from (2.13) on an interval $[\tau_i, \tau_{i+1}]$ with $t_n \in (\tau_i, \tau_{i+1})$ for some sampling time t_n, this becomes a major issue since the control v and thus the map $t \mapsto f_c(x, v(t))$ is in general discontinuous and thus in particular nonsmooth at the sampling times. We will discuss this issue in Sect. 11.4, below.

After discussing the assumptions and how these assumptions can be checked, we are now ready to state the main result of this section.

Theorem 11.5 *If a one step method Φ satisfies the Lipschitz condition* (11.5) *and the consistency condition* (11.6) *with order p, then the approximation $\tilde{\varphi}$ from 11.2 is convergent in the sense of Definition 11.1(iii) with order of convergence p.*

Proof We will show (11.1) for each grid $\mathscr{G}$ on $[0, T]$ with $\overline{h} > 0$ sufficiently small. For simplicity of notation, we drop the index j in (11.1). To this end, fix two compact sets $K \subset \mathbb{R}^d$ and $Q \subset U$. Then the set

$$K_1 := \{\varphi(t; 0, x_0, u) \,|\, t \in [0, T], x_0 \in K, u \in Q\}$$

is again compact, since φ is continuous in all variables and images of compact sets under continuous maps are again compact. We choose some $\delta > 0$ and consider the compact set

$$K_2 := \overline{\mathscr{B}}_\delta(K_1) = \bigcup_{x \in K_1} \overline{\mathscr{B}}_\delta(x)$$

which contains exactly those points $x \in \mathbb{R}^d$ which have a distance less or equal δ to a point on a solution $x(t; 0, x_0, u)$ with $x_0 \in K$ and $u \in Q$. Let $\Lambda > 0$ and $C > 0$ be the constants in the Lipschitz condition (11.5) and the consistency condition (11.6), respectively, for $K = K_2$ and the set Q fixed above.

We first prove (11.1) under the following condition, which we will verify afterwards.

For all grids $\mathscr{G}$ with sufficiently small $\overline{h} > 0$, all initial values $x_0 \in K$ and all $u \in Q$ the grid function $\tilde{\varphi}$ from (11.2) satisfies $\varphi(\tau_i, \tau_0, x_0, u) \in K_2$ for all $\tau_i \in \mathscr{G}$. (11.9)

For proving (11.1) we choose $x_0 \in K$ and $u \in Q$ and abbreviate $\varphi(t) = \varphi(t; \tau_0, x_0, u)$ and $\tilde{\varphi}(\tau_i) = \tilde{\varphi}(\tau_i; \tau_0, x_0, u)$. With

$$e(\tau_i) := \|\tilde{\varphi}(\tau_i) - \varphi(\tau_i)\|$$

we denote the error at time $\tau_i \in \mathscr{G}$. Then from (11.4) we obtain

$$\begin{aligned} e(\tau_{i+1}) &\leq \|\Phi(\tilde{\varphi}(\tau_i), u, h_i) - \Phi(\varphi(\tau_i), u, h_i) \;+\; \|\Phi(\varphi(\tau_i), u, h_i) - \varphi(\tau_{i+1})\|[0] \\ &\leq (1 + \Lambda h_{i-1})\|\tilde{\varphi}(\tau_{i-1}) - \varphi(\tau_{i-1})\| \;+\; C h_{i-1}^{p+1}[0] \\ &= (1 + \Lambda h_{i-1}) e(\tau_{i-1}) + C h_{i-1}^{p+1}, \end{aligned}$$

using (11.5) and (11.6) for $K = K_2$ in the second inequality. These inequalities apply since the construction of K_1 and K_2 implies $\varphi(\tau_i) \in K_1 \subset K_2$ and (11.9) ensures $\tilde{\varphi}(\tau_i) \in K_2$.

By induction over i we now show that this inequality implies the estimate

$$e(\tau_i) \le C\overline{h}^p \frac{1}{\Lambda}(\exp(\Lambda(\tau_i - \tau_0)) - 1).$$

For $i = 0$ this inequality follows immediately. For $i - 1 \to i$ we use

$$\exp(\Lambda h_i) = 1 + \Lambda h_i + \frac{\Lambda^2 h_i^2}{2} + \ldots \ge 1 + \Lambda h_i$$

which together with the induction assumption yields

$$\begin{aligned} e(\tau_i) &\le (1 + \Lambda h_{i-1}) e(\tau_{i-1}) + C h_{i-1}^{p+1} [0] \\ &\le (1 + \Lambda h_{i-1}) C\overline{h}^p \frac{1}{\Lambda}(\exp(\Lambda(\tau_{i-1} - \tau_0)) - 1) + h_{i-1} \underbrace{C h_{i-1}^p}_{\le C\overline{h}^p} [0] \\ &= C\overline{h}^p \frac{1}{\Lambda}\Big(h_{i-1}\Lambda + (1 + \Lambda h_{i-1})(\exp(\Lambda(\tau_{i-1} - \tau_0)) - 1)\Big)[0] \\ &= C\overline{h}^p \frac{1}{\Lambda}\Big(h_{i-1}\Lambda + (1 + \Lambda h_{i-1})\exp(\Lambda(\tau_{i-1} - \tau_0)) - 1 - \Lambda h_{i-1}\Big)[0] \\ &= C\overline{h}^p \frac{1}{\Lambda}\Big((1 + \Lambda h_{i-1})\exp(\Lambda(\tau_{i-1} - \tau_0)) - 1\Big)[0] \\ &\le C\overline{h}^p \frac{1}{\Lambda}\Big(\exp(\Lambda h_{i-1})\exp(\Lambda(\tau_{i-1} - \tau_0)) - 1\Big)[0] \\ &= C\overline{h}^p \frac{1}{\Lambda}(\exp(\Lambda(\tau_i - \tau_0)) - 1). \end{aligned}$$

Since $\tau_0 = 0$ this implies (11.1) with $M = C(\exp(\Lambda T) - 1)/\Lambda$.

It remains to show that condition (11.9) is satisfied. We show that this assumption holds for all grids $\mathscr{G}$ whose maximal step size satisfies

$$C\overline{h}^p \le \frac{\delta\Lambda}{\exp(\Lambda(T - \tau_0)) - 1}.$$

To this end, we consider a numerical solution $\tilde{\varphi}(\tau_i) = \tilde{\varphi}(\tau_i, \tau_0, x_0, u)$ for some $x_0 \in K$ and $u \in Q$ and show $\tilde{\varphi}(\tau_i) \in K_2$ by induction. Since $\tilde{\varphi}(\tau_0) = x_0 \in K \subset K_2$ the assertion holds for $i = 0$.

For the induction step $i - 1 \to i$ assume that the induction assumption $\tilde{\varphi}(\tau_k) \in K_2$ holds for $k = 0, 1, \ldots, i - 1$. We have to show $\tilde{\varphi}(\tau_i) \in K_2$. Observe, that for the inequality

$$e(\tau_i) \le C\overline{h}^p \frac{1}{\Lambda}(\exp(\Lambda(T - \tau_0)) - 1)$$

to hold it is sufficient that $\tilde{\varphi}(\tau_k) \in K_2$ holds for $k = 0, 1, \ldots, i - 1$. By choice of $\overline{h}$ we thus obtain $e(\tau_i) \le \delta$, i.e.,

$$\|\tilde{\varphi}(\tau_i) - \varphi(\tau_i)\| \le \delta.$$

Since by construction of K_1 we have $\varphi(\tau_i) \in K_1$, it follows that $\tilde{\varphi}(\tau_i) \in \overline{B}_\delta(\varphi(\tau_i)) \subset K_2$, i.e., the desired property. □

11.3 Adaptive Step Size Control

The convergence theorem from the previous section shows that the presented one step methods are applicable to solve the underlying continuous time dynamics of the form (2.6) of a problem. Yet, so far we can only guarantee those methods to exhibit small errors if each time step h_i in the grid $\mathscr{G}$ is sufficiently small since the error bound in (11.1) depends on $\overline{h} = \max h_i$. In the literature, it is occasionally proposed to use the grid induced by the sampling times as computational grid, i.e., to choose $\tau_n = t_n = nT$. This, however, results in $\overline{h} = T$ and thus requires the sampling period T to be small in order to obtain an accurate approximation. Apart from the fact that it may not be desirable to use very small sampling periods, there are subtle pitfalls regarding stability of the closed-loop system when the accuracy of the approximate model and the sampling rate are linked, see the discussion in Sect. 11.6.

A way to avoid linking $\overline{h}$ and T is to use a constant step size $h_i \equiv \overline{h}$ with $\overline{h} = T/K$ for some $K \in \mathbb{N}$. Adjusting $\overline{h}$ appropriately, we can make the error term in (11.1) arbitrarily small without changing T. This, however, leads to equidistant grids which are known to be computationally inefficient since they do not reflect the properties of the solution. A much more efficient way is to choose the time steps h_i adapted to the solution, i.e., we allow for large h_i if the error is small and use small h_i when large errors are observed. However, we surely do not want to manually adapt the step sizes to every situation the NMPC controller may face since this would render such an algorithm to be inapplicable.

In order to obtain an efficient way to construct an adaptive grid $\mathscr{G}$, we consider step size control algorithms. Such methods are well established in the numerics of ordinary differential equations. In this section, we explain the central idea behind step size control algorithms. The key idea is to use two different one step methods Φ_1, Φ_2 with different orders of consistency $p_1 < p_2$ in order to compute a step length $h_i = \tau_{i+1} - \tau_i$ at time τ_i for the next time step which guarantees a predefined local error bound $\mathrm{tol}_{\mathrm{ODE}}$. Here, by $p_1 < p_2$ we mean that for $\Phi = \Phi_1$ the inequality (11.6) cannot hold for $p = p_2$, i.e., no matter how C is chosen (11.6) will be violated for all sufficiently small h. As in the previous sections, we consider the solution of (2.6) on one sampling interval $[0, T]$ on which the control u is constant.

In (11.4) we used the auxiliary term $\Phi(\varphi(\tau_i; \tau_0, x, u), u, h_i)$ in order to quantify the local error. Since the value of $\varphi(\tau_i; 0, x, u)$ is not available at runtime of a one step method, we cannot use it to guarantee the local error (a) to satisfy

$$\|\Phi(\varphi(\tau_i; \tau_0, x, u), u, h_i) - \varphi(\tau_i; \tau_0, x, v)\| \leq \mathrm{tol}_{\mathrm{ODE}}.$$

To circumvent this problem, in the triangle inequality for estimating $e(\tau_{i+1})$ we insert the term $\varphi(\tau_{i+1}; \tau_i, \tilde{\varphi}(\tau_i; \tau_0, x, u), u)$ instead of $\Phi(\varphi(\tau_i; \tau_0, x, u), u, h_i)$. Using that by the cocycle property we have $\varphi(\tau_{i+1}; \tau_0, x, u) = \varphi(\tau_{i+1}; \tau_i, \varphi(\tau_i; \tau_0, x, u), u)$, this leads to the inequality

$$
\begin{aligned}
&\|\tilde{\varphi}(\tau_{i+1}; \tau_0, x, u) - \varphi(\tau_{i+1}; \tau_0, x, u)\| \leq \\
&\qquad \leq \|\Phi(\tilde{\varphi}(\tau_i; \tau_0, x, u), u, h_i) - \varphi(\tau_{i+1}; \tau_i, \tilde{\varphi}(\tau_i; \tau_0, x, u), u)\| \\
&\qquad\quad + \|\varphi(\tau_{i+1}; \tau_i, \tilde{\varphi}(\tau_i; \tau_0, x, u), u) - \varphi(\tau_{i+1}; \tau_i, \varphi(\tau_i; \tau_0, x, u), u)\|.
\end{aligned}
$$

In this sum, the second term essentially depends on the error of the approximation at time instant τ_i, which is independent of the choice of $h_i = \tau_{i+1} - \tau_i$. Hence, for choosing h_i we only consider the first summand. More precisely, we attempt to choose h_i such that the tolerable error bound

$$
\|\Phi(\tilde{\varphi}(\tau_i; \tau_0, x, u), u, h_i) - \varphi(\tau_{i+1}; t_i, \tilde{\varphi}(\tau_i; \tau_0, x, u), u)\| \leq \mathrm{tol}_{\mathrm{ODE}}
$$

is satisfied.

When trying to implement this method, one faces the problem that the value $\varphi(\tau_{i+1}; t_i, \tilde{\varphi}(\tau_i; \tau_0, x, u), u)$ is not known. This is where the idea of using two methods Φ_1 and Φ_2 with different orders of consistency $p_2 > p_1$ is used. Setting $\Phi = \Phi_1$ and approximating $\varphi(\tau_{i+1}; t_i, \tilde{\varphi}(\tau_i; \tau_0, x, u), u)$ by the more accurate method Φ_2 one can show the following theorem.

