Chapter 2
Background on Nonlinear Systems, Control, and Optimization

This chapter provides a brief review of several concepts that are used throughout this book. The first section presents the notation. In the second section, stability of nonlinear systems is discussed followed by a brief overview of stabilization (control) of nonlinear systems. For a more detailed and complete overview of stability and control of nonlinear systems, the reader is referred to, for example, the classical textbooks [1, 2]. In the last section, a review of nonlinear and dynamic optimization concepts is presented.

2.1 Notation

The set of real numbers is denoted by $\mathbb{R}$, while the set of integers is denoted by $\mathbb{I}$. The symbol $\mathbb{R}_{\geq 0}$ ($\mathbb{I}_{\geq 0}$) is used to denote positive reals (integers), and $\mathbb{R}^n$ is an n-dimensional real (Euclidean) space. The variable t where $t \in \mathbb{R}$ will typically be reserved for time and thus, the notation $x(t) \in \mathbb{R}^n$ represents a time-dependent vector. The symbol $|\cdot|$ denotes the Euclidean norm of a vector, i.e., $|x| = \sqrt{x^T x}$ where $x \in \mathbb{R}^n$ and x^T denotes the transpose of x, and $|\cdot|_Q^2$ denotes the square of a weighted Euclidean norm of a vector, i.e., $|x|_Q^2 = x^T Q x$ where Q is a weighting positive definite matrix. A square diagonal matrix with diagonal elements equal to the elements of a vector v and off-diagonal elements equal to zero is written as $\text{diag}(v)$. An infinite sequence is denoted by $\{t_k\}_{k\geq 0}$, while a finite sequence is written as $\{t_i\}_{i=0}^N$ which describes the sequence: $t_0, t_1, \ldots, t_{N-1}, t_N$.

With regard to functions, a function, $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, is said to be positive definite with respect to $\bar{x} \in \mathbb{R}^n$ if $V(x) > 0$ for all $x \in \mathbb{R}^n$ except for $\bar{x}$ when $V(\bar{x}) = 0$. When a function is positive definite with respect to the origin ($\bar{x} = 0$), the function may be referred to as positive definite, and the distinction that it is positive definite with respect to the origin is omitted. A function, $V : \mathbb{R}^n \to \mathbb{R}_{\leq 0}$, is negative definite (with respect to the origin) if $-V$ is positive definite. A continuous function $\alpha : [0, a) \to \mathbb{R}_{\geq 0}$ is said to be of class $\mathcal{K}$ if it is strictly increasing and

M. Ellis et al., *Economic Model Predictive Control*,
Advances in Industrial Control, DOI 10.1007/978-3-319-41108-8_2

$\alpha(0) = 0$, and it is of class $\mathscr{K}_\infty$ if it is of class $\mathscr{K}$, $a = \infty$, and $\alpha(r) \to \infty$ as $r \to \infty$, i.e., it is radially unbounded. A function $\beta : [0, a) \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be of class-$\mathscr{KL}$ if, for each fixed t, the mapping $\beta(s, t)$ is of class-$\mathscr{K}$ with respect to s and for each fixed s, the mapping $\beta(s, t)$ is non-increasing with respect to t and $\beta(s, t) \to 0$ as $t \to \infty$. The family of piecewise constant, right-continuous functions with period Δ is denoted as $S(\Delta)$. With a slight abuse of notation, we will say $u(\cdot) \in S(\Delta)$ (or simply, $u \in S(\Delta)$) when the vector-valued function $u : [0, N\Delta) \to \mathbb{R}^m$, $u : t \mapsto u(t)$, may be described by

$$u(t) = \bar{u}_i, \text{ for } t \in [i\Delta, (i+1)\Delta)$$

for $i = 0, 1, \ldots, N-1$ where $\Delta > 0$ is the period and $\bar{u}_i \in \mathbb{R}^m$; the appropriate domain of the function u will be implied by the context. The floor and ceiling functions, denoted as $\lfloor a \rfloor$ and $\lceil a \rceil$ for a scalar $a \in \mathbb{R}$, respectively, are the largest integer not greater than a and the smallest integer not less than a, respectively.

The set Ω_r is a level set, also referred to as a level surface or sub-level set in other contexts, of a scalar-valued positive definite function: $\Omega_r := \{x \in \mathbb{R}^n : V(x) \leq r\}$ where $r > 0$. A ball of radius $R > 0$ is given by $B_R := \{x \in \mathbb{R}^n : |x| \leq R\}$. The notation $B \setminus A$ denotes the relative complement of the set A in B, i.e., $B \setminus A = \{x \in B : x \notin A\}$. Finally, for algorithms, the notation $j \leftarrow j + 1$ is used to denote that at the next time step or at the next iteration, the index j is incremented by one.

2.2 Stability of Nonlinear Systems

First, unforced nonlinear systems are considered to present some definitions and stability properties. Specifically, consider the following class of time-invariant nonlinear systems, which is described by the following system of first-order nonlinear ordinary differential equations (ODEs):

$$\dot{x} = f(x) \tag{2.1}$$

where $x \in D \subset \mathbb{R}^n$, $f : D \to \mathbb{R}^n$ is a locally Lipschitz map from a domain $D \subset \mathbb{R}^n$ to $\mathbb{R}^n$. The vector x describes the current state of the system. Thus, x is referred to as the *state* vector, and the space $\mathbb{R}^n$ is referred to as *state-space*. The initial condition of system of Eq. 2.1 is given by $x_0 \in D$, i.e., $x(t_0) = x_0$ where $t_0 \in \mathbb{R}$ is the initial time.

The solution of Eq. 2.1 starting from x_0 at time t_0, is denoted as $x(t, x_0, t_0)$ for $t \in [t_0, t_1]$ with $x(t_0, x_0, t_0) = x_0$ and where $t_1 > t_0$ is the maximal time that the solution exists. The initial time may be taken to be zero with no loss of generality. The solution of Eq. 2.1 is also referred to as the state trajectory, and with slight abuse of notation, the notation of the solution of Eq. 2.1 at time $t \geq t_0$ may be abbreviated to $x(t)$. Two of most fundamental properties of the system of Eq. 2.1 are the existence

and uniqueness of a solution to the system of Eq. 2.1 for a given initial condition. If it can be shown that every solution lies in some compact set $\mathbb{X} \subset D$ for all $t \geq t_0$, then a unique solution is guaranteed for all $t \geq t_0$, e.g., [2].

Owing to the fact that the vector field, f, of Eq. 2.1 is nonlinear, the system may possess multiple isolated equilibrium points. Without loss of generality, the origin $x = 0$ is taken to be an equilibrium point of the system of Eq. 2.1, i.e., $f(0) = 0$. If the origin is not the equilibrium point of interest, deviation variables may be introduced such that the origin of the shifted coordinate system is the equilibrium point. For example, consider the system $\dot{x} = f(x)$ with an equilibrium $x_s \neq 0$ ($f(x_s) = 0$). Defining a shifted state $z := x - x_s$, the system may be rewritten in the following coordinates:

$$\dot{z} = f(z + x_s) =: g(z) \tag{2.2}$$

where the equilibrium point of the shifted system is $z = 0$ and $g(0) = 0$.

Within the context of the system of Eq. 2.1, stability of solutions is considered. In particular, the stability of the solution $x \equiv 0$ is considered using Lyapunov stability concepts. The origin of Eq. 2.1 is

- *stable* if, for each $\varepsilon > 0$, there is $\delta(\varepsilon) > 0$ such that

$$|x(0)| < \delta \Rightarrow |x(t)| < \varepsilon, \forall\, t \geq 0 \tag{2.3}$$

- *unstable* if it is not stable
- *locally asymptotically stable* if it is stable and δ may be chosen such that

$$|x(0)| < \delta \Rightarrow \lim_{t \to \infty} |x(t)| = 0 \tag{2.4}$$

- *globally asymptotically stable* if it is stable and $|x(t)| \to 0$ as $t \to 0$ for all $x(0) \in \mathbb{R}^n$
- *locally exponentially stable* if there exist positive real constants δ, c, and λ such that all solutions of Eq. 2.1 with $|x(0)| \leq \delta$ satisfy the inequality:

$$|x(t)| \leq c|x(0)|e^{-\lambda t} \quad \forall\, t \geq 0 \tag{2.5}$$

- *globally exponentially stable* if there exist positive real constants c, and λ such that all solutions of Eq. 2.1 satisfy the inequality:

$$|x(t)| \leq c|x(0)|e^{-\lambda t} \quad \forall\, t \geq 0 \tag{2.6}$$

  for all $x(0) \in \mathbb{R}^n$.

Since the system of Eq. 2.1 is time-invariant, the stability properties above are *uniform*; that is, they do not depend on the initial time. The stability definitions may be written in equivalent forms using so-called comparison functions.

The stability definitions are restated using comparison functions.

Lemma 2.1 ([2, Lemma 4.5]) *The equilibrium point $x = 0$ of Eq. 2.1 is*

- *stable if and only if there exist $\alpha \in \mathscr{K}$ and a positive constant c, such that*

$$|x(t)| \leq \alpha(|x(0)|) \tag{2.7}$$

  for all $t \geq 0$ and $|x(0)| < c$.
- *locally asymptotically stable if and only if there exist $\beta \in \mathscr{KL}$ and a positive constant c such that*

$$|x(t)| \leq \beta(|x(0)|, t) \tag{2.8}$$

  for all $t \geq 0$ and $|x(0)| < c$.
- *globally asymptotically stable if and only if there exist $\beta \in \mathscr{KL}$ such that*

$$|x(t)| \leq \beta(|x(0)|, t) \tag{2.9}$$

  for all $t \geq 0$ and $x(0) \in \mathbb{R}^n$.

When the origin is asymptotically stable, the state-space set of initial conditions where the solution to Eq. 2.1 will asymptotically converge to the origin is of interest. This gives rise to the notion of the *domain of attraction*, which is the set $D = \{x_0 \in \mathbb{R}^n : \lim_{t\to\infty} x(t, t_0, x_0) = 0\}$.

A weaker notion of stability than asymptotic and exponential stability of the origin is boundedness of the solution. Specifically, the solutions of Eq. 2.1 are

- *bounded* if there exists a positive constant c and for every $a \in (0, c)$, there is $\beta(a) > 0$ such that

$$|x(0)| \leq a \Rightarrow |x(t)| \leq \beta, \forall\, t \geq 0 \tag{2.10}$$

- *ultimately bounded* with ultimate bound b if there exist positive constants b and c and for every $a \in (0, c)$ there is $T(a, b) \geq 0$ such that

$$|x(0)| \leq a \Rightarrow |x(t)| \leq b, \ \forall\, t \geq t_0 + T \tag{2.11}$$

For practical systems, global stability properties are often not relevant owing to system constraints. Therefore, we extend the stability concepts to the case where the state of Eq. 2.1 is constrained to be in the set $\tilde{\mathbb{X}} \subset \mathbb{R}^n$. We need the following definition to state the stability properties of the constrained system of Eq. 2.1.

Definition 2.1 A set M is said to be *positively invariant set* with respect to the system of Eq. 2.1 if

$$x(0) \in M \Rightarrow x(t) \in M, \quad \forall\, t \geq 0$$

We will also use the term *forward invariant set* to refer to a positively invariant set. Consider a set $\mathbb{X} \subseteq \tilde{\mathbb{X}}$ to be an positively invariant set for the system of Eq. 2.1 that contains the origin in its interior. Then, the origin is, e.g., [3]:

- *stable* in $\mathbb{X}$ if, for each $\varepsilon > 0$, there is $\delta(\varepsilon) > 0$ such that $B_\delta \subseteq \mathbb{X}$ and

$$|x(0)| < \delta \Rightarrow |x(t)| < \varepsilon, \forall\, t \geq 0 \tag{2.12}$$

- *locally attractive* in $\mathbb{X}$ if there exists a $\eta > 0$ such that $x \in B_\eta \subseteq \mathbb{X}$ implies $|x(t)| \to 0$ as $t \to \infty$
- *attractive* in $\mathbb{X}$ if $|x(t)| \to 0$ as $t \to \infty$ for all $x(0) \in \mathbb{X}$
- *locally asymptotically stable* in $\mathbb{X}$ if it is stable and locally attractive
- *asymptotically stable* in $\mathbb{X}$ if it is stable and attractive
- *locally exponentially stable* in $\mathbb{X}$ if there exist $\eta > 0$, $c > 0$, and $\gamma > 0$ such that

$$|x(t)| \leq c|x(0)|e^{-\lambda t} \quad \forall\, t \geq 0 \tag{2.13}$$

  for all $x(0) \in B_\eta \subseteq \mathbb{X}$
- *exponentially stable* with a region of attraction $\mathbb{X}$ if there exist $c > 0$, and $\gamma > 0$ such that

$$|x(t)| \leq c|x(0)|e^{-\lambda t} \quad \forall\, t \geq 0 \tag{2.14}$$

  for all $x(0) \in \mathbb{X}$.

2.2.1 Lyapunov's Direct Method

For nonlinear systems, stability of the equilibrium points may be characterized in the sense of Lyapunov's direct method. Lyapunov's direct second method uses a scalar-valued positive definite function whose time-derivative is negative (semi-)definite along the state trajectory.

Theorem 2.1 (Lyapunov Stability Theorem, c.f. [2, Theorem 4.1]) *Let $x = 0$ be an equilibrium point for Eq. 2.1 and $D \subset \mathbb{R}^n$ be a domain containing the origin ($x = 0$). Let $V : D \to \mathbb{R}$ be a continuously differentiable positive definite function such that*

$$\dot{V}(x) \leq 0 \tag{2.15}$$

for all $x \in D$. Then, $x = 0$ is stable. If

$$\dot{V}(x) < 0 \tag{2.16}$$

for all $x \in D \setminus \{0\}$, then $x = 0$ is asymptotically stable.

A continuously differentiable positive definition function V as in Theorem 2.1 is called a *Lyapunov function*. The time-derivative of V along the state trajectory x is given by:

$$\dot{V}(x) = \frac{\partial V(x)}{\partial x}\dot{x} = \frac{\partial V(x)}{\partial x} f(x). \tag{2.17}$$

Theorem 2.1 is a sufficient condition for stability and asymptotic stability of the origin. Various converse Lyapunov theorems show that the conditions of Theorem 2.1 are also necessary (under a few additional mild conditions), see, for example, [2, 4–8].

Lyapunov's direct method has an intuitive interpretation by regarding the Lyapunov function as an abstract notion of the total energy of a given system. Specifically, consider any x on the level or Lyapunov surface $V(x) = c$, which is the boundary of the set $\Omega_c = \{x \in \mathbb{R}^n : V(x) \leq c\} \subset D$. When $\dot{V}(x) < 0$ for all $x \in D$, the state trajectory evolves from the boundary of Ω_c to the interior of Ω_c. Over time, the level surface that state trajectory evolves along shrinks to the origin owing to the fact that $\dot{V}(x) < 0$ for all $x \in D$. In other words, the energy of the system decays with time when $\dot{V} < 0$. If, instead, $\dot{V}(x) \leq 0$ for all $x \in D$, this implies that the state trajectory evolves inside the set $\Omega_c \subset D$ without coming out, and the energy over time may only stay the same or decrease (it cannot increase). This in turn means that a trajectory starting from the boundary of Ω_c will stay in the set Ω_c for all time without coming out. In this case, the conclusion that may be made is the origin is stable since the trajectory is contained inside any ball, B_ε, by requiring that the initial state x_0 to lie inside a Lyapunov surface contained in that ball.

2.2.2 LaSalle's Invariance Principle

LaSalle's invariance principle allows for making stronger conclusions about the behavior of solution of Eq. 2.1 when $\dot{V}(x) \leq 0$ for all $x \in D$.

LaSalle's invariance principle states that any state starting in any compact forward invariant subset of D will converge to the largest invariant set where $\dot{V}(x) = 0$.

Theorem 2.2 (LaSalle, c.f. [2, Theorem 4.4]) *Let $\Omega \subset D$ be a compact set that is positively invariant with respect to Eq. 2.1. Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω. Let $E := \{x \in \Omega : \dot{V}(x) = 0\}$ and M be the largest invariant set in E. Then every solution in Ω approaches M as $t \to \infty$.*

A consequence of LaSalle's invariance principle, one may show asymptotic stability of the origin when $M = \{0\}$, i.e., when M is the set containing the point $x = 0$. This result is stated in the following corollary.

Corollary 2.1 (c.f. [2, Corollary 4.1]) *Let $V : D \to \mathbb{R}$ be a continuously differentiable positive definite function on a domain D containing the origin $x = 0$, which is an equilibrium point of Eq. 2.1, such that $\dot{V}(x) \leq 0$ for all $x \in D$. Let $S = \{x \in D : \dot{V}(x) = 0\}$ and suppose that no solution can stay identically in S, other than the trivial solution $x \equiv 0$. Then, the origin is asymptotically stable.*

2.3 Stabilization of Nonlinear Systems

Consider, now, the class of forced nonlinear systems described by the following system of nonlinear ordinary differential equations:

$$\dot{x} = f(x, u, w) \tag{2.18}$$

where $x \in D \subseteq \mathbb{R}^n$ is the state, $u \in \mathbb{U} \subset \mathbb{R}^m$ is the manipulated (control) input, and $w \in \mathbb{R}^l$ is a disturbance. The set of admissible input values $\mathbb{U}$ is assumed to be compact, and the disturbance vector is bounded in the set $\mathbb{W} := \{w \in \mathbb{R}^l : |w| \leq \theta\}$ where $\theta > 0$ bounds the norm of the disturbance vector. Throughout this book, the disturbances vector as in Eq. 2.18 taken to be unknown and un-modeled forcing of the system. Disturbance models, e.g., integrating disturbance models, may readily be incorporated into the model of Eq. 2.18 through augmenting the state vector. However, a complete and thorough discussion of disturbance modeling is beyond the scope of this book and is not considered further. The vector function f is assumed to be locally Lipschitz on $D \times \mathbb{U} \times \mathbb{W}$. Without loss of generality, the origin of the unforced nominal system is assumed to be the equilibrium point of the system of Eq. 2.18, i.e., $f(0, 0, 0) = 0$.

Regarding existence and uniqueness of solutions of the system of Eq. 2.18, first it is important to point out that the input and disturbance trajectories are often not continuous functions of time. In the deterministic framework that we consider, the input and disturbance trajectories require a degree of continuity, and the disturbance may not rigorously be treated as noise. A standing assumption throughout the book is that the disturbance trajectory poses enough continuity to ensure existence of the solution of Eq. 2.18 almost everywhere. In practice such assumption poses little restrictions. For a more complete discussion of conditions that guarantee existence and uniqueness of a solution the interested reader is referred to [9].

2.3.1 Control Lyapunov Functions

The concept of control Lyapunov functions is described, which is utilized in many Lyapunov-based control design techniques. For simplicity of presentation, the case of a system with a single input is presented. Nonetheless, this concept extends to systems with multiple inputs. Thus, consider the following single-input system of the form:

$$\dot{x} = f(x, u) \tag{2.19}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}$, and $f(0, 0) = 0$. The control objective considered is to design a feedback control law $h : D \to \mathbb{U}$ that renders the origin of the closed-loop systems given by:

$$\dot{x} = f(x, h(x)) \tag{2.20}$$

globally asymptotically stable.

One potential approach may be to pick a function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ as a Lyapunov function candidate, and find a control law that guarantees that the time-derivative of the Lyapunov function candidate along the solutions of the closed-loop system of Eq. 2.20 satisfy:

$$\frac{\partial V(x)}{\partial x} f(x, h(x)) \leq -W(x) \tag{2.21}$$

for all $x \in \mathbb{R}^n$ where $W : D \to \mathbb{R}$ is a positive definite function. It may be possible to find a stabilizing control law but Eq. 2.21 may fail to be satisfied for all $x \in \mathbb{R}^n$ because of a poor choice of functions V and W. Therefore, picking a control law that satisfies Eq. 2.21 is a difficult task in general. A system for which a good choice of the functions V and W exist is said to possess a control Lyapunov function.

Definition 2.2 A control Lyapunov function (CLF) for the system of Eq. 2.19 is a smooth positive definite radially unbounded function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ that satisfies:

$$\inf_{u \in \mathbb{R}} \left\{ \frac{\partial V(x)}{\partial x} f(x, u) \right\} < 0, \ \forall\, x \neq 0. \tag{2.22}$$

Equation 2.22 is necessary and sufficient for the existence of a control law satisfying Eq. 2.21 [10]. Also, it may be shown that the existence of a CLF is equivalent to global asymptotic stabilizability.

For control-affine systems of the form:

$$\dot{x} = f(x) + g(x)u, \tag{2.23}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^n$, and $f(0) = 0$. Using the Lie derivative notation:

$$L_f V(x) := \frac{\partial V(x)}{\partial x} f(x),$$
$$L_g V(x) := \frac{\partial V(x)}{\partial x} g(x),$$

the CLF condition of Eq. 2.21 is given by:

$$L_f V(x) + L_g V(x) u \leq -W(x) \tag{2.24}$$

for all $x \in \mathbb{R}^n$. Note that Eq. 2.24 may be satisfied only if:

$$L_g V(x) = 0 \Rightarrow L_f V(x) < 0, \ \forall\, x \neq 0 \tag{2.25}$$

If V is a CLF for the system of Eq. 2.23, then one choice of stabilizing control law is given by Sontag's formula [11]:

$$h(x) = \begin{cases} -\dfrac{L_f V(x) + \sqrt{(L_f V(x))^2 + (L_g V(x))^4}}{(L_g V(x))^2} L_g V(x), & L_g V(x) \neq 0 \\ 0, & L_g V(x) = 0 \end{cases} \tag{2.26}$$

In this case, the positive definite function, W is given by:

$$W(x) = \sqrt{(L_f V(x))^2 + (L_g V(x))^4} > 0, \ x \neq 0 \tag{2.27}$$

While the construction of CLFs is difficult for the general class of nonlinear systems of Eq. 2.19, systematic methods exist for several important classes of nonlinear systems that allow for the construction of CLFs.

2.3.2 Stabilization of Nonlinear Sampled-Data Systems

In the subsequent chapters, EMPC methods are considered. The Lyapunov-based EMPC methods that are presented take advantage of an explicit stabilizing feedback controller. The explicit controller satisfies the following assumption.

Assumption 2.1 There exists a feedback controller $h(x) \in \mathbb{U}$ with $h(0) = 0$ that renders the origin of the closed-loop system of Eq. 2.18 with $u = h(x)$ and $w \equiv 0$ asymptotically stable for all $x \in D_0$ where D_0 is an open neighborhood of the origin.

There are several methods to design an explicit feedback control law, $h : D \to \mathbb{U}$, that renders the origin of Eq. 2.18 asymptotically stable. Specifically, methodologies for (explicit) feedback control design for nonlinear systems include employing linear feedback control techniques, Lyapunov-based control techniques, and geometric control methods, e.g., [12–16].

Applying converse theorems [2, 4], Assumption 2.1 implies that there exists a continuously differentiable Lyapunov function, $V : D \to \mathbb{R}^n$, for the closed-loop system of Eq. 2.18 with $u = h(x) \in \mathbb{U}$ and $w \equiv 0$ such that the following inequalities hold:

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|), \tag{2.28a}$$

$$\frac{\partial V(x)}{\partial x} f(x, h(x), 0) \leq -\alpha_3(|x|), \tag{2.28b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq \alpha_4(|x|) \tag{2.28c}$$

for all $x \in D$ where D is an open neighborhood of the origin and $\alpha_i, i = 1, 2, 3, 4$ are functions of class $\mathcal{K}$. A level set of the Lyapunov function Ω_ρ, which defines a subset of D (ideally the largest subset contained in D), is taken to be the stability region of the closed-loop system under the controller $h(x)$. Standard techniques

exist for designing a stabilizing control law for various classes of continuous-time nonlinear systems (see, for instance, [1, 2, 13, 15–17] as well as the references contained therein).

While there are no general methods for constructing Lyapunov functions for broad classes of nonlinear systems with constraints, there exists some general methods for constructing Lyapunov functions for certain classes of systems, e.g., Zubov's method [18] and the sum of squares decomposition [19]. Within the context of chemical process control, quadratic Lyapunov functions have been widely used and have been demonstrated to be effective for estimating the region of attraction of a given equilibrium point of a system (see, for example, the numerous examples in [16] as well as the examples of the subsequent chapters of this book).

The explicit controller poses a degree of robustness to disturbances/uncertainty in the sense that when $w \not\equiv 0$, the controller will force the closed-loop state to a small neighborhood of the origin if the bound on the disturbance, θ, is sufficiently small. Moreover, owing to the fact that digital computers are often used in the implementation of controllers, we must also consider the closed-loop stability properties of the controller $h(x)$ applied in a sample-and-hold fashion. When the feedback controller $h(x)$ is applied in a sample-and-hold fashion, the resulting closed-loop system is a nonlinear sampled-data system given by:

$$\dot{x}(t) = f(x(t), h(x(t_k)), w(t)) \tag{2.29}$$

for $t \in [t_k, t_{k+1})$, $t_k = k\Delta$, $k = 0, 1, \ldots$, and $\Delta > 0$ is the sampling period. Regarding the disturbance in Eq. 2.29, in many applications, it is sufficient to take w to be constant over the sampling periods. This is essentially what is done when considering a discrete-time model for a sampled-data system.

Applying standard results on sampled-data systems, e.g., [20–24], it can be shown that when the bound on the disturbances and the sampling period are both sufficiently small the origin is practically stable for all initial conditions in Ω_ρ. More specifically, the state trajectory of Eq. 2.29 starting in Ω_ρ will remain bounded in Ω_ρ and converge to a small compact set containing the origin where it will be maintained thereafter when the bound on the disturbance and the sampling period are sufficiently small. It is important to emphasize that asymptotic stability of the origin of Eq. 2.29 is typically not achieved unless additional conditions hold.

To achieve asymptotic stability of the origin of sampled-data system of Eq. 2.29, a stronger assumption is required. The following assumption and result are stated generally in the sense that no restrictions are placed on the state and input.

Assumption 2.2 There exists a locally Lipschitz feedback controller $u = h(x)$ with $h(0) = 0$ such that the vector field of the closed-loop system $f(x, h(x), 0)$ is continuously differentiable on $\mathbb{R}^n$. Furthermore, the origin of the nominal closed-loop system of Eq. 2.18 ($w \equiv 0$) under the controller $h(x)$ implemented continuously is locally exponentially stable and globally asymptotically stable.

The following theorem characterizes the type of stability achieved when the controller $h(x)$ is applied in a sample-and-hold fashion with a sufficiently small hold

period. The result below extends to a more general setting where asynchronous sampling is considered; see, [25] for this more general version of the following result.

Theorem 2.3 *If Assumption 2.2 holds, then given $R > 0$, there exist $\Delta^* > 0$ and $M, \sigma > 0$ such that for $\Delta \in (0, \Delta^*)$ the nominal closed-loop sampled-data system of Eq. 2.29 with arbitrary initial condition $x(0) = x_0 \in B_R$ satisfies the estimate:*

$$|x(t)| \leq M \exp(-\sigma t)|x_0| \tag{2.30}$$

for all $t \geq 0$.

Proof By virtue of Proposition 4.4 of [26], there exists a C^1 positive definite and radially unbounded function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, constants $\mu,\ \varepsilon > 0$ and a symmetric, positive definite matrix $P \in \mathbb{R}^{n \times n}$ for the nominal closed-loop system of Eq. 2.18 under the controller $h(x)$ implemented continuously such that

$$\frac{\partial V(x)}{\partial x} f(x, h(x), 0) \leq -\mu |x|^2, \quad \text{for all } x \in \mathbb{R}^n, \tag{2.31}$$

$$V(x) = x^T P x, \quad \text{for all } x \in \mathbb{R}^n \text{ with } |x| \leq \varepsilon. \tag{2.32}$$

Let $R > 0$ and define $\hat{\rho} := \max\{V(x) : x \in B_R\}$. By virtue of Eq. 2.32 and the compactness of $\Omega_{\hat{\rho}}$, there exist constants $c_1,\ c_2 > 0$ and $c_4 > 0$ such that:

$$c_1 |x|^2 \leq V(x) \leq c_2 |x|^2, \tag{2.33}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq c_4 |x| \tag{2.34}$$

for all $x \in \Omega_{\hat{\rho}}$. Since f and h are locally Lipschitz mappings with $f(0, 0, 0) = 0$ and $h(0) = 0$, there exist constants $L,\ M > 0$ such that:

$$|f(x, h(z), 0) - f(x, h(x), 0)| \leq L|x - z|, \tag{2.35}$$

$$|f(x, h(z), 0)| \leq M|x| + M|z| \tag{2.36}$$

for all $x,\ z \in \Omega_{\hat{\rho}}$. Let $\Delta^* > 0$ be sufficiently small so that the following inequality holds:

$$c_4 L \frac{2M\Delta^* \exp(M\Delta^*)}{1 - 2M\Delta^* \exp(M\Delta^*)} < \mu \tag{2.37}$$

In order to prove of estimate of Eq. 2.30, it suffices to show that for every initial condition $x(0) \in \Omega_{\hat{\rho}}$ and for every integer $k \geq 0$ it holds that:

$$\frac{\partial V(x(t))}{\partial x} f(x(t), h(x(t_k)), 0) \leq -\frac{q}{2} |x(t)|^2, \tag{2.38}$$

for all $t \in [t_k,\ t_{k+1})$ where

$$q := \mu - c_4 L \frac{2M\Delta^* \exp(M\Delta^*)}{1 - 2M\Delta^* \exp(M\Delta^*)} > 0. \tag{2.39}$$

Using Eqs. 2.33 and 2.38, local exponential stability can be established. The proof of Eq. 2.38 is given below for $k = 0$ and $t \in [0, t_1)$. For every other interval, the proof is similar.

If $x(0) = 0$, then Eq. 2.38 trivially holds (since $x(t) = 0$ for $t \in [0, t_1)$). Therefore, consider the case when $x(0) \neq 0$. The proof is made by contradiction. Suppose that there exists $t \in [0, t_1)$ with

$$\frac{\partial V(x(t))}{\partial x} f(x(t), h(x(0)), 0) > -\frac{q}{2}|x(t)|^2.$$

The case that $x(t)$ is not defined for some $t \in [0, t_1)$ is also covered by this assumption. Define

$$a := \inf \left\{ t \in [0, t_1) : \frac{\partial V(x(t))}{\partial x} f(x(t), h(x(0)), 0) > -\frac{q}{2}|x(t)|^2 \right\}.$$

A standard continuity argument in conjunction with the fact that

$$\frac{\partial V(x(0))}{\partial x} f(x(0), h(x(0)), 0) \leq -\mu|x(0)|^2 < -\frac{q}{2}|x(0)|^2$$

shows that $a \in (0, t_1)$ and that

$$\frac{\partial V(x(t))}{\partial x} f(x(t), h(x(0)), 0) \leq -\frac{q}{2}|x(t)|^2$$

for all $t \in [0, a]$ with $(\partial V(x(a))/\partial x) f(x(a), h(x(0)), 0) = -\frac{q}{2}|x(a)|^2$. Moreover, for all $t \in [0, a]$ the inequality of Eq. 2.38 implies that $V(x(t)) \leq V(x(0)) \leq \hat{\rho}$. Therefore, $x(t) \in \Omega_{\hat{\rho}}$ for all $t \in [0, a]$. Using inequalities Eqs. 2.31, 2.34, 2.35, we obtain:

$$\frac{\partial V(x(t))}{\partial x} f(x(t), h(x(0)), 0) \leq -\mu|x(t)|^2 + c_4 L|x(t)||x(t) - x(0)| \tag{2.40}$$

for all $t \in [0, a]$. Using Eq. 2.36 and since $a \leq t_1 \leq \Delta^*$, a bound on the difference between $x(t)$ and $x(0)$ is obtained:

$$|x(t) - x(0)| \leq 2M\Delta^*|x(0)| + M \int_0^t |x(\tau) - x(0)|\, d\tau \tag{2.41}$$

for all $t \in [0, a]$. Applying the Gronwall-Bellman lemma to Eq. 2.41, we obtain:

$$|x(t) - x(0)| \leq 2M\Delta^* \exp(M\Delta^*)|x(0)| \tag{2.42}$$

for all $t \in [0, a]$. Using Eq. 2.42, the triangle inequality and the fact that

$$2M\Delta^* \exp(M\Delta^*) < 1$$

which is implied by Eq. 2.37, we get for all $t \in [0, a]$:

$$|x(t) - x(0)| \leq \frac{2M\Delta^* \exp(M\Delta^*)}{1 - 2M\Delta^* \exp(M\Delta^*)}|x(t)|. \tag{2.43}$$

Thus, using Eqs. 2.40, 2.43 and the fact that

$$q := \mu - c_4 L \frac{2M\Delta^* \exp(M\Delta^*)}{1 - 2M\Delta^* \exp(M\Delta^*)} > 0$$

we get for all $t \in [0, a]$:

$$\frac{\partial V(x(t))}{\partial x} f(x(t), h(x(0)), 0) \leq -q|x(t)|^2. \tag{2.44}$$

Consequently, we must have:

$$\frac{\partial V(x(a))}{\partial x} f(x(a), h(x(0)), 0) \leq -q|x(a)|^2 \leq -\frac{q}{2}|x(a)|^2. \tag{2.45}$$

Since $(\partial V(x(a))/\partial x) f(x(a), h(x(0)), 0) = -\frac{q}{2}|x(a)|^2$, we get $x(a) = 0$. However, this contradicts Eq. 2.42 (since Eq. 2.42 in conjunction with the fact that

$$2M\Delta^* \exp(M\Delta^*) < 1$$

implies that $|x(a) - x(0)| < |x(0)|$), which completes the proof.

Explicit feedback controllers that may be designed to satisfy Assumption 2.2 include, for example, feedback linearizing controller and some Lyapunov-based controllers [2, 15]. Owing to the input constraints, it may not be possible to design a controller $h(x)$ that achieves global asymptotic stability of the origin. In this case, we must modify the assumption which is considered in the following corollary.

Corollary 2.2 *Suppose there exists a locally Lipschitz feedback controller $u = h(x)$ with $h(0) = 0$ for the system of Eq. 2.18 that renders the origin of the nominal closed-loop system under continuous implementation of the controller $h(x)$ locally exponentially stable. More specifically, there exist constants $\rho > 0$, $c_i > 0$, $i = 1, 2, 3, 4$ and a continuously differentiable Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}_+$ such that the following inequalities hold:*

$$c_1 |x|^2 \leq V(x) \leq c_2 |x|^2 , \tag{2.46a}$$

$$\frac{\partial V(x)}{\partial x} f(x, h(x), 0) \leq -c_3 |x|^2 , \tag{2.46b}$$

$$\left| \frac{\partial V(x)}{\partial x} \right| \leq c_4 |x| , \tag{2.46c}$$

for all $x \in \Omega_\rho$. There exists $\Delta^* > 0$ and $M, \sigma > 0$ such that for all $\Delta \in (0, \Delta^*)$ the estimate of Eq. 2.30 holds for the nominal closed-loop sampled-data system of Eq. 2.29 with arbitrary initial condition $x(0) \in \Omega_\rho$.

Proof The proof follows along the same lines of Theorem 2.3 and shows that V is a Lyapunov function for the closed-loop sampled-data system and takes advantage of the compactness of the set Ω_ρ to establish an exponentially decaying estimate for the state trajectory of the closed-loop sample-data system for any initial condition $x(0) \in \Omega_\rho$.

Remark 2.1 Sufficient conditions such that there exists a function V satisfying the inequalities of Eq. 2.46 are when $x = 0$ is a locally exponentially stable (LES) equilibrium point for the closed-loop system $\dot{x} = f(x, h(x), 0)$ and the mapping $f(x, h(x), 0)$ is continuously differentiable on $\mathbb{R}^n$. Indeed, by Lemma 8.1 in [2] the region of attraction A of $x = 0$ is an open, connected, invariant set. Let $r > 0$ be such that the set $S = \{x \in \mathbb{R}^n : |x| \leq r\}$ is contained in the region of attraction A. Then LES and compactness of S imply that an exponential bound holds for the solutions of the closed-loop system $\dot{x} = f(x, h(x), 0)$ with initial conditions $x(0) \in S$. It follows from Theorem 4.14 in [2] that there exists a Lyapunov function V for the closed-loop system $\dot{x} = f(x, h(x), 0)$ that satisfies inequalities of Eq. 2.46 for certain constants $c_1, c_2, c_3, c_4 > 0$ and for all $x \in \text{int}(S)$ ($\text{int}(S)$ denotes the interior of S). Let $R < r$ be an arbitrary positive number and define $V(x) = V(\text{Proj}(x))$ for all $x \in \mathbb{R}^n$, where $\text{Proj}(x)$ denotes the projection on the closed ball of radius R centered at $x = 0$. Then all inequalities of Eq. 2.46 hold with arbitrary $\rho < c_1 R^2$.

2.3.3 Tracking Model Predictive Control

Designing an explicit feedback control such as one that satisfies Assumption 2.1 to stabilize the origin of the system of Eq. 2.18 has many advantages such as it may be shown to possess robustness to disturbances and sample-and-hold or discrete-time implementation. However, the most significant drawback of such an approach to controller design is that performance considerations and system constraints are not explicitly handled in a general framework. For example, consider that the system of Eq. 2.18 is subject to the following constraint:

$$(x(t), u(t)) \in \mathbb{Z} \tag{2.47}$$

for all $t \geq 0$ where $\mathbb{Z}$ is assumed to be compact, which accounts, for example, state, input, and other process constraints. One such mathematical framework that allows for one to explicitly account for these considerations is optimization. This is the framework employed in model predictive control.

Tracking model predictive control (MPC), also referred to as receding horizon control, is an on-line optimization-based control technique that optimizes a performance index or cost function over a prediction horizon by taking advantage of a dynamic nominal process model, i.e., Eq. 2.18 with $w \equiv 0$, while accounting for system/process constraints, e.g., [27–32]. The main objective of tracking MPC is to steer the system to and maintain operation thereafter at the economically optimal steady-state or the economically optimal trajectory computed in an upper-layer optimization problem (real-time optimization). To manage the trade-off between the speed of response of the closed-loop system and the amount of control energy required to generate the response, MPC is typically formulated with a quadratic objective function which penalizes the deviations of the state and inputs from their corresponding optimal steady-state or reference values over the prediction horizon. Within this book, the term *tracking MPC* will refer to both regulation MPC or MPC that forces a system to steady-state and tracking MPC or MPC that forces a system track a reference trajectory.

The tracking MPC problem is given by the following dynamic optimization problem:

$$\min_{u \in S(\Delta)} \quad \int_{t_k}^{t_{k+N}} l_T(\tilde{x}(\tau), u(\tau))\, d\tau \tag{2.48a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t), u(t), 0) \tag{2.48b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{2.48c}$$

$$(x(t), u(t)) \in \mathbb{Z}, \ \forall\, t \in [t_k,\ t_{k+N}) \tag{2.48d}$$

where

$$l_T(x, u) = |x|^2_{Q_c} + |u|^2_{R_c} \tag{2.49}$$

and Q_c is a positive semidefinite matrix and R_c is a positive definite matrix that manage the trade-off between the speed of response and the cost of control action. Given that the cost function is positive definite with respect to the origin, which is the steady-state of the system of Eq. 2.18, the global minimum of the cost function occurs at the optimal steady-state. The stage cost function of Eq. 2.49 may be readily extended to be positive definite with respect to a reference trajectory. The state trajectory $\tilde{x}$ is the predicted evolution of the state using the nominal dynamic model ($w \equiv 0$) of Eq. 2.18 under the piecewise constant input profile computed by the MPC. The initial condition on the dynamic model are given in Eq. 2.48c which are obtained at each sampling period through a measurement. The constraints of Eq. 2.48d are the system/process constraints, e.g., input and state constraints.

MPC is the resulting control law when the problem of Eq. 2.48 computes the control action applied to the system in a receding horizon fashion. Specifically,

at the sampling time t_k, the problem of Eq. 2.48 is initialized with a state feedback measurement and the problem is solved. The optimal input trajectory, i.e., the optimal solution, is denoted by $u^*(t|t_k)$ and defined for $t \in [t_k, t_{k+N})$. A brief overview of methods for solving such dynamic optimization problems of the form of Eq. 2.48 is given in Sect. 2.4.4. The (constant) input trajectory $u^*(t|t_k)$ defined for $t \in [t_k, t_{k+1})$, which may be denoted by $u^*(t_k|t_k)$, is send to the control actuators to be implemented on the system for $t \in [t_k, t_{k+1})$. At t_{k+1}, the problem is re-initialized with an updated measurement and the problem of Eq. 2.48 is re-solved by shifting the horizon one sampling period into the future. Thus, the resulting input trajectory under MPC is given by:

$$u(t) = u^*(t_k|t_k), \ \forall \, t \in [t_k, t_{k+1}). \tag{2.50}$$

When the prediction horizon N is finite, it is well-known that the MPC scheme of Eq. 2.48 may not be stabilizing, e.g., [29]. To handle guaranteed stabilization of the closed-loop system when N is finite, various constraints and variations to the cost function may be made to guarantee stability such as using a sufficiently long prediction horizon, incorporating terminal constraints and/or a terminal cost in the optimization problem, or the use of contractive constraints (see, for example, [29], and the references therein).

2.3.4 *Tracking Lyapunov-Based MPC*

To address stability of the closed-loop system with tracking model predictive control (MPC) and recursive feasibility, one tracking MPC technique unites the stability and robustness properties of the Lyapunov-based controller, i.e., a control law that satisfies Assumption 2.1, with the optimal control properties of model predictive control (MPC) [21, 33–35]. The resulting tracking MPC is called Lyapunov-based MPC (LMPC) and is characterized by the following optimization problem:

$$\min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} l_T(\tilde{x}(\tau), u(\tau)) \, d\tau \tag{2.51a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t), u(t), 0) \tag{2.51b}$$

$$\tilde{x}(t_k) = x(t_k) \tag{2.51c}$$

$$u(t) \in \mathbb{U}, \ \forall \, t \in [t_k, t_{k+N}) \tag{2.51d}$$

$$\frac{\partial V(x(t_k))}{\partial x} f(x(t_k), u(t_k), 0) \leq \frac{\partial V(x(t_k))}{\partial x} f(x(t_k), h(x(t_k)), 0) \tag{2.51e}$$

where $\tilde{x}$ is the predicted state trajectory over the prediction horizon with the computed input trajectory by the LMPC, and $N > 0$ is the number of sampling periods in the finite prediction horizon. The constraint of Eq. 2.51d is the input constraint, while the constraint of Eq. 2.51e is a contractive constraint for guaranteed stability that is explained further below.

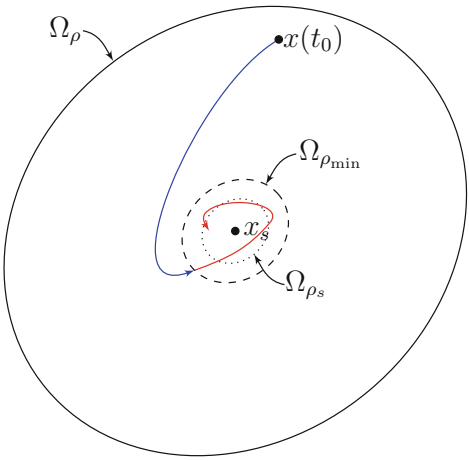

Fig. 2.1 A state-space illustration of a closed-loop state trajectory under LMPC

Specifically, the constraint of Eq. 2.51e ensures that the LMPC computes a control action for the first sampling period that decreases the Lyapunov function by at least the rate achieved by the Lyapunov-based controller at t_k. The Lyapunov-based constraint of Eq. 2.51e is a contractive constraint and ensures that the Lyapunov function decays until the closed-loop state converges to a small neighborhood of steady-state. Moreover, from the Lyapunov-based constraint, the LMPC inherits the closed-loop stability and robustness properties and the stability region Ω_ρ of the Lyapunov-based controller in the sense that for any initial condition $x(0) \in \Omega_\rho$, the closed-loop system state is guaranteed to converge to a small neighborhood of the origin and the optimization problem of Eq. 2.51 is guaranteed to be feasible.

Figure 2.1 gives an illustration of the closed-loop state trajectory under LMPC. The state trajectory starts in $\Omega_\rho \setminus \Omega_{\rho_s}$ whereby in this region the Lyapunov function is guaranteed to decay with time. Once the state trajectory converges to Ω_{ρ_s}, the Lyapunov function is no longer guaranteed to decay owing to the sampling-and-hold implementation of LMPC and the effect of persistent disturbances. However, the state will be maintained in a small forward invariant set $\Omega_{\rho_{\min}} \supset \Omega_{\rho_s}$ when the the sampling period and the bound on the disturbance are sufficiently small.

2.4 Brief Review of Nonlinear and Dynamic Optimization

Although this book does not directly deal with developing nonlinear and dynamic optimization techniques, a brief review of nonlinear optimization (also commonly referred to as nonlinear programming) and dynamic optimization/optimal control concepts is provided in this section. The presentation is meant to demonstrate to the reader how one may approach obtaining a solution to dynamic optimization problems

which is required to understand the concepts presented in the subsequent chapters. This section includes definitions, optimality conditions, nonlinear optimization solution techniques, and practical dynamic optimization strategies. For a comprehensive and detailed presentation on optimization methods, the reader is referred to one of the many textbooks on the topic, e.g., [36–40]. For more details relating to dynamic optimization or optimal control, see, for instance, [37, 41–43].

2.4.1 Notation

For a vector $x \in \mathbb{R}^n$, $x \geq 0$ means component-wise inequality, i.e., $x_i \geq 0$, $i = 1, \ldots, n$. The transpose of a vector or matrix is denoted $(\cdot)^T$, e.g., the transpose of $x \in \mathbb{R}^n$ is denoted x^T. The *gradient* (n-dimensional vector) of a differentiable scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ evaluated at $x \in \mathbb{R}^n$ is denoted as

$$\nabla f(x) := \left[\frac{\partial f(x)}{\partial x_1} \; \frac{\partial f(x)}{\partial x_2} \cdots \frac{\partial f(x)}{\partial x_n} \right]^T.$$

When a scalar-valued differentiable function has multiple arguments, for example, $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $f : (x, y) \mapsto f(x, y)$, the notation $\nabla_x f(x, y)$ may be used to denote the gradient of f with respect to x. For a vector-valued differentiable function $g : \mathbb{R}^n \to \mathbb{R}^m$, the *gradient matrix* is an $n \times m$ matrix whose ith column is the gradient vector $\nabla g_i(x)$ ($i = 1, \ldots, m$):

$$\nabla g(x) := \left[\nabla g_1(x) \cdots \nabla g_m(x) \right],$$

while the *Jacobian* of g is

$$\frac{\partial g(x)}{\partial x} := \begin{bmatrix} \frac{\partial g_1(x)}{\partial x_1} & \cdots & \frac{\partial g_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m(x)}{\partial x_1} & \cdots & \frac{\partial g_m(x)}{\partial x_n} \end{bmatrix}.$$

With the definitions above, the gradient matrix is the transpose of the Jacobian:

$$\nabla g(x)^T = \frac{\partial g(x)}{\partial x}.$$

The *Hessian matrix* of a scalar-valued differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is denoted as

$$\nabla_{xx} f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

2.4.2 *Definitions and Optimality Conditions*

Consider the following nonlinear constrained optimization problem:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ \text{s.t.} & g(x) \leq 0 \\ & h(x) = 0 \end{array} \tag{2.52}$$

where $x \in \mathbb{R}^n$ is the decision variable or the unknown variable to be determined that minimizes the objective function ($f : \mathbb{R}^n \to \mathbb{R}$) while satisfying the inequality constraints ($g : \mathbb{R}^n \to \mathbb{R}^{n_g}$) and the equality constraints ($h : \mathbb{R}^n \to \mathbb{R}^{n_h}$). The objective function is also referred to as the cost function or cost functional in the context of dynamic optimization problems. Here and elsewhere in the book, the usage of the notation "min" in Eq. 2.52 is more aligned with that typically found in the engineering literature, that is, it refers to the greatest lower bound or infimum of $f(x)$ over $\mathbb{X}$. Nevertheless, in the application studies contained in this book, the optimization problems are formulated in a manner that guarantee that they may be numerically solved in the sense that $f^* = \inf_{x \in \mathbb{X}} f(x)$ where $\mathbb{X} = \{x \in \mathbb{R}^n : g(x) \leq 0,\ h(x) = 0\}$ is non-empty, f^* is finite, and there exists a vector $x^* \in \mathbb{X}$ where the minimum is attained. The issue of the existence of a minimizing vector will not be treated in depth.

The functions f, g, and h are assumed to be continuously differentiable. A vector $x \in \mathbb{R}^n$ is said to be a *feasible* point if $g(x) \leq 0$ and $h(x) = 0$. The set of all feasible points or the *feasible set* to the problem of Eq. 2.52 is the set $\mathbb{X} \subseteq \mathbb{R}^n$. For the problem of Eq. 2.52 to be meaningful, the feasible set must be non-empty. Otherwise, the problem of Eq. 2.52 is said to be *infeasible*. Feasibility of the optimization problems formulated in this book will be carefully examined which is a crucial property for control purposes. A vector $x^* \in \mathbb{X}$ is said to be a *local minimum* if there exists $\varepsilon > 0$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{X}$ with $|x - x^*| < \varepsilon$. A vector $x^* \in \mathbb{X}$ is said to be a *global minimum* if $f(x^*) \leq f(x)$ for all $x \in \mathbb{X}$. A local or global minimum is called a *strict* minimum if the inequalities are strict for all $x \neq x^*$. For a given local or global minimum $x^* \in \mathbb{X}$, $f(x^*)$ is called the local or global optimal objective function value or optimal value.

In subsequent chapters, non-convex dynamic nonlinear optimization problems will be considered. It is sufficient, for purposes of this book, to understand a non-convex optimization problem as one possibly having multiple local minima. For

non-convex problems, most general nonlinear optimization solvers are capable of computing a local solution to the problem. Generally, no guarantee can be made that the computed local solution is or is not a global solution without further analysis. To ensure that a global solution is returned, one needs to employ more advanced techniques that are typically more computationally expensive, (see, for example, [44, 45] on global optimization techniques). Owing to this consideration, local minima will be of interest in this book. Also, in the subsequent chapters, the term *optimal solution* may be used to refer to a local minimum of an optimization problem, and the explicit distinction that the minimum is a local minimum may be omitted.

While minimization problems are treated here, maximization problems, e.g., $\max_{x\in\mathbb{X}} f(x)$ may readily be converted into a minimization problem by minimizing the negative of the objective function, e.g., $\min_{x\in\mathbb{X}} -f(x)$. The optimal solution of each optimization problem are the same, and the optimal objective function value of the maximization problem is equal to the negative of the optimal value of the minimization problem. Thus, there is no loss of generality by considering only minimization problems.

To present general necessary and sufficient optimality conditions for optimality, some regularity conditions or constraint qualifications must be satisfied. First, the active set is defined. For a feasible vector $x \in \mathbb{X}$, the index set of active inequality constraints is defined as $\mathscr{A}(x) := \{j \in \{1, \ldots, n_g\} : g_j(x) = 0\}$. For all $j \notin \mathscr{A}(x) \setminus \{1, \ldots, n_g\}$, the jth inequality constraint is said to be *inactive* at x, i.e., $g_j(x) < 0$. Since the equality constraints are always active, the *active set* includes all the active inequality constraints and all equality constraints. The linear independence constraint qualification (LICQ) holds at $x \in \mathbb{X}$ if the gradients of all active constraints are linearly independent at x, that is, the vectors $\nabla g_j(x)$, $j \in \mathscr{A}(x)$ and $\nabla h_i(x)$, $i = 1, \ldots, n_h$ are linearly independent.

The Lagrangian function of the problem of Eq. 2.52 is given by

$$\mathscr{L}(x, \lambda, \nu) = f(x) + \lambda^T g(x) + \nu^T h(x) \tag{2.53}$$

where $\lambda \in \mathbb{R}^{n_g}$ and $\nu \in \mathbb{R}^{n_h}$ are the Lagrange multipliers. Necessary and sufficient optimality conditions have been derived for the problem of Eq. 2.52. These conditions are not only fundamental to the theory of optimization, but also, allow for the development of computational algorithms that are capable of computing solutions to the optimization problem of Eq. 2.52. The Karush-Kuhn-Tucker (KKT) optimality conditions [46, 47] are first-order necessary conditions for a (local) solution to the problem of Eq. 2.52.

Theorem 2.4 (KKT Conditions, e.g., [36, Proposition 3.3.1]) *Let $x^* \in \mathbb{R}^n$ be a local minimum of the problem of Eq. 2.52 and the LICQ holds at x^*. Then there exist unique $\lambda^* \in \mathbb{R}^{n_g}$ and $\nu^* \in \mathbb{R}^{n_h}$ such that:*

$$\nabla_x \mathscr{L}(x^*, \lambda^*, \nu^*) = 0 \tag{2.54a}$$
$$g(x^*) \leq 0 \tag{2.54b}$$
$$h(x^*) = 0 \tag{2.54c}$$

$$\lambda^* \geq 0 \tag{2.54d}$$

$$\lambda_i^* g_i(x^*) = 0, \; i = 1, \; \ldots, \; n_h \tag{2.54e}$$

If in addition f, h, and g are twice continuously differentiable, then

$$y^T \nabla_{xx} \mathscr{L}(x^*, \lambda^*, \nu^*) y \geq 0, \tag{2.55}$$

for all $y \in \mathbb{R}^n$ such that

$$\begin{aligned} \nabla h_i(x^*)^T y = 0, \forall \, i = 1, \ldots, n_h, \\ \nabla g_j(x^*)^T y = 0, \forall \, j \in \mathscr{A}(x^*). \end{aligned} \tag{2.56}$$

As pointed out in Theorem 2.4, LICQ at a local minimum x^* guarantees existence of Lagrange multipliers. It can be shown that the multipliers are unique as well if the KKT conditions are satisfied and the LICQ holds at x^*. Any triple (x^*, λ^*, ν^*) satisfying the KKT conditions is said to be a *KKT point*. The KKT conditions mean: the gradient of the Lagrangian with respect to the decision variable must vanish at the KKT point, the primal problem, i.e., Eq. 2.52, must be feasible at a KKT point, the dual problem (not discussed here) must be feasible at the KKT point, and *complementarity* or *complementary slackness*, i.e., the condition of Eq. 2.54e, must hold at the KKT point. Since the KKT conditions are necessary conditions, not all KKT points are local minimums. Often second-order necessary conditions are included with the KKT conditions, which is the condition of Eq. 2.55. The second-order necessary conditions mean that it is necessary for a local minimum that the Hessian of the Lagrangian must be positive semidefinite in all feasible directions. If the Hessian of the Lagrangian is shown to be positive definite for a KKT point and *strict complementarity* or *strict complementary slackness* holds, i.e., $\lambda^* > 0$ if $g_i(x^*) = 0$ and $\lambda_i^* = 0$ if $g_i(x^*) < 0$, it can be concluded that the KKT point is a local minimum. This is stated in the following second order sufficient optimality conditions.

Theorem 2.5 (Second Order Optimality Conditions, e.g., [36, Proposition 3.3.2]) *Let the triple (x^*, λ^*, ν^*) be a KKT point that also satisfies:*

$$y^T \nabla_{xx} \mathscr{L}(x^*, \lambda^*, \nu^*) y > 0 \tag{2.57}$$

for all $y \in \mathbb{R}^n$ such that

$$\nabla h_i(x^*)^T y = 0, \; \forall \, i = 1, \ldots, n_h, \tag{2.58}$$

$$\nabla g_j(x^*)^T y = 0, \; \forall \, j \in \mathscr{A}(x^*), \tag{2.59}$$

$$\lambda_j > 0, \quad j \in \mathscr{A}(x^*). \tag{2.60}$$

Then x^* is a strict local minimum of Eq. 2.52.

Numerous variants of the optimality conditions and constraint qualifications given above exist (see, for example, [36]).

2.4.3 Nonlinear Optimization Solution Strategies

The KKT conditions form a set of nonlinear equations and many computation methods for solving for a local minimum of Eq. 2.52 seek to find a solution to the KKT conditions. However, the inequality constraints and the complementary slackness condition, which poses a non-differentiability in the equations, must be handled carefully, and one cannot simply solve the KKT conditions directly in general. Owing to the fact that these conditions form a set of nonlinear equations, the most widely adapted method employed to solve the KKT conditions is Newton's method. More precisely, variants of Newton's method are typically used. From a high level perspective, most nonlinear optimization solvers utilize the user-supplied input information shown in Fig. 2.2 to compute a solution to an optimization problem. The input information includes the functions and their corresponding derivatives. The Hessian of the Lagrangian may also be supplied to the solver. However, in some algorithms such as quasi-Newton methods, the Hessian is approximated in the algorithm. In this section, a basic review of Newton's method is given along with the basic concepts of two widely employed solution techniques for solving nonlinear optimization problems. The two methods include sequential quadratic programming (SQP) and interior point (IP) methods.

2.4.3.1 Newton's Method

The core of most nonlinear optimization solution strategies relies on some variant of Newton's method to solve a set of nonlinear algebraic equations. The standard Newton method is presented to facilitate the discussion of SQP and IP methods for

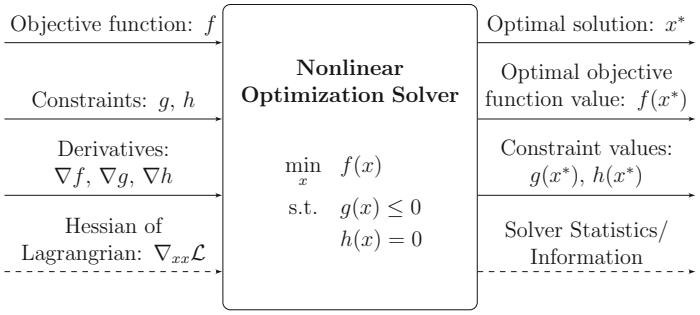

Fig. 2.2 Typical inputs and outputs of a nonlinear optimization solver

solving nonlinear optimization problems. However, it is important to emphasize that key modifications are made to ensure computational efficiency, robustness, etc. To review Newton's method, consider the following nonlinear algebraic equation:

$$F(y) = 0 \tag{2.61}$$

where $F : \mathbb{R}^{n_y} \to \mathbb{R}^{n_y}$ is a vector-valued differentiable function.

Newton's method is an iterative algorithm that is initialized with a starting guess $y_0 \in \mathbb{R}^{n_y}$. At each iteration, the following system of linear equations, which is a linearized version of Eq. 2.61 around the iterate $y_k \in \mathbb{R}^{n_y}$, is solved:

$$\frac{\partial F(y_k)}{\partial y} d_k = -F(y_k) \tag{2.62}$$

where k denotes the iteration number and $d_k \in \mathbb{R}^{n_y}$ is the unknown variable. At the next iteration, i.e., the $(k+1)$th iteration, the iterate is updated as follows:

$$y_{k+1} = y_k + d_k. \tag{2.63}$$

From the update formula of Eq. 2.63, the variable d_k can be interpreted as a descent direction of Newton's method. One of the most advantageous properties of Newton's method is that it has a locally quadratic convergence rate meaning for an initial guess y_0 that is sufficiently close to the solution of Eq. 2.61, Newton's method will converge at a quadratic rate.

To demonstrate applying Newton's method to solving an optimization problem, consider the following unconstrained problem:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2.64}$$

where f is a twice differentiable scalar-valued function. The necessary conditions for optimality at a point $x^* \in \mathbb{R}^n$ are

$$\nabla f(x^*) = 0, \tag{2.65a}$$

$$\nabla^2 f(x^*) \geq 0. \tag{2.65b}$$

Applying Newton's method to solve the nonlinear equations, we have the following update:

$$x_{k+1} = x_k - \nabla_{xx} f(x_k)^{-1} \nabla f(x_k). \tag{2.66}$$

Owing to the fact that the Hessian, i.e., $\nabla_{xx} f$ in the unconstrained case or $\nabla_{xx} \mathscr{L}$ in the constrained case, may be expensive to compute, quasi-Newton methods have been designed to approximate the Hessian; see, for example, [40]. Let $H_k > 0$ be the approximation of the Hessian at iteration k. Then, the update takes form of:

$$x_{k+1} = x_k - H_k^{-1} \nabla f(x_k). \tag{2.67}$$

The fundamental requirement for the convergence of Newton's method algorithm is that the initial guess supplied to the algorithm be sufficiently close to the solution. Globalization strategies are used to allow for convergence to a solution from initial guesses that are not close to the solution. Numerous numerical nonlinear optimization solvers have been developed that are equipped with various globalization strategies. Globalization strategies are briefly discussed here (see, [36, 37, 40], for the details), and they include algorithm monitoring strategies to decide if a computed iterate update is acceptable and modification strategies to modify the iterate updates.

In the first category, merit functions and filter methods are used as a measure of the progress of the algorithm. This adds logic to the algorithm to decide if the step/update is acceptable (loosely speaking, defining the step as d_k in Eq. 2.63). *Merit functions* are scalar-valued functions that are typically chosen to have the same local minimum as the nonlinear optimization problem. At each iteration, the step is accepted if the update yields a decrease in the merit function. Otherwise, the step is rejected. However, merit functions may lead to rejecting pure Newton steps near the optimum and thus, slow down the convergence of the algorithm. As an alternative, *filter methods* treat making the objective function as small as possible and reducing the constraint violations as equal goals. In a filter method, a *filter* keeps track of previous iterates with the best objective function value and amount of the constraint violation. A step is accepted if the update yields a better objective function value or smaller constraint violation. If the update yields a step that is such that one of the previous iterates have a better objective function value and constraint violation, the step is rejected.

In the second category, strategies are used to modify the step or update of the iterates. *Line search* methods take potentially shortened steps if necessary. In other words, line search methods add a dampening factor α in the update formula. The update formula for the iterates is given by:

$$y_{k+1} = y_k + \alpha d_k \tag{2.68}$$

where $\alpha \in (0, 1]$ is selected by the line search method. On the other hand, *trust region* methods recognize that Newton's method utilizes a linearization of the nonlinear function. The resulting linearization is only valid in a neighborhood of the current iterate. Thus, it restricts selection of the step d_k in a small region of the current iterate y_k.

2.4.3.2 Sequential Quadratic Programming

One of the two main solution strategies of nonlinear constrained optimization is to consider successive linearization of the KKT conditions (Eq. 2.54). It turns out that the linearized KKT conditions are the KKT conditions for the following quadratic program (QP) (for a complete derivation one may refer to [37]):

$$\begin{aligned} \min_{d_x} \quad & \nabla f(x_k) d_x + \frac{1}{2} d_x^T H_k d_x \\ \text{s.t.} \quad & g(x_k) + \nabla g(x_k)^T d_x \leq 0 \\ & h(x_k) + \nabla h(x_k)^T d_x = 0 \end{aligned} \tag{2.69}$$

where H_k is either the exact or an approximation of the Hessian of the Lagrangian evaluated at iteration k, i.e., $\nabla_{xx}\mathscr{L}(x_k, \lambda_k, \nu_k)$. If H_k is positive semidefinite, the problem of Eq. 2.69, which is a quadratic program (QP), is convex and efficient methods exist that can readily solve the quadratic program to global optimality. This approach to solving a nonlinear optimization problem is referred to as sequential quadratic programming. Many primal-dual methods used to solve each QP work to find a KKT point of the KKT conditions of Eq. 2.69. The KKT conditions are given by:

$$\begin{aligned} H_k d_x + \nabla f(x_k) + \nabla g(x_k) d_\lambda + \nabla h(x_k) d_\nu &= 0 \\ g(x_k) + \nabla g(x_k)^T d_x &\leq 0 \\ h(x_k) + \nabla h(x_k)^T d_x &= 0 \\ d_\lambda &\geq 0 \\ \left[g(x_k) + \nabla g(x_k)^T d_x \right]_i d_{\lambda,i} &= 0, i = 1, \ldots, n_g \end{aligned} \tag{2.70}$$

With d_x, d_ν, and d_λ, the primal and dual iterates, i.e., x_k, λ_k, and ν_k, are updated. Using this type of the solution strategy, the active set is automatically discovered once the algorithm converges. Under strict complementarity, the solutions of the QP subproblems converge to a local solution to the nonlinear optimization problem once the iterates x_k are in the neighborhood of x^*, e.g. [37].

2.4.3.3 Interior Point Methods

The second widely used class of numerical nonlinear optimization solvers is based on interior point methods. In a number of applications in this book, an open-source interior-point solver, called Ipopt [48], is applied to solve the MPC (EMPC) problems. One interpretation of interior point methods is that the inequality constraints are replaced by the barrier function. The barrier function has the property that the function becomes very large in value as one of the constraint values goes to zero, i.e., if the barrier function is denoted as B, then $B(x) \to \infty$ if $g_i(x) \to 0$ for some $i \in \{1, \ldots, n_g\}$. One widely used barrier function is the logarithmic function:

$$B(x) = -\sum_{i=1}^{n_g} \ln(-g_i(x)). \tag{2.71}$$

The resulting optimization problem is obtained:

$$\begin{aligned} \min_{x} \quad & f(x) - \tau \sum_{i=1}^{n_h} \log(-g_i(x)) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned} \tag{2.72}$$

where $\tau > 0$ is a parameter. In particular, to solve the original nonlinear optimization problem for a local solution, a sequence of modified problems of the form of Eq. 2.72 are solved for a given parameter $\tau > 0$. Under certain conditions, it may be shown that the solution of the original nonlinear optimization problem is the same as the one of the modified problem when the parameter τ approaches zero, e.g., [37, 40].

Another interpretation of interior point methods is that they replace the non-smooth complementary slackness condition of the KKT conditions (Eq. 2.54e) by a smooth approximation. Specifically, the smooth approximation of the KKT conditions is given by:

$$\nabla f(x) + \nabla g(x)\lambda + \nabla h(x)\nu = 0 \tag{2.73a}$$

$$h(x) = 0 \tag{2.73b}$$

$$\lambda_i g_i(x) + \tau = 0, \quad i = 1, \ldots, n_g \tag{2.73c}$$

where $\tau > 0$ is a smoothing parameter. From the last condition, $\lambda_i = -\tau / g_i(x)$, $i = 1, \ldots, n_g$, and thus, the modified KKT conditions of Eq. 2.73 are the KKT conditions of the modified nonlinear optimization problem of Eq. 2.72.

2.4.4 Dynamic Optimization

The specific class of optimization problems that will be of interest in this book is dynamic optimization problems. In particular, the EMPC methods presented in subsequent chapters require the repeated solution of a dynamic optimization problem to compute control actions to apply to a dynamic system. Dynamic optimization or optimal control problems are optimization problems that have a dynamic model embedded in the problem. At this point, the literature on dynamic optimization is vast and impossible to summarize in this brief overview. For general references on theoretical and applied optimal control, the interested reader is referred to one of the many texts on the subject, for example, [41, 49–52]. Important early results that helped shape the foundations of optimal control include optimal control based on the Hamilton-Jacobi-Bellman equation and dynamic programming [53], Pontryagin's maximum principle [54], and the linear quadratic regulator [55].

In this section, direct methods are considered, which are the most commonly employed solution technique for dynamic optimization problem in practical applications. Direct methods first discretize a continuous-time dynamic model and then, use a nonlinear optimization solver to solve the resulting nonlinear optimization problem. Besides direct methods other methods exist including dynamic programming [53] and solving the Hamilton-Jacobi-Bellman partial differential equations,

and indirect methods, which optimize first and then, discretize. The latter methods include setting up the so-called Euler-Lagrange differential equations and applying Pontyragin's Maximum Principle [54], which is a necessary condition, to solve for the optimal control. For a comprehensive review on dynamic optimization with a particular focus on the application of solutions method to moving/receding horizon problems within the context of chemical processes, the reader is referred to the review [42].

Consider the following nonlinear dynamic system:

$$\dot{x} = f(x, u), \quad x(t_0) = x_0 \tag{2.74}$$

where $x \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state vector, $u \in \mathbb{U} \subset \mathbb{R}^m$ is the input vector, and $f : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$. For simplicity, the sets $\mathbb{X}$ and $\mathbb{U}$ are assumed to be compact sets and the vector field f is assumed to satisfy enough smoothness assumptions so that a unique solution to Eq. 2.74 exists over the interval $[t_0, t_f]$ with an initial condition x_0 and piecewise continuous input function $u : [t_0, t_f] \to \mathbb{U}$. The smoothness properties will also be needed to solve the resulting optimization problems below. The solution is denoted as $x(\cdot, x_0, u(\cdot))$, i.e., $x(t, x_0, u(\cdot))$ denotes the solution at $t \in [t_0, t_f]$ and $x(t_0, x_0, u(\cdot)) = x_0$.

Since the purpose of this section is to highlight the various computational approaches to solving dynamic optimization problems, a simple dynamic optimization problem is considered. Nevertheless, more complex problems may be considered while utilizing the presented techniques, e.g., problems with algebraic constraints like path or end-point constraints. Specifically, consider a dynamic optimization problems in Mayer form:

$$\begin{aligned} \min_{x(\cdot), u(\cdot)} \quad & \phi(x(t_f)) \\ \text{s.t.} \quad & \dot{x}(t) = f(x(t), u(t)), \ x(t_0) = x_0 \\ & x(t) \in \mathbb{X}, \ u(t) \in \mathbb{U}, \forall\, t \in [t_0, t_f] \end{aligned} \tag{2.75}$$

where $x_0 \in \mathbb{X}$ denotes the initial condition, which also could be a decision variable in the optimization problem. For simplicity, the initial condition will be assumed to be fixed in the remainder of this chapter. The set constraints are assumed to take the form of inequality constraints, i.e., of the form $h_x(x) \leq 0$ where $\mathbb{X} = \{x \in \mathbb{R}^n : h_x(x) \leq 0\}$ and similarly, for the input constraint with $h_u(u) \leq 0$. The existence of minimizing trajectories for the problem of Eq. 2.75 is assumed; for conditions that guarantee existence of a solution, the interested reader is referred to [56, 57].

It is important to point out that if one seeks the solution to a dynamic optimization problem that optimizes an objective function of the form, i.e., Bolza form:

$$J(x_0, u(\cdot)) = \int_{t_0}^{t_f} l(x(t), u(t))\, dt + V_f(x(t_f)), \tag{2.76}$$

one could readily convert this type problem into Mayer form by defining the state vector as:

$$\bar{x} := \begin{bmatrix} x \\ \bar{\phi} \end{bmatrix} \tag{2.77}$$

with dynamics:

$$\dot{\bar{x}} = \begin{bmatrix} f(x,u) \\ l(x,u) \end{bmatrix}. \tag{2.78}$$

Then, the Mayer term is given by $\phi(\bar{x}(t_f)) := \bar{\phi}(t_f) + V(x(t_f))$.

The major difference between the optimization problem of Eq. 2.52 and the optimization problem of Eq. 2.75 is the presence of the dynamic model embedded in the optimization problem. To avoid an infinite dimensional optimization problem, the control function $u(\cdot)$ must be parameterized by a finite set of parameters, which is referred to as control vector parameterization. The most widely used control vector parameterization is zeroth-order hold, i.e., the input trajectory is assumed to take the form of:

$$u(t) = \bar{u}_i \tag{2.79}$$

for $t \in [\tau_i, \tau_{i+1})$ where $\bar{u}_i \in \mathbb{U}$, $\tau_i := i\Delta + t_0$ for $i = 0, \ldots, N-1$, $\tau_N = t_f$, and $\Delta > 0$ is the hold period. In what follows, the family of (possibly vector-valued) functions that take the form of Eq. 2.79 is generally denoted in this book by $S(\Delta)$ where $\Delta > 0$ is the hold period. For the remainder of this chapter, zeroth order hold control vector parameterization is assumed for simplicity.

Given that the dynamic model embedded in the problem of Eq. 2.75 may be nonlinear, an analytic solution is often difficult to obtain for a given initial condition and input trajectory. Therefore, some numerical method that obtains the solution of the dynamic model is required to solve the optimization problem. The choice of numerical solution techniques used to solve the optimization problem substantially influences the computational efficiency and therefore, is an important implementation consideration. From a nonlinear optimization point of view, dynamic optimization problems typically have a high degree of sparsity and therefore, using sparsity-exploiting nonlinear optimization solvers may also be an important implementation constraint. On the other hand, from a numerical integration standpoint, the most computationally expensive part tends to lie in solving the dynamic model. In particular, computing sensitivity information of the dynamic model tends to be the most computationally expensive step. Therefore, selecting the numerical solver is critical for the success of the solution technique. Below three solution techniques are briefly described.

2.4.4.1 Single Shooting Approach

With a given input trajectory and initial condition, the dynamic model of Eq. 2.74 may be solved forward in time using an ODE solver, i.e., numerical integration method,

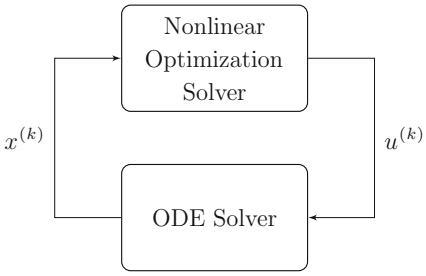

Fig. 2.3 In a single shooting approach, the input trajectory is computed by the nonlinear solver at each iteration. The input trajectory is passed to an ODE solver to compute the corresponding state trajectory, and the corresponding sensitivity information

to obtain the solution over the time interval $[t_0, t_f]$. In this respect, the solution to the dynamic model is a function of the input trajectory and of the initial condition. After the solution to the dynamic model is obtained from the ODE solver, the input trajectory may be updated using a nonlinear optimization solver. These concepts are used in the design of a solution strategy to the optimization problem of Eq. 2.75, which is the single shooting approach.

A block diagram of the methodology is given in Fig. 2.3. At each iteration, the model is first solved over the interval to obtain $x^{(k)}(t, x_0, u^{(k)}(\cdot))$ for $t \in [t_0, t_f]$ where the notation $x^{(k)}$ and $u^{(k)}$ denote the state and input trajectory at the kth iteration of the nonlinear solver. With $x^{(k)}$ and $u^{(k)}$, the objective value, the state and input constraint values, and the sensitivity information (first and second-order derivatives) of the problem of Eq. 2.75 are computed and the nonlinear optimization solver computes the updated input trajectory for the next iteration. The algorithm repeats until the solver converges to a local optimal input trajectory.

For the sake of simplicity, the input trajectory is assumed to be the only decision variable of the dynamic optimization problem and piecewise constant input trajectory is assumed. The resulting formulation of the optimization is given by:

$$\begin{aligned} \min_{\bar{u}_0,\ldots,\bar{u}_{N-1}} \quad & \phi(x(t_f, x_0, u(\cdot))) \\ \text{s.t.} \quad & u(t) = \bar{u}_i, \ \forall\, t \in [\tau_i, \tau_{i+1}), \ i \in \mathbb{I}_{0:N-1} \\ & h_u(\bar{u}_i) = 0, \ \forall\, i \in \mathbb{I}_{0:N-1} \\ & h_x(x(\tau_j, x_0, u(\cdot))) \leq 0, \ \forall\, j \in \mathbb{I}_{0:N_x} \end{aligned} \tag{2.80}$$

where $\tau_j \in [t_0, t_f]$ for $j = 0, \ldots, N_x$ denotes the time grid that the state constraints are imposed with $\tau_0 = t_0$ and $\tau_{N_x} = t_f$. In many cases, taking the time grids used for the control parameterizations and for imposing the state constraints to be equal yields acceptable results.

As a first-pass implementation, one may employ a finite-difference method to approximate sensitivity information of the problem of Eq. 2.80. However, this tends to be inefficient, result in large numerical error, and yield unreliable performance,

e.g., [37]. Instead, one may obtain exact first-order sensitivity information from direct sensitivity, adjoint sensitivity, or automatic (algorithmic) differentiation, e.g., [37, 41]. While methods exist that are capable of computing the exact Hessian of the Lagrangian of the problem of Eq. 2.80, efficient methods for obtaining an approximation of the Hessian have been developed such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. Moreover, the BFGS method tends to yield good computational performance; see, for example, [36, 40].

The key advantages of the single shooting approach to solving a dynamic optimization problem relative to the other two approaches described below are that the method tends to be the easiest to implement and the dynamic model is satisfied (up to numerical precision) at each iteration of the solution method. However, the single shooting method tend to be substantially slower than the multiple shooting and collocation approaches described below. Also, the input trajectory being computed by the nonlinear optimization solver at each iteration of the solution method is an open-loop one, i.e., the input trajectory is first specified by the optimization solver and then, the dynamic equations are solved forward in time with the given open-loop input trajectory. Therefore, solving dynamic optimization problems with a single shooting method when the dynamic model is open-loop unstable may result in numerical problems like unbounded solutions being computed or convergence failure.

2.4.4.2 Multiple Shooting Approach

The multiple shooting method [58] serves as an alternative to a single shooting method. Instead of solving for the solution of the dynamic model over the entire time interval $[t_0, t_f]$, the time interval may be divided into subintervals and the dynamic model may be initialized and solved within each of these subintervals. For simplicity of the presentation, the time horizon $[t_0, t_f]$ is divided into N intervals of constant size $\Delta > 0$, referred to as nodes. While the node intervals are taken to be equally spaced and equal to the hold period of the controls, neither assumption is required for implementation. Let $\tau_i := i\Delta + t_0$ for $i = 0, \ldots, N$. At the beginning of the ith subinterval, the dynamic model is initialized with an initial condition denoted by $s_i \in \mathbb{X}$, which is determined by the optimization solver, and the solution, $x(t, s_i, \bar{u}_i)$ defined for $t \in [\tau_i, \tau_{i+1}]$ where $\bar{u}_i$ denotes the constant input applied over the ith subinterval, is computed by employing a numerical integration method. To ensure that the dynamic model is satisfied over the entire time interval, a constraint is imposed in the optimization problem to ensure that the initial condition specified for the ith subinterval is equal to $x(\tau_i, s_{i-1}, \bar{u}_{i-1})$. In other words, the solutions of each subinterval are pieced together by imposing a constraint in the optimization problem to obtain the solution over the entire interval $[t_0, t_f]$.

The resulting optimization problem for the multiple shooting approach is given by:

$$\min_{\substack{s_0,\ldots,s_N,\\ \bar{u}_0,\ldots,\bar{u}_{N-1}}} \quad \phi(s_N) \tag{2.81a}$$

$$\text{s.t.} \quad s_0 - x_0 = 0 \tag{2.81b}$$

$$x(\tau_{i+1}, s_i, \bar{u}_i) - s_{i+1} = 0, \ \forall\, i \in \mathbb{I}_{0:N-1} \tag{2.81c}$$

$$h_u(\bar{u}_i) = 0, \ \forall\, i \in \mathbb{I}_{0:N-1} \tag{2.81d}$$

$$h_x(s_i) \leq 0, \ \forall\, i \in \mathbb{I}_{1:N} \tag{2.81e}$$

where the constraint of Eq. 2.81b ensures that the initial condition for the first subinterval is equal to x_0 and the constraint of Eq. 2.81c ensures that the solution value at τ_{i+1} is equal to the initial condition specified for the $(i+1)$th subinterval. The constraints of Eqs. 2.81d–2.81e are the input and state constraints, respectively. The state constraint of Eq. 2.81e may readily be extended so that it is imposed over a different time grid like that of the problem of Eq. 2.80.

The multiple shooting method has advantages over a single shooting method in that, loosely speaking, the open-loop instabilities and nonlinearities are distributed amongst the nodes of the time grid. Since solving the dynamic model and the corresponding sensitivity information of the dynamic equations is typically the most computationally expensive task when solving dynamic optimization problems, the multiple shooting method offers a clear way to parallelize the one of the most computationally expensive calculation. Additionally, the problem presents a high-degree of sparsity which may be exploited. Therefore, although the problem of Eq. 2.81 clearly has more decision variables than the problem of Eq. 2.80, the problem of Eq. 2.81 tends to be more computationally efficient than a single shooting approach owing to the aforementioned reasons. The main disadvantage of the method are that the iterates of the method do not necessarily satisfy the dynamic model in the sense each iterate may not satisfy the constraint of Eq. 2.81c.

2.4.4.3 Orthogonal Collocation Approach

The third technique to solve the optimization problem of Eq. 2.75 is to employ orthogonal collocation to obtain the solution to the dynamic model of Eq. 2.74 [59, 60]. Orthogonal collocation approximates the solution of a system of the form of Eq. 2.74 with an interpolating polynomial, e.g., Lagrange polynomials. The coefficients of the polynomial are adjusted such that the interpolating polynomial satisfies the dynamic equation at the collocation points, which are points along the time horizon chosen on the basis of a quadrature rule.

Similar to the multiple shooting approach, the time interval $[t_0, t_f]$ is divided into N subintervals of length $\Delta > 0$. Again, for simplicity of presentation, the subintervals are assumed to be of equal length and constant, i.e., may not be adjusted by the optimization solver. The method extends to the more general case when the

subintervals are not of equal length and may be a decision variable of the optimization problem; please see, for example, [37] and the references therein. Let $\tau_i = t_0 + i\Delta$, $i = 0, 1, \ldots, N$, and the interval $[\tau_i, \tau_{i+1}]$ is the ith subinterval. Within the interval $[\tau_i, \tau_{i+1}]$, n_c collocation points are chosen. Let $p_i(t, c_i)$ be an interpolating polynomial defined for the ith subinterval, i.e., $t \in [\tau_i, \tau_{i+1}]$, that is parameterized by a coefficient vector c_i and τ_i^j denotes the jth collocation point in the ith subinterval where $j = 1, 2, \ldots, n_c$. The solution of Eq. 2.74, obtained through collocation, is computed by solving the following equations:

$$p_i(\tau_i, c_i) = s_i \tag{2.82a}$$

$$\dot{p}_i(\tau_i^j, c_i) = f(p_i(\tau_i^j, c_i), \bar{u}_i), \ \forall j \in \mathbb{I}_{1:n_c}. \tag{2.82b}$$

The optimization problem solved through the collocation approach is given by:

$$\min_{\substack{s_0,\ldots,s_N \\ \bar{u}_0,\ldots,\bar{u}_{N-1} \\ c_0,\ldots,c_{N-1}}} \quad \phi(s_N) \tag{2.83a}$$

$$\text{s.t.} \quad s_0 - x_0 = 0 \tag{2.83b}$$

$$p_i(\tau_i, c_i) - s_i = 0, \ \forall\, i \in \mathbb{I}_{0:(N-1)} \tag{2.83c}$$

$$\dot{p}_i(\tau_{i,j}, c_i) - f(p_i(\tau_{i,j}, c_i), \bar{u}_i) = 0, \ \forall j \in \mathbb{I}_{1:n_c}, \ \forall\, i \in \mathbb{I}_{0:(N-1)} \tag{2.83d}$$

$$p_i(\tau_{i+1}, c_i) - s_{i+1} = 0, \ \forall\, i \in \mathbb{I}_{0:N-1} \tag{2.83e}$$

$$h_u(\bar{u}_i) = 0, \ \forall\, i \in \mathbb{I}_{0:N-1} \tag{2.83f}$$

$$h_x(s_i) \leq 0, \ \forall\, i \in \mathbb{I}_{1:N} \tag{2.83g}$$

where the decision variables of the optimization problem are states values at the nodes, s_i for $i = 0, \ldots, N$, the input trajectory parameterization vectors, $\bar{u}_k$ for $k = 0, 1, \ldots, N-1$, and the coefficients of the interpolating polynomial, c_i for $i = 0, 1, \ldots, (N-1)$.

Owing to the similarities in the structures of the problems of Eqs. 2.81 and 2.83, the orthogonal collocation approach has similar advantages and disadvantages to the ones of the multiple shooting approach. Since the solution to the dynamic model is approximated as a polynomial, analytic computation of the sensitivity information is perhaps easier to obtain than the shooting method approaches because the latter may use a general ODE solver. Within the context of solving the problem of Eq. 2.83, the use of a sparsity exploiting optimization solver is critical because the resulting optimization problem is large-scale with a high degree of sparsity.

References

1. Isidori A (1995) Nonlinear control systems: an introduction, 3rd edn. Springer, New York, NY
2. Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, NJ
3. Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design. Nob Hill Publishing, Madison, WI
4. Massera JL (1956) Contributions to stability theory. Ann Math 64:182–206
5. Krasovskii NN (1963) Stability of motion. Stanford University Press, Stanford
6. Hahn W (1967) Stability of motion. Springer, New York
7. Lin Y, Sontag E, Wang Y (1996) A smooth converse Lyapunov theorem for robust stability. SIAM J Control Optim 34:124–160
8. Teel AR, Praly L (1999) Results on converse Lyapunov functions from class-KL estimates. In: Proceedings of the 38th IEEE conference on decision and control, vol. 3. Phoenix, AZ, pp 2545–2550
9. Hale J (1980) Ordinary differential equations, 2nd edn. Krieger Publishing Company
10. Artstein Z (1983) Stabilization with relaxed controls. Nonlinear Anal: Theory Meth Appl 7:1163–1173
11. Sontag ED (1989) A 'universal' construction of Artstein's theorem on nonlinear stabilization. Syst Control Lett 13:117–123
12. Kravaris C, Kantor JC (1990) Geometric methods for nonlinear process control. 1. Background. Ind Eng Chem Res 29:2295–2310
13. Kravaris C, Kantor JC (1990) Geometric methods for nonlinear process control. 2. Controller synthesis. Ind Eng Chem Res 29:2310–2323
14. Sepulchre R, Janković M, Kokotović PV (1997) Constructive nonlinear control. Communications and control engineering. Springer, London, England
15. Kokotović P, Arcak M (2001) Constructive nonlinear control: a historical perspective. Automatica 37:637–662
16. Christofides PD, El-Farra NH (2005) Control of nonlinear and hybrid process systems: designs for uncertainty, constraints and time-delays. Springer, Berlin, Germany
17. Isidori A (1989) Nonlinear control systems, 2nd edn. Springer, New York
18. Dubljević S, Kazantzis N (2002) A new Lyapunov design approach for nonlinear systems based on Zubov's method. Automatica 38:1999–2007
19. Papachristodoulou A, Prajna S (2002) On the construction of Lyapunov functions using the sum of squares decomposition. In: Proceedings of the 41st IEEE conference on decision and control. Las Vegas, NV, pp 3482–3487
20. Teel AR, Nešić D, Kokotovic PV (1998) A note on input-to-state stability of sampled-data nonlinear systems. In: Proceedings of the 37th IEEE conference on decision and control, vol. 3, pp 2473–2478
21. Muñoz de la Peña D, Christofides PD (2008) Lyapunov-based model predictive control of nonlinear systems subject to data losses. IEEE Trans Autom Control 53:2076–2089
22. Laila DS, Nešić D, Astolfi A (2006) Sampled-data control of nonlinear systems. In: Loría A, Lamnabhi-Lagarrigue F, Panteley E (eds) Advanced Topics in Control Systems Theory, vol 328., Lecture Notes in Control and Information Science. Springer, London, pp 91–137
23. Nešić D, Teel AR (2001) Sampled-data control of nonlinear systems: an overview of recent results. In: Moheimani SOR (ed) Perspectives in robust control, vol 268., Lecture Notes in Control and Information Sciences. Springer, London, pp 221–239
24. Nešić D, Teel AR, Carnevale D (2009) Explicit computation of the sampling period in emulation of controllers for nonlinear sampled-data systems. IEEE Trans Autom Control 54:619–624
25. Ellis M, Karafyllis I, Christofides PD (2014) Stabilization of nonlinear sampled-data systems and economic model predictive control application. In: Proceedings of the American Control conference. Portland, OR, pp 5594–5601
26. Karafyllis I, Grüne L (2011) Feedback stabilization methods for the numerical solution of ordinary differential equations. Discrete Continuous Dyn Syst 16:283–317

27. García CE, Prett DM, Morari M (1989) Model predictive control: theory and practice-A survey. Automatica 25:335–348
28. Morari M, Lee JH (1999) Model predictive control: past, present and future. Comput Chem Eng 23:667–682
29. Mayne DQ, Rawlings JB, Rao CV, Scokaert POM (2000) Constrained model predictive control: stability and optimality. Automatica 36:789–814
30. Rawlings JB (2000) Tutorial overview of model predictive control. IEEE Control Syst Mag 20:38–52
31. Qin SJ, Badgwell TA (2003) A survey of industrial model predictive control technology. Control Eng Pract 11:733–764
32. Camacho EF, Alba CB (2013) Model predictive control, 2nd edn. Springer
33. Mhaskar P, El-Farra NH, Christofides PD (2005) Predictive control of switched nonlinear systems with scheduled mode transitions. IEEE Trans Autom Control 50:1670–1680
34. Mhaskar P, El-Farra NH, Christofides PD (2006) Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. Syst Control Lett 55:650–659
35. Christofides PD, Liu J, Muñoz de la Peña D (2011) Networked and distributed predictive control: methods and nonlinear process network applications. Advances in Industrial Control Series. Springer, London, England
36. Bertsekas DP (1999) Nonlinear programming, 2nd edn. Athena Scientific, Belmont, MA
37. Biegler LT (2010) Nonlinear programming: concepts, algorithms, and applications to chemical processes. SIAM, Philadelphia, PA
38. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge, UK
39. Luenberger DG (2003) Linear and nonlinear programming, 2nd edn. Kluwer Academic Publishers, Boston, MA
40. Nocedal J, Wright S (2006) Numerical optimization, 2nd edn. Springer, New York, NY
41. Betts JT (2010) Practical methods for optimal control and estimation using nonlinear programming, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA
42. Binder T, Blank L, Bock HG, Bulirsch R, Dahmen W, Diehl M, Kronseder T, Marquardt W, Schlöder JP, von Stryk O (2001) Introduction to model based optimization of chemical processes on moving horizons. In: Grötschel M, Krumke SO, Rambau J (eds) Online optimization of large scale systems. Springer, Berlin Heidelberg, pp 295–339
43. Diehl M, Ferreau HJ, Haverbeke N (2009) Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: Magni L, Raimondo DM, Allgöwer F (eds) Nonlinear model predictive control, vol 384., Lecture Notes in Control and Information Sciences. Springer, Berlin Heidelberg, pp 391–417
44. Horst R, Tuy H (1996) Global optimization: deterministic approaches. Springer
45. Floudas CA (2000) Deterministic global optimization: theory, methods and applications. Nonconvex Optimization and its Application, vol 37. Kluwer Academic Publishers (2000)
46. Karush W (1939) Minima of functions of several variables with inequalities as side conditions. Master's thesis, University of Chicago
47. Kuhn HW, Tucker AW (1951) Nonlinear programming. In: Proceedings of the Second Berkeley symposium on mathematical statistics and probability. University of California Press, Berkeley, CA , pp 481–492
48. Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 106:25–57
49. Bertsekas DP (2005) Dynamic programming and optimal control, vol 1, 3rd edn. Athena Scientific, Belmont, MA
50. Bryson AE, Ho YC (1975) Applied optimal control: optimization. Estimation and control. Taylor & Francis Group, New York
51. Kirk DE (2012) Optimal control theory: an introduction. Dover Publications, Mineola, NY
52. Lewis FL, Vrabie DL, Syrmos VL (2012) Optimal control, 3rd edn. Wiley, Hoboken, New Jersey
53. Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton, N.J

54. Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishchenko EF (1961) Mathematical theory of optimal processes. Fizmatgiz, Moscow
55. Kalman RE (1960) Contributions to the theory of optimal control. Boletin de la Sociedad Matematica Mexicana 5:102–119
56. Lee EB, Markus L (1967) Foundations of optimal control theory. SIAM series in applied mathematics. Wiley, New York, NY
57. Vinter R (2010) Optimal control. Modern Birkhäuser Classics. Birkhäuser, Boston, MA
58. Bock HG, Plitt KJ (1984) A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings of the 9th IFAC world congress. Budapest, Hungary, pp 242–247
59. Biegler LT (2007) An overview of simultaneous strategies for dynamic optimization. Chem Eng Process 46:1043–1053
60. Biegler LT (1984) Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. Comput Chem Eng 8:243–247

http://www.springer.com/978-3-319-41107-1

Economic Model Predictive Control
Theory, Formulations and Chemical Process Applications

Ellis, M.; Liu, J.; Christofides, P.D.

2017, XXIV, 292 p. 95 illus., 16 illus. in color., Hardcover
ISBN: 978-3-319-41107-1