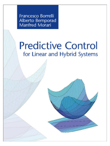

Predictive Control: for Linear and Hybrid Systems
ISBN 9781139061759
III Optimal Control
7 General Formulation and Discussion

7
General Formulation and Discussion

In this chapter we introduce the optimal control problem we will be studying in a very general form. We want to communicate the basic definitions and essential concepts. We will sacrifice mathematical precision for the sake of simplicity. In later chapters we will study specific versions of this problem for specific cost functions and system classes in greater detail.

7.1 Problem Formulation

We consider the nonlinear time-invariant system

$$x(t+1) = g(x(t), u(t)), \tag{7.1}$$

subject to the constraints

$$h(x(t), u(t)) \leq 0 \tag{7.2}$$

at all time instants $t \geq 0$. In (7.1)–(7.2), $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ are the state and input vector, respectively. Inequality (7.2) with $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n_c}$ expresses the n_c constraints imposed on the input and the states. These may be simple upper and lower bounds or more complicated expressions. We assume that the origin is an equilibrium point ($g(0, 0) = 0$) in the interior of the feasible set, i.e., $h(0, 0) < 0$.

We assumed the system to be specified in discrete time. One reason is that we are looking for solutions to engineering problems. In practice, the controller will almost always be implemented through a digital computer by sampling the variables of the system and transmitting the control action to the system at discrete time points. Another reason is that for the solution of the optimal control problems for discrete-time systems we will be able to make ready use of powerful mathematical programming software.

We want to caution the reader, however, that in many instances the discrete time model is an approximation of the continuous time model. It is generally difficult to derive "good" discrete time models from nonlinear continuous time models, and especially so when the nonlinear system has discontinuities as would be the case for switched systems. We also note that continuous time switched systems can exhibit behavioral characteristics not found in discrete-time systems, for example, an ever increasing number of switches in an ever decreasing time interval (*Zeno behavior* [127]).

We define the following *performance objective* or *cost function* from time instant 0 to time instant N

$$J_{0\to N}(x_0, U_{0\to N}) = p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k), \tag{7.3}$$

where N is the time *horizon* and x_k denotes the state vector at time k obtained by starting from the measured state $x_0 = x(0)$ and applying to the system model

$$x_{k+1} = g(x_k, u_k), \tag{7.4}$$

the input sequence $u_0, \ldots, u_{k-1}$. From this sequence we define the vector of future inputs $U_{0\to N} = [u_0', \ldots, u_{N-1}']' \in \mathbb{R}^s$, $s = mN$. The terms $q(x_k, u_k)$ and $p(x_N)$ are referred to as *stage cost* and *terminal cost*, respectively, and are assumed to be positive definite ($q \succ 0$, $p \succ 0$):

$$\begin{array}{l} p(x,u) > 0 \ \forall x \neq 0,\ u \neq 0, \quad p(0,0) = 0 \\ q(x,u) > 0 \ \forall x \neq 0,\ u \neq 0, \quad q(0,0) = 0. \end{array}$$

The form of the cost function (7.3) is very general. If a practical control objective can be expressed as a scalar function then this function usually takes the indicated form. Specifically, we consider the following constrained finite time optimal control (CFTOC) problem.

$$\begin{array}{rl} J_{0\to N}^*(x_0) = \min_{U_{0\to N}} & J_{0\to N}(x_0, U_{0\to N}) \\ \text{subj. to} & x_{k+1} = g(x_k, u_k),\ k = 0, \ldots, N-1 \\ & h(x_k, u_k) \leq 0,\ k = 0, \ldots, N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(0). \end{array} \tag{7.5}$$

Here $\mathcal{X}_f \subseteq \mathbb{R}^n$ is a *terminal region* that we want the system states to reach at the end of the horizon. The terminal region could be the origin, for example. We define $\mathcal{X}_{0\to N} \subseteq \mathbb{R}^n$ to be the set of initial conditions $x(0)$ for which there exists an input vector $U_{0\to N}$ so that the inputs $u_0, \ldots, u_{N-1}$ and the states $x_0, \ldots, x_N$ satisfy the model $x_{k+1} = g(x_k, u_k)$ and the constraints $h(x_k, u_k) \leq 0$ and that the state x_N lies in the terminal set $\mathcal{X}_f$.