Theorem 11.6 *Consider two one step methods Φ_1, Φ_2 with orders of consistency p_1, p_2 satisfying $p_2 \geq p_1 + 1$. Then there exist constants $k_1, k_2 > 0$ such that for all sufficiently small $h_i > 0$ the computable error*

$$
\overline{\varepsilon} := \|\Phi_1(\tilde{\varphi}(\tau_i; 0, x, u), u, h_i) - \Phi_2(\tilde{\varphi}(\tau_i; 0, x, u), u, h_i)\| \tag{11.10}
$$

and the local error of the one step method Φ_1

$$
\varepsilon := \|\Phi_1(\tilde{\varphi}(\tau_i; 0, x, u), u, h_i) - \varphi(\tau_{i+1}; \tau_i, \tilde{\varphi}(\tau_i; 0, x, u), u)\|
$$

satisfy the inequality

$$
k_1\varepsilon \leq \overline{\varepsilon} \leq k_2\varepsilon.
$$

Proof First we define the errors

$$
\eta_{i,j} := \Phi_j(\tilde{\varphi}(\tau_i; 0, x, u), u, h_i) - \varphi(\tau_{i+1}; \tau_i, \tilde{\varphi}(\tau_i; 0, x, u), u)
$$

for both one step methods Φ_j, $j = 1, 2$. By Definition 11.4(ii) we obtain the local error bounds $\varepsilon_{i,j} := \|\eta_{i,j}\| \leq C_j h_i^{p_j+1}$. Using $p_2 \geq p_1 + 1$ and the fact that this implies $\varepsilon_{i,1} \geq C h_i^{p_2+1}$ for all $C > 0$ and all sufficiently small $h_i > 0$, we can conclude $\theta := \varepsilon_{i,2}/\varepsilon_{i,1} < 1$ if h_i is chosen sufficiently small since $\theta \to 0$ as $h_i \to 0$. We fix $\theta_0 < 1$, consider $h_i > 0$ such that $\theta < \theta_0 < 1$ holds and define

$$
\overline{\eta} := \Phi_1(\tilde{\varphi}(\tau_i; 0, x, u), u, h_i) - \Phi_2(\tilde{\varphi}(\tau_i; 0, x, u), u, h_i) = \eta_{i,1} - \eta_{i,2}.
$$

Then we have

$$\begin{aligned}(1-\theta)\varepsilon_{i,1} &= (1-\theta)\|\eta_{i,1}\| = \left(1 - \frac{\|\eta_{i,1}-\overline{\eta}\|}{\|\eta_{i,1}\|}\right)\|\eta_{i,1}\| = \\ &= \|\eta_{i,1}\| - \|\eta_{i,1}-\overline{\eta}\| \leq \|\overline{\eta}\| = \overline{\varepsilon}\end{aligned}$$

which yields the lower bound $k_1 = 1 - \theta_0$ and

$$\begin{aligned}\overline{\varepsilon} = \|\overline{\eta}\| &\leq \|\eta_{i,1}\| + \|\eta_{i,1}-\overline{\eta}\| = \left(1 + \frac{\|\eta_{i,1}-\overline{\eta}\|}{\|\eta_{i,1}\|}\right)\|\eta_{i,1}\| = \\ &= (1+\theta)\|\eta_{i,1}\| = (1+\theta)\varepsilon_{i,1}\end{aligned}$$

giving the upper bound $k_2 = 1 + \theta_0$. □

Using Theorem 11.6 we can now compute a suitable step size h_i if we additionally assume that the local error is of the form $\varepsilon_{i,1} \approx c_i h_i^{p_1+1}$ for small h_i. Note that for Runge–Kutta methods this assumption is satisfied if the vector field f is p_1+2 times continuously differentiable. In this case, c_i is given by the coefficient of the $h_i^{p_1+1}$ term in the Taylor approximation of the method.

For small step sizes it follows from the proof of Theorem 11.6 that $k_1 \approx k_2 \approx 1$, i.e., $\overline{\varepsilon} \approx \varepsilon_{i,1} \approx c_i h_i^{p_1+1}$ which gives us the estimate $\overline{c}_i \approx \overline{\varepsilon}/h_i^{p_1+1}$ for the coefficient c_i. Hence, the error tolerance tol_{ODE} is satisfied (approximately) for the step size

$$\text{tol}_{\text{ODE}} = \overline{c}_i h_{i,\text{new}}^{p_1+1} = \frac{\overline{\varepsilon}}{h_i^{p_1+1}} h_{i,\text{new}}^{p_1+1} \iff h_{i,\text{new}} = \sqrt[p_1+1]{fac\frac{\text{tol}_{\text{ODE}}}{\overline{\varepsilon}}}\,h_i \tag{11.11}$$

Since all these equalities are only satisfied approximately, a security factor $fac \in (0, 1)$ has been introduced to compensate for these approximation errors. For this factor, $fac = 0.9$ is a typical choice in many algorithms.

A schematic implementation of a one step scheme with adaptive step size is given in Algorithm 11.7, below. This algorithm combines the iteration (11.2) with the computation of the step size h_i described above. Here, we solve (2.6) on one sampling interval $[0, T]$ using the length T of the sampling interval as an initial choice for the first step size h_0. For large T, one may alternatively choose $h_0 < T$. In each step the error $\overline{\varepsilon}$ is computed. If $\overline{\varepsilon}$ exceeds the tolerance tol_{ODE}, then the step is rejected and repeated using the new step size from (11.11). If $\overline{\varepsilon}$ maintains the desired tolerance, then the step is accepted and the new step size from (11.11) is used as an initial choice for the next time step.

Algorithm 11.7 Suppose an initial value x, a control value u, a tolerance tol_{ODE}, and sampling period T are given.

(1) Set $\tilde{\varphi}(0; 0, x, v) = x$, $i = 0$, $\tau_0 = 0$, $h_0 = T$
(2) If $\tau_i = T$ stop; If $\tau_i + h_i > T$ set $h_i = T - \tau_i$

(3) Set $\tau_{i+1} = \tau_i + h_i$ and compute $\Phi_1(\tilde{\varphi}(\tau_{i+1}; t_j, x, v), v, h_i)$, $\Phi_2(\tilde{\varphi}(\tau_{i+1}; t_j, x, v), v, h_i)$

(4) Compute $\overline{\varepsilon}$ and $h_{i,\text{new}}$ according to (11.10), (11.11)

(5) If $\overline{\varepsilon} > \text{tol}_{\text{ODE}}$ set $h_i = h_{i,\text{new}}$ and goto (3)

(6) If $\overline{\varepsilon} \leq \text{tol}_{\text{ODE}}$ set $\tilde{\varphi}(\tau_{i+1}; \tau_0, x, u) = \Phi_2(\tilde{\varphi}(\tau_{i+1}; \tau_0, x, u), u, h_i)$, $h_{i+1} = h_{i,\text{new}}$, $i = i + 1$ and goto (2)

In practical implementations, this basic algorithm is often refined in various ways. For instance, the new step size may be derived on the basis of a weighted sum of the absolute and the relative error instead of using only the absolute error as above. Upper and lower bounds for the time step h_i as well as for the ratio between h_i and h_{i+1} are also frequently used in practice.

Although the evaluation of two methods Φ_1 and Φ_2 and their possibly repeated evaluation in every step seems to be computationally more demanding, step size control algorithms are usually much more efficient than the use of equidistant time grids. This is due to two different aspects: on the one hand, there typically exist regions which allow for larger time steps and thus allow for a faster progress of the adaptive iteration procedure. On the other hand, the additional effort of simultaneously evaluating two methods can be reduced significantly by embedding these methods into each other. This means that the less accurate Runge–Kutta method Φ_1 uses the same stages k_i, cf. Definition 11.2, as the more accurate methods Φ_2 and thus the stages k_i only need to be evaluated once for both methods. One standard embedded method is the Dormand–Prince method of order (4)5, also called DoPri5, in which Φ_1 has order $p_1 = 4$ and Φ_2 is of order $p_2 = 5$. The Butcher tableau is displayed in Table 11.2. The second last line specifies the coefficients b_i for Φ_1 and the last line the b_i for Φ_2.

With the same induction as in the proof of Theorem 11.5, one sees that if the local errors maintain the tolerances $\text{tol}_{\text{ODE}} = \varepsilon h_i$ for some $\varepsilon > 0$, then the overall error at time T can be estimated as $e(T) \leq \varepsilon(\exp(\Lambda T) - 1)/\Lambda$ and thus scales linearly with ε. It should, however, be mentioned that adaptive step size selection schemes usually do not *rigorously* maintain the specified error tolerance. The reason for this is that Theorem 11.6 and the derivation of (11.11) require h_i to be sufficiently small. Suitable upper bounds which quantify this "sufficiently small" are, however, difficult to obtain without an extensive a priori analysis of the individual system and can therefore not be enforced in practice. Hence, the step size selection algorithm may select large step sizes for which the error estimation is no longer valid and thus the desired accuracy is no longer guaranteed. Thus, in general only equidistant grids with sufficiently small maximal step size $\overline{h}$ provide rigorous error bounds. Still, numerical experience shows that in the vast majority of examples error estimation-based adaptive step size algorithms like Algorithm 11.7 perform very reliably.

Table 11.2 Butcher tableau of the DoPri(4)5 method

0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{85}$	
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

11.4 Using the Methods Within the NMPC Algorithms

Looking at the NMPC Algorithm 3.11 and its variants, we see that in every iteration an optimal control problem has to be solved. To this end, the optimization algorithm needs to be able to compute the solution x_u and to evaluate the functional J_N. In fact, there are various ways for incorporating x_u into the optimization algorithm, for details see Sect. 12.1. However, no matter which method from this section we use, we need to be able to evaluate $\varphi(T, 0, x, u)$ in (2.8) numerically.

To this end, we replace the unknown map $\varphi(T, 0, x, u)$ in (2.8) by its approximation $\tilde{\varphi}(T, 0, x, u)$ from Algorithm 11.7. This way we end up with the definition

$$x^+ = f(x, u) := \tilde{\varphi}(T, 0, x, u). \tag{11.12}$$

Iterating this map according to (2.2), which amounts to calling Algorithm 11.7 N times with initial values $x_u(n, x)$ and control values $u(n)$, $n = 0, \dots, N - 1$, we can then obtain an approximate predicted solution trajectory. Proceeding this way, one should keep in mind that the numerical scheme provides only an approximation of the exact solution. The effects of the approximation errors will be discussed in Sect. 11.5, below.

When the stage cost ℓ is defined via the integral formula (3.4) with running cost L, then we can efficiently include the numerical evaluation of the integral

$$\ell(x,u) = \int_0^T L(\varphi(t,0,x,u),u)dt$$

into the computation of $\tilde{\varphi}$. Here, we have removed the argument t from u because—following the convention in this chapter—u is constant on the sampling interval $[0,T]$. In order to compute the integral, consider the augmented ordinary differential equation

$$\dot{\overline{x}}(t) = \overline{f}(\overline{x}(t),u) \tag{11.13}$$

with

$$\overline{x}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \in \mathbb{R}^d \times \mathbb{R} \quad \text{and} \quad \overline{f}(\overline{x},u) = \begin{pmatrix} f_c(x,u) \\ L(x,u) \end{pmatrix}.$$

Solving (11.13) with initial condition $\overline{x} = (x,0)$ we obtain the solution

$$\overline{\varphi}(T,x,u) = \begin{pmatrix} \varphi(T,x,u) \\ \ell(x,u) \end{pmatrix}.$$

Thus, solving (11.13) numerically yields a numerical solution whose first n components equal $\tilde{\varphi}(T,x,u)$ and whose $(n+1)$st component approximates $\ell(x,u)$. Proceeding this way we avoid the use of a separate numerical integration formula, in particular, we do not have to store the intermediate values $\tilde{\varphi}(\tau_i,x,u)$ for a subsequent numerical integration of L. Furthermore, the adaptive step size algorithm ensures that ℓ is approximated with the same accuracy as the solution φ.

As we will see in detail in Sect. 12.1, one way to incorporate the dynamics of the system into the numerical optimization algorithm is to externally compute the whole trajectory $x_u(\cdot,x_0)$, an approach called *recursive elimination*. In order to compute this trajectory, instead of defining f via (11.12) and then iterating f according to (2.2) one could apply a numerical one step method directly on the interval $[0,NT]$. This way we obtain a numerical approximation of x_u (and of J_N if we include the computation of ℓ) on $[0,NT]$ invoking Algorithm 11.7 only once. However, this has to be done with care. As already mentioned, in order to guarantee consistency with order p of the numerical schemes, it is important that the map $(t,x) \mapsto f_c(x,v(t))$ in (2.6) is p times continuously differentiable. Formally, this can be shown by extending the Formulas (11.7) and (11.8) to time varying vector fields f_c.

However, we can also give an informal explanation of this fact: when considering the solution of (2.6) with zero-order hold, then the control function v is discontinuous at the sampling times t_n. Consequently, the solution $\varphi(t,0,x,v)$ is not differentiable for $t=t_n$, as sketched in Fig. 11.2. Since we cannot approximate nonsmooth functions by a Taylor approximation, Formula (11.8) will not hold if we replace $\varphi(h,0,x,u)$ by $\varphi(\tau_i+h_i,\tau_i,x,v)$ with $t_n \in (\tau_i,\tau_i+h_i) = (\tau_i,\tau_{i+1})$ for some sampling time t_n. Thus, we have to make sure that this situation does not happen.

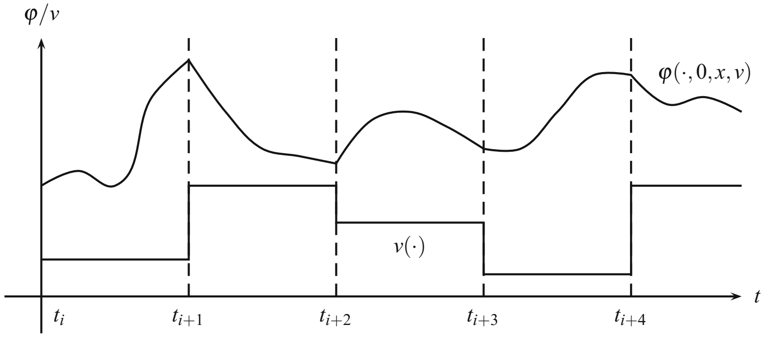

Fig. 11.2 Approximation of the sampled data solution

Defining the set of sampling times

$$\mathscr{T} := \{t_n \in \mathbb{R} \mid t_n = nT, n = 0, \ldots, N\} \tag{11.14}$$

and using the time grid

$$\mathscr{G} := \{\tau_i \in [0, NT] \mid \tau_i \text{ is a discretization time in the one step method}\}, \tag{11.15}$$

in order to exclude the existence of i and n with $t_n \in (\tau_i, \tau_i + h_i) = (\tau_i, \tau_{i+1})$ we need to make sure that the inclusion $\mathscr{T} \subset \mathscr{G}$ holds. This assumption is not very restrictive, however, in order to ensure it we need to appropriately adjust Algorithm 11.7.

11.5 Numerical Approximation Errors and Stability

Defining the discrete dynamics f via the numerical approximation $\tilde{\varphi}$, cf. (11.12), introduces errors in the predictions x_u in the optimal control problems and its variants. In this section, we shift our focus from analyzing the effects of these errors on the open loop toward their effect on the closed loop. To this end, we utilize the techniques from Sects. 7.5–7.9 and – similar to these sections – restrict ourselves to constant reference $x^{\text{ref}} \equiv x_*$ in order to simplify the exposition. For the extension to time varying x^{ref} we refer to the remarks following the main results in Sects. 7.5–7.9.

As a general assumption we suppose that for each $\varepsilon > 0$ we can compute a numerical approximation $\tilde{\varphi}^\varepsilon$ which satisfies

$$\|\tilde{\varphi}^\varepsilon(T, 0, x, u) - \varphi(T, 0, x, u)\| \leq \varepsilon \tag{11.16}$$

for some $\varepsilon > 0$, all $x \in \mathbb{X}$ and $u \in \mathbb{U}(x)$. As discussed at the end of Sect. 11.3, such an estimate is rigorously ensured for $\tilde{\varphi}^\varepsilon$ generated by one step methods on equidistant grids with sufficiently small $\overline{h} > 0$ but can typically also be expected for $\tilde{\varphi}$ from the adaptive step size Algorithm 11.7 by adjusting the tolerance $\mathrm{tol}_{\mathrm{ODE}}$ appropriately. Observe that since (11.1) only holds for x and u from compact sets, in the case of equidistant grids we may have to adjust $\overline{h} > 0$ to x and u in order to ensure (11.16) if $\mathbb{X}$ or $\mathbb{U}(x)$ are noncompact. In case of Algorithm 11.7 the step size will be automatically adjusted by the step size selection mechanism.

Given that $\tilde{\varphi}^\varepsilon$ is an approximation of the true solution φ it seems natural to consider $\tilde{\varphi}^\varepsilon$ as a perturbed version of φ. However, since by definition the model used in the NMPC algorithm—i.e., the numerical approximation $\tilde{\varphi}^\varepsilon$—is the nominal model, we need the converse interpretation in order to apply the results from Sects. 7.5–7.9. In what follows we show that the closed loop obtained from the exact sampled data system (2.8) can be considered as a perturbed system in the sense of Sect. 7.5. To this end, we consider the following setting.

The NMPC algorithm is run with the numerically approximated discrete dynamics

$$f(x,u) = f^\varepsilon(x,u) := \tilde{\varphi}^\varepsilon(T,0,x,u) \tag{11.17}$$

as a nominal model. The resulting NMPC-feedback law is denoted by $\mu_N^\varepsilon x_{\mu_N^\varepsilon}^\varepsilon$ and $\tilde{x}_{\mu_N^\varepsilon}^\varepsilon$ we denote the corresponding *nominal* and *perturbed NMPC closed-loop trajectory* from (3.5) and (7.7) with $f = f^\varepsilon = \tilde{\varphi}^\varepsilon$ and $\mu_N = \mu_N^\varepsilon$, respectively, i.e.,

$$x_{\mu_N^\varepsilon}^\varepsilon(n+1) = f^\varepsilon(x_{\mu_N^\varepsilon}^\varepsilon(n), \mu_N^\varepsilon(x_{\mu_N^\varepsilon}^\varepsilon(n)))$$

and

$$\tilde{x}_{\mu_N^\varepsilon}^\varepsilon(n+1) = f^\varepsilon(\tilde{x}_{\mu_N^\varepsilon}^\varepsilon(n), \mu_N^\varepsilon(\tilde{x}_{\mu_N^\varepsilon}^\varepsilon(n) + e(n))) + d(n).$$

The closed-loop system obtained from applying the numerically computed NMPC-feedback μ_N^ε to the exact model $f = \varphi$ from (2.8) according to (3.5), i.e.,

$$x_{\mu_N^\varepsilon}^{ex}(n+1) = f(x_{\mu_N^\varepsilon}^{ex}(n), \mu_N^\varepsilon(x_{\mu_N^\varepsilon}^{ex}(n))),$$

will be called the *exact closed-loop system*. The resulting trajectories will be denoted by $x_{\mu_N^\varepsilon}^{ex}$.

Note that the same NMPC-feedback law μ_N^ε—computed from $f = f^\varepsilon = \tilde{\varphi}^\varepsilon$—is used in (3.5) for generating $x_{\mu_N^\varepsilon}^\varepsilon$ and $x_{\mu_N^\varepsilon}^{ex}$. The difference between the two trajectories only lies in the map f in (3.5) which is given by $f = f^\varepsilon = \tilde{\varphi}^\varepsilon$ for $x_{\mu_N^\varepsilon}^\varepsilon$ and by $f = \varphi$ for $x_{\mu_N^\varepsilon}^{ex}$. Using this notation we obtain the following result.

Lemma 11.8 (Perturbed solution) *Consider the discrete time dynamics $f = f^\varepsilon$ from* (11.17) *obtained from a numerical approximation $\tilde{\varphi}^\varepsilon$ satisfying* (11.16)*, an NMPC-feedback law μ_N^ε with $\mu_N^\varepsilon(x) \in \mathbb{U}(x)$ and the solution $x_{\mu_N^\varepsilon}^\varepsilon$ of the corresponding closed-loop system* (3.5)*. Consider, furthermore, the solution $x_{\mu_N^\varepsilon}^{ex}$ of the exact closed-loop system.*

Then for each $x_0 \in \mathbb{X}$ there exists a perturbation sequence $d(\cdot) \in (\mathbb{R}^d)^\infty$ with $\|d(n)\| \leq \varepsilon$ such that the solution $\tilde{x}^\varepsilon_{\mu^\varepsilon_N}(n, x_0)$ of the perturbed system (7.7) with $f = f^\varepsilon$ and $e \equiv 0$ satisfies

$$x^{ex}_{\mu^\varepsilon_N}(n, x_0) = \tilde{x}^\varepsilon_{\mu^\varepsilon_N}(n, x_0)$$

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

Proof Define

$$d(n) := \varphi(T, 0, x^{ex}_{\mu^\varepsilon_N}(n, x_0), \mu^\varepsilon_N(x^{ex}_{\mu^\varepsilon_N}(n, x_0))) - \tilde{\varphi}^\varepsilon(T, 0, x^{ex}_{\mu^\varepsilon_N}(n, x_0), \mu^\varepsilon_N(x^{ex}_{\mu^\varepsilon_N}(n, x_0)))$$

for all $n \in \mathbb{N}_0$. Then (11.16) with $x = x^{ex}_{\mu^\varepsilon_N}(n, x_0)$ and $u = \mu^\varepsilon_N(x^{ex}_{\mu^\varepsilon_N}(n, x_0))$ implies $\|d(n)\| \leq \varepsilon$ for all $n \in \mathbb{N}_0$. We show the desired identity by induction over n. For $n = 0$ we obtain $x^{ex}_{\mu^\varepsilon_N}(0, x_0) = x_0 = \tilde{x}^\varepsilon_{\mu^\varepsilon_N}(0, x_0)$. For $n \to n + 1$ assume that $x^{ex}_{\mu^\varepsilon_N}(n, x_0) = \tilde{x}^\varepsilon_{\mu^\varepsilon_N}(n, x_0)$ holds. Then we get

$$\begin{aligned}
x^{ex}_{\mu^\varepsilon_N}(n+1, x_0) &= \varphi(T, 0, x^{ex}_{\mu^\varepsilon_N}(n, x_0), \mu^\varepsilon_N(x^{ex}_{\mu^\varepsilon_N}(n, x_0))) \\
&= \tilde{\varphi}^\varepsilon(T, 0, x^{ex}_{\mu^\varepsilon_N}(n, x_0), \mu^\varepsilon_N(x^{ex}_{\mu^\varepsilon_N}(n, x_0))) + d(n) \\
&= f^\varepsilon(x^{ex}_{\mu^\varepsilon_N}(n, x_0), \mu^\varepsilon_N(x^{ex}_{\mu^\varepsilon_N}(n, x_0))) + d(n) \\
&= f^\varepsilon(\tilde{x}^\varepsilon_{\mu^\varepsilon_N}(n, x_0), \mu^\varepsilon_N(\tilde{x}^\varepsilon_{\mu^\varepsilon_N}(n, x_0))) + d(n) \quad = \quad \tilde{x}^\varepsilon_{\mu^\varepsilon_N}(n+1, x_0).
\end{aligned}$$

This shows the assertion. □

Lemma 11.8 shows that the closed-loop solution for the discrete time model obtained from the exact sampled data system (2.8) can be interpreted as a perturbed solution of the discrete time model obtained from the numerical approximation (11.17). The size of the perturbation $d(\cdot)$ directly corresponds to the numerical error (11.16).

This lemma enables us to use all results from Sects. 7.5–7.9 in order to conclude stability properties for $x^{ex}_{\mu^\varepsilon_N}$. The appropriate stability property is given by the following definition, cf. Definition 7.24.

Definition 11.9 Consider the exact closed-loop system (2.5) with $f = \varphi$ from (2.8) with μ^ε_N computed from $f = f^\varepsilon = \tilde{\varphi}^\varepsilon$ from (11.17) satisfying (11.16) for some $\varepsilon > 0$. Given a set $A \subseteq \mathbb{X}$ such that the optimal control problem defining μ^ε_N is feasible for all $x_0 \in A$, we say that x_* is *semiglobally practically asymptotically stable on A with respect to the numerical error ε* if there exists $\beta \in$ such that the following property holds: for each $\delta > 0$ and $\Delta > \delta$ there exists $\overline{\varepsilon} > 0$, such that for each initial value $x_0 \in A$ with $|x_0|_{x_*} \leq \Delta$ and each $\varepsilon \in (0, \overline{\varepsilon}]$ the solution $x^{ex}_{\mu^\varepsilon_N}(\cdot, x_0)$ satisfies $x^{ex}_{\mu^\varepsilon_N}(k, x_0) \in A$ and

$$|x^{ex}_{\mu^\varepsilon_N}(k, x_0)|_{x_*} \leq \max\{\beta(|x_0|_{x_*}, k),\ \delta\}$$

for all $k \in \mathbb{N}_0$.

The following theorem now gives conditions under which this stability property holds.

Theorem 11.10 *(*Stability for perturbed solution*) Consider the NMPC-feedback laws μ_N^ε obtained from one of the NMPC algorithms from Theorems 7.26 and 7.36 or 7.41 with $f = f^\varepsilon = \tilde{\varphi}^\varepsilon$ from* (11.17)*. Assume that* (11.16) *holds and that there is $\varepsilon_0 > 0$ such that one of the following assumptions is satisfied for all $\varepsilon \in (0, \varepsilon_0]$.*

(i) In case of Theorem 7.26, assume that α, α_1, α_2, α_3 in Theorem 4.11 as well as ω_V and ω_f can be chosen independently of $\varepsilon > 0$.
(ii) In case of Theorem 7.36, assume that α, α_1, $\tilde{\alpha}$ in Theorem 6.20, β from Assumption 7.35 and η in Definition 7.33 can be chosen independently of $\varepsilon > 0$.
(iii) In case of Theorem 7.41, assume that α, α_1, α_2, α_3 in Theorem 4.11, δ, γ, ε' in Assumption 7.38 and the bound on f as well as the moduli of continuity of f and ℓ can be chosen independently of $\varepsilon > 0$.

Then the exact closed-loop system (2.5) *with $f = \varphi$ from* (2.8) *is semiglobally practically asymptotically stable with respect to ε from* (11.16) *in the sense of Definition 11.9 on the set A specified in the respective theorem.*

Proof The respective theorems ensure semiglobal practical asymptotic stability for all perturbed trajectories $\tilde{x}^\varepsilon_{\mu_N^\varepsilon}$ with respect to $\overline{d}$ and $\overline{e}$ in the sense of Definition 7.24. An inspection of the proofs of the respective theorems then reveals that the uniformity assumptions (i)–(iii) guarantee that for given δ and $\varDelta$ the bounds $\overline{d}$ and $\overline{e}$ and the function $\beta \in$ in Definition 7.24 are independent of $\varepsilon > 0$.

Fixing δ and $\varDelta$ we thus find $\overline{d} > 0$ such that each perturbed solution $\tilde{x}^\varepsilon_{\mu_N^\varepsilon}$ with perturbations $\|d(n)\| \le \overline{d}$ and $e \equiv 0$ satisfies the conditions of Definition 7.24 for all $\varepsilon \in (0, \varepsilon_0]$. Setting $\overline{\varepsilon} = \min\{\overline{d}, \varepsilon_0\}$ and using that by Lemma 11.8 the exact closed-loop trajectory $x^{ex}_{\mu_N^\varepsilon}$ equals one of the trajectories $\tilde{x}^\varepsilon_{\mu_N^\varepsilon}$ with $\overline{d} = \varepsilon$ and $\overline{e} = 0$, we obtain that $x^{ex}_{\mu_N^\varepsilon}$ satisfies the conditions of Definition 11.9 for the given δ and $\varDelta$ and all $\varepsilon \in (0, \overline{\varepsilon}]$. This yields the assertion. □

Note that Theorem 11.10 only guarantees the stability of the discrete time closed-loop system (2.5) with f from (2.8) but not for the sampled data closed loop (2.30). In order to conclude stability properties of (2.30) the techniques from Sect. 2.4 can be used. While Theorem 2.27 and its assumptions are formulated for the case of "real" asymptotic stability, its statement, and proof can be straightforwardly extended to the semiglobal practical setting of Definition 11.9. Recall from Remark 4.13 that the assumptions of Theorem 2.27 are satisfied for suitable integral costs (3.4). Although we have not rigorously analyzed the effect of the error induced by the numerical approximation of such integral costs, we conjecture that the estimates in Remark 4.13 remain valid in a suitable approximate sense if these errors are sufficiently small.

Since numerical approximations are used in virtually all NMPC algorithms for sampled data systems, Theorem 11.10 implies that all such algorithms need appropriate robustness—either inherently as in case (i) or by an appropriate design of the state constraints as in cases (ii) and (iii) of Theorem 11.10—in a uniform way with respect to ε in order to ensure semiglobal practical stability in the presence of numerical errors. In practice, however, this is hardly ever rigorously ensured. The reason for this is that for good numerical methods numerical errors are usually very small compared to other error sources like model errors, external perturbations, etc. Although even very small errors may in the worst case be destabilizing, as illustrated by Example 7.31, it is not very likely that this indeed happens and—also according to our experience—such phenomena are hardly ever observed in simulations or practical examples. Hence, unless robustness is needed in order to cope with error sources which are significantly larger than the numerical errors discussed in this chapter, for most practical purposes it seems justified to neglect the robustness issue, provided, of course, the numerical errors are indeed sufficiently small. Still, one has to keep in mind that proceeding this way does not rigorously ensure stability of the exact closed-loop system.

11.6 Notes and Extensions

The material contained in Sects. 11.1–11.3 can be found in many textbooks on numerical analysis for ordinary differential equations, like, e.g., the books by Deuflhard an Bornemann [2], Hairer, Nørsett and Wanner [8] or Stoer and Bulirsch [11]. Clearly, the presentation in this chapter cannot replace any of these textbooks and aims at giving an introduction into the subject rather than a comprehensive treatment.

Among the many topics we have not covered in this chapter we would in particular like to mention stiff problems and differential algebraic equations (DAEs), often called descriptor systems in systems theory. While stiff problems “look” like normal ordinary differential equations, they are very difficult to solve with the explicit methods presented in Sect. 11.1. For stiff equations, which often appear when modeling technical systems, an adaptive step size algorithm like Algorithm 11.7 will typically select very small time steps even though the solution is almost constant. Explaining the precise mathematical reasons for this behavior goes beyond these notes, but we would at least like to mention that so-called implicit methods perform much better for stiff equations. DAEs are ordinary differential equations with additional algebraic constraints, often given implicitly. DAEs appear as models, e.g., in mechanics and electrical engineering and NMPC is perfectly suited for handling DAES, however, the solution methods presented in this chapter do not apply to such equations and specialized numerical schemes are needed, which are again often of the implicit type. While also covered in some standard textbooks, there is a large amount of literature particularly devoted to stiff and DAE problems, as, e.g., Hairer and Wanner [7], and we refer the reader to such books for more details.

As Examples 2.12 and 6.32 show, NMPC is also suitable for infinite-dimensional systems generated by controlled PDEs. NMPC for PDEs requires the solution of an optimal control problem for PDEs in each step. The monograph by Troeltzsch [12] provides a good introduction into such problems. A simple way to approach this problem numerically is to proceed similar as described for the ordinary differential equations in this chapter with an additional spatial discretization by, e.g., a finite difference method (which is what we used in Example 6.32), see, e.g., LeVeque [9] or a finite element method, see, e.g., Braess [1]. However, it is by no means clear whether this is the most efficient way of approaching the problem numerically; in fact, the development of suitable numerical schemes is currently a very active research area. Furthermore, we are not aware of a rigorous analysis of the effects of spatial discretization errors in NMPC controller design.

The need to use numerical approximations and the consequences for the stability analysis discussed in Sect. 11.5 are largely ignored in the NMPC literature. An exception to this rule are the papers by Gyurkovics and Elaiw [5, 6], which are in the same spirit as cases (i) and (iii) of Theorem 11.10 in the sense that they exploit uniform continuity properties, in particular of the optimal value function V_N. However, these results require Lyapunov function terminal costs and do not consider state constraints as in cases (ii) and (iii) of Theorem 11.10.

More generally, the problem considered in Sect. 11.5 can be seen as a special case of a nonlinear controller design based on approximate models. A comprehensive treatment of this topic in a rather general setting can be found in Nešić and Teel [10]. An application to infinite horizon optimal control based feedback design was given in Grüne and Nešić [4]. The idea to treat numerical errors as perturbations is classical in numerical analysis. In a control theoretic framework this idea was used extensively in the monograph Grüne [3]. All these approaches are similar to our approach in the sense that the stability property of the approximate system is required to be robust in some suitable sense, that the robustness can be quantified and that this quantitative measure of the robustness is independent of the numerical accuracy. In all cases the obtained stability property is semiglobal practical stability, just as in Theorem 11.10. State constraints are, however, again not considered in these references.

Nešić and Teel [10] also nicely illustrate the pitfalls of feedback design based on approximate models by means of simple examples and discuss the case in which the numerical accuracy is linked to the sampling period T. Roughly speaking, in this case uniform continuity of the Lyapunov function under consideration is not sufficient in order to ensure stability of the exact closed-loop system. Rather, a stronger property like Lipschitz continuity with Lipschitz constant independent of the numerical accuracy ε is needed in this case.

There are numerous issues related to numerical errors we have not addressed in this chapter. For instance, numerical errors may lead to the situation that the inequalities in Assumption 5.9(ii) or Assumptions 6.3 or 6.5 are only satisfied up to an error term ε, which has to be taken into account in the results relying on these assumptions. While we conjecture that in both cases the respective proofs can be modified in order to obtain at least semiglobal practical asymptotic stability of $x^{\varepsilon}_{\mu^{\varepsilon}_{n}}$, we are not aware of respective results in the literature. Hence, this area certainly offers a number of open questions for future research.

Problems

1. Prove that the solution $\varphi(t, 0, x_0, u)$ of (2.6) with $t \in [0, T]$ and constant control function u satisfies the integral equation

$$\varphi(\tau_{i+1}; \tau_0, x_0, u) = \varphi(\tau_i; \tau_0, x_0, u) + \int\limits_{\tau_i}^{\tau_{i+1}} f_c(\varphi(t; \tau_0, x_0, u), u)dt.$$

for all $\tau_i, \tau_{i+1} \in [0, T]$ with $\tau_{i+1} > \tau_i$.
2. Prove that the Euler and the Heun scheme satisfy the Lipschitz condition (11.5) if the vector field f_c satisfies the Lipschitz condition from Assumption 2.4.
3. Given the control system $\dot{x}(t) = x(t) + u(t)$ with stage cost $\ell(x, u) = x^2 + u^2$.

   (a) Consider the NMPC Algorithm 3.1 with $N = 2$ and f generated by the Euler method with $\mathscr{G} = \mathscr{T}$ for (11.14) and (11.15). Prove that the control $\mu_N(x)$ converges tends to zero as $T \to 0$ for each $x \in \mathbb{R}$.
   (b) Consider the same situation as in (a) but with the grid

$$\mathscr{G} := \{\tau_i = iT/k \mid i = 0, \ldots, Nk\}$$

   with $k \in \mathbb{N}$. Does the control value $\mu_N(x)$ converge if $T > 0$ is fixed and k tends to infinity?

4. Consider the differential equation

$$\begin{aligned}\dot{x}_1(t) &= -x_2(t)\\ \dot{x}_2(t) &= x_1(t)\end{aligned}$$

whose solution shall be used to generate a time varying reference for an NMPC algorithm.

   (a) Using a transformation to polar coordinates, compute the analytical solution of the system.
   (b) Show that the numerical solution of the system using Euler's method will deviate from the analytical solution from (a) for every step size $h > 0$ and every initial value $x_0 \neq (0, 0)^\top$.
   (c) Applying the transformation to polar coordinates, show that the occurring error from (b) can be avoided if the resulting differential equation is solved using Euler's method.

5. Consider the continuous time control system

$$\begin{aligned}\dot{x}_1(t) &= -x_2(t) + v(t)\\ \dot{x}_2(t) &= x_1(t)\end{aligned}$$

where u shall be computed via NMPC to track the (exact) time varying reference solution from Problem 4.

(a) Show that this system is (uniformly) asymptotically controllable in the sense of Definition 4.2 for control functions which are piecewise constant on each interval $[iT, (i+1)T)$ for arbitrary sampling time $T > 0$.

(b) Consider the approximate discrete time system (11.12) with $\tilde{\varphi}$ obtained from applying the Euler method with step size $h = T/k$ for arbitrary $k \in \mathbb{N}$ to the (non transformed) differential equation. Show that this approximate system is not asymptotically controllable regardless how $T > 0$ and $k \in \mathbb{N}$ are chosen.

Hint for (b): A necessary condition for asymptotic controllability is that the reference is a solution of the system.

References

1. Braess, D.: Finite Elements, 3rd edn. Cambridge University Press, Cambridge (2007) (Theory, fast solvers, and applications in elasticity theory. Translated from the German by Larry L. Schumaker)
2. Deuflhard, P., Bornemann, F.: Scientific Computing with Ordinary Differential Equations. Texts in Applied Mathematics, vol. 42. Springer, New York (2002) (Translated from the1994 German original by Werner C. Rheinboldt)
3. Grüne, L.: Asymptotic Behavior of Dynamical and Control Systems under Perturbation and Discretization. Lecture Notes in Mathematics, vol. 1783. Springer, Berlin (2002)
4. Grüne, L., Nešić, D.: Optimization based stabilization of sampled-data nonlinear systems via their approximate discrete-time models. SIAM J. Control Optim. **42**, 98–122 (2003)
5. Gyurkovics, E., Elaiw, A.M.: Stabilization of sampled-data nonlinear systems by receding horizon control via discrete-time approximations. Automatica **40**(12), 2017–2028 (2004)
6. Gyurkovics, E., Elaiw, A.M.: Conditions for MPC based stabilization of sampled-data nonlinear systems via discrete-time approximations. In: Findeisen, R., Allgöwer, F., Biegler, L.T. (eds.) Assessment and Future Directions of Nonlinear Model Predictive Control. Lecture Notes in Control and Information Sciences, vol. 358, pp. 35–48. Springer, Berlin (2007)
7. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations. II, Springer Series in Computational Mathematics, vol. 14, 2nd edn. Springer, Berlin (1996)
8. Hairer, E., Nørsett, S.P., Wanner, G.: Solving Ordinary Differential Equations. I, 2nd edn. Springer Series in Computational Mathematics, vol. 8, 2nd edn. Springer, Berlin (1993)
9. LeVeque, R.J.: Finite Difference Methods for Ordinary and Partial Differential Equations. SIAM, Philadelphia (2007)
10. 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)
11. Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis. Texts in Applied Mathematics, vol. 12, 3rd edn. Springer, New York (2002) (Translated from the German by R. Bartels, W. Gautschi and C. Witzgall)
12. Tröltzsch, F.: Optimal Control of Partial Differential Equations. Graduate Studies in Mathematics, vol. 112. American Mathematical Society, Providence (2010) (Theory, methods and applications. Translated from the 2005 German original by Jürgen Sprekels)