We can determine this set of feasible initial conditions in a recursive manner. Let us denote with $\mathcal{X}_{j\to N}$ the set of states x_j at time j which can be steered into $\mathcal{X}_f$ at time N, i.e., for which the model $x_{k+1} = g(x_k, u_k)$ and the constraints $h(x_k, u_k) \leq 0$ are feasible for $k = j, \ldots, N-1$ and $x_N \in \mathcal{X}_f$. This set can be defined recursively by

$$\begin{array}{rl} \mathcal{X}_{j\to N} = & \{x \in \mathbb{R}^n :\ \exists u \text{ such that } (h(x,u) \leq 0, \text{ and } g(x,u) \in \mathcal{X}_{j+1\to N})\}, \\ & j = 0, \ldots, N-1 \end{array} \tag{7.6}$$

$$\mathcal{X}_{N\to N} = \mathcal{X}_f. \tag{7.7}$$

The set $\mathcal{X}_{0\to N}$ is the final result of these iterations starting with $\mathcal{X}_f$.

The optimal cost $J^*_{0\to N}(x_0)$ is also called *value function*. In general, the problem (7.3)–(7.5) may not have a minimum. We will assume that there exists a minimum. This is the case, for example, when the set of feasible input vectors $U_{0\to N}$ (defined by h and $\mathcal{X}_f$) is compact and when the functions g, p and q are continuous. Also, there might be several input vectors $U^*_{0\to N}$ which yield the minimum $(J^*_{0\to N}(x_0) = J_{0\to N}(x_0, U^*_{0\to N}))$. In this case we will define one of them as the minimizer $U^*_{0\to N}$.

Note that throughout the book we will distinguish between the *current* state $x(k)$ of system (7.1) at time k and the variable x_k in the optimization problem (7.5), that is the *predicted* state of system (7.1) at time k obtained by starting from the state x_0 and applying to system (7.4) the input sequence $u_0, \dots, u_{k-1}$. Analogously, $u(k)$ is the input applied to system (7.1) at time k while u_k is the k-th optimization variable of the optimization problem (7.5). Clearly, $x(k) = x_k$ for any k if $u(k) = u_k$ for all k (under the assumption that our model is perfect).

In the rest of this chapter we will be interested in the following questions related to the general optimal control problem (7.3)–(7.5).

Solution. We will show that the problem can be expressed and solved either as one general nonlinear programming problem, or in a recursive manner by invoking Bellman's Principle of Optimality.

Infinite horizon. We will investigate if a solution exists as $N \to \infty$, the properties of this solution and how it can be obtained or at least approximated by using a *receding horizon* technique.

7.2 Solution via Batch Approach

If we write the equality constraints appearing in (7.5) explicitly

$$\begin{aligned} x_1 &= g(x(0), u_0) \\ x_2 &= g(x_1, u_1) \\ &\vdots \\ x_N &= g(x_{N-1}, u_{N-1}), \end{aligned} \tag{7.8}$$

then the optimal control problem (7.3)–(7.5), rewritten below

$$
\begin{array}{rl}
J^*_{0\to N}(x_0) = \min_{U_{0\to N}} & p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k) \\
\text{subj. to} & x_1 = g(x_0, u_0) \\
& x_2 = g(x_1, u_1) \\
& \vdots \\
& x_N = g(x_{N-1}, u_{N-1}) \\
& h(x_k, u_k) \le 0,\ k = 0, \ldots, N-1 \\
& x_N \in \mathcal{X}_f \\
& x_0 = x(0)
\end{array}
\tag{7.9}
$$

is recognized more easily as a general nonlinear programming problem with variables $u_0, \ldots, u_{N-1}$ and $x_1, \ldots, x_N$.

As an alternative we may try to eliminate the state variables and equality constraints (7.8) by successive substitution so that we are left with $u_0, \ldots, u_{N-1}$ as the only decision variables. For example, we can express x_2 as a function of $x(0)$, u_0 and u_1 only, by eliminating the intermediate state x_1

$$
\begin{array}{l}
x_2 = g(x_1, u_1) \\
x_2 = g(g(x(0), u_0), u_1).
\end{array}
\tag{7.10}
$$

Except when the state equations are linear this successive substitution may become complex. Even when they are linear it may be bad from a numerical point of view.

Either with or without successive substitution the solution of the nonlinear programming problem is a sequence of present and future inputs $U^*_{0\to N} = [u_0^{*'}, \ldots, u_{N-1}^{*'}]'$ determined for the particular initial state $x(0)$.

7.3 Solution via Recursive Approach

The recursive approach, Bellman's dynamic programming technique, rests on a simple idea, the *principle of optimality*. It states that for a trajectory $x_0, x_1^*, \ldots, x_N^*$ to be optimal, the trajectory starting from any intermediate point x_j^*, i.e., $x_j^*, x_{j+1}^*, \ldots, x_N^*$, $0 \le j \le N-1$, must be optimal.

Consider the following example to provide an intuitive justification [53]. Suppose that the fastest route from Los Angeles to Boston passes through Chicago. Then the principle of optimality formalizes the obvious fact that the Chicago to Boston portion of the route is also the fastest route for a trip that starts from Chicago and ends in Boston.

We can utilize the principle of optimality for the optimal control problem we are investigating. We define the cost over the reduced horizon from j to N

$$
J_{j\to N}(x_j, u_j, u_{j+1}, \ldots, u_{N-1}) = p(x_N) + \sum_{k=j}^{N-1} q(x_k, u_k),
\tag{7.11}
$$

also called the *cost-to-go*. Then the *optimal cost-to-go* $J^*_{j\to N}$ is

$$\begin{array}{lll} J^*_{j\to N}(x_j) = \min_{u_j,u_{j+1},\dots,u_{N-1}} & J_{j\to N}(x_j,u_j,u_{j+1},\dots,u_{N-1}) \\ \quad\quad \text{subj. to} & x_{k+1} = g(x_k,u_k),\ k=j,\dots,N-1 \\ & h(x_k,u_k)\le 0,\ k=j,\dots,N-1 \\ & x_N \in \mathcal{X}_f. \end{array} \tag{7.12}$$

Note that the optimal cost-to-go $J^*_{j\to N}(x_j)$depends only on the initial state x_j.

The principle of optimality implies that the optimal cost-to-go $J^*_{j-1\to N}$ from time $j-1$ to the final time N can be found by minimizing the sum of the stage cost $q(x_{j-1}, u_{j-1})$ and the optimal cost-to-go $J^*_{j\to N}(x_j)$ from time j onwards:

$$\begin{array}{lll} J^*_{j-1\to N}(x_{j-1}) = & \min\limits_{u_{j-1}} & q(x_{j-1},u_{j-1}) + J^*_{j\to N}(x_j) \\ & \text{subj. to} & x_j = g(x_{j-1},u_{j-1}) \\ & & h(x_{j-1},u_{j-1}) \le 0 \\ & & x_j \in \mathcal{X}_{j\to N}. \end{array} \tag{7.13}$$

Here the only decision variable left for the optimization is u_{j-1}, the input at time $j-1$. All the other inputs $u^*_j,\dots,u^*_{N-1}$ have already been selected optimally to yield the optimal cost-to-go $J^*_{j\to N}(x_j)$. We can rewrite (7.13) as

$$\begin{array}{lll} J^*_{j-1\to N}(x_{j-1}) = & \min\limits_{u_{j-1}} & q(x_{j-1},u_{j-1}) + J^*_{j\to N}(g(x_{j-1},u_{j-1})) \\ & \text{subj. to} & h(x_{j-1},u_{j-1}) \le 0 \\ & & g(x_{j-1},u_{j-1}) \in \mathcal{X}_{j\to N}, \end{array} \tag{7.14}$$

making the dependence of x_j on the initial state x_{j-1} explicit.

The optimization problem (7.14) suggests the following recursive algorithm backwards in time to determine the optimal control law. We start with the terminal cost and constraint

$$J^*_{N\to N}(x_N) = p(x_N) \tag{7.15}$$

$$\mathcal{X}_{N\to N} = \mathcal{X}_f, \tag{7.16}$$

and then proceed backwards

$$\begin{array}{lll} J^*_{N-1\to N}(x_{N-1}) = & \min\limits_{u_{N-1}} & q(x_{N-1},u_{N-1}) + J^*_{N\to N}(g(x_{N-1},u_{N-1})) \\ & \text{subj. to} & h(x_{N-1},u_{N-1}) \le 0, \\ & & g(x_{N-1},u_{N-1}) \in \mathcal{X}_{N\to N} \\ \vdots & & \\ J^*_{0\to N}(x_0) = & \min\limits_{u_0} & q(x_0,u_0) + J^*_{1\to N}(g(x_0,u_0)) \\ & \text{subj. to} & h(x_0,u_0) \le 0, \\ & & g(x_0,u_0) \in \mathcal{X}_{1\to N} \\ & & x_0 = x(0). \end{array} \tag{7.17}$$

This algorithm, popularized by Bellman, is referred to as *dynamic programming*. The dynamic programming problem is appealing because it can be stated compactly and because at each step the optimization takes place over one element u_j of the optimization vector only. This optimization is rather complex, however. It is not a standard nonlinear programming problem, since we have to construct the optimal cost-to-go $J^*_{j\to N}(x_j)$, a *function* defined over the subset $\mathcal{X}_{j\to N}$ of the state space.

In a few special cases we know the type of function and we can find it efficiently. For example, in the next chapter we will cover the case when the system is linear and the cost is quadratic. Then the optimal cost-to-go is also quadratic and can be constructed rather easily. Later in the book we will show that, when constraints are added to this problem, the optimal cost-to-go becomes piecewise quadratic and efficient algorithms for its construction are also available.

In general, however, we may have to resort to a "brute force" approach to construct the cost-to-go function $J^*_{j-1\to N}$ and to solve the dynamic program. Let us assume that at time j–1 the cost-to-go $J^*_{j\to N}$ is known and discuss how to construct an approximation of $J^*_{j-1\to N}$. With $J^*_{j\to N}$ known, for a fixed x_{j-1} the optimization problem (7.14) becomes a standard nonlinear programming problem. Thus, we can define a grid in the set $\mathcal{X}_{j-1\to N}$ of the state space and compute the optimal cost-to-go function on each grid point. We can then define an approximate value function $\tilde{J}^*_{j-1\to N}(x_{j-1})$ at intermediate points via interpolation. The complexity of constructing the cost-to-go function in this manner increases rapidly with the dimension of the state space ("curse of dimensionality").

The extra benefit of solving the optimal control problem via dynamic programming is that we do not only obtain the vector of optimal inputs $U^*_{0\to N}$ for a particular initial state $x(0)$ as with the batch approach. At each time j the optimal cost-to-go function defines implicitly a nonlinear feedback control law.

$$\begin{array}{lll} u^*_j(x_j) = \arg \min\limits_{u_j} & q(x_j,u_j) + J^*_{j+1\to N}(g(x_j,u_j)) \\ \quad\quad\quad\text{subj. to} & h(x_j,u_j) \le 0, \\ & g(x_j,u_j) \in \mathcal{X}_{j+1\to N}. \end{array} \tag{7.18}$$

For a fixed x_j this nonlinear programming problem can be solved quite easily in order to find $u^*_j(x_j)$. Because the optimal cost-to-go function $J^*_{j\to N}(x_j)$ changes with time j, the nonlinear feedback control law is time-varying.

7.4 Optimal Control Problem with Infinite Horizon

We are interested in the optimal control problem (7.3)–(7.5) as the horizon N approaches infinity.

$$\begin{array}{lll} J^*_{0\to\infty}(x_0) = \min_{u_0,u_1,\ldots} & \sum\limits_{k=0}^{\infty} q(x_k,u_k) \\ \quad\quad\quad\text{subj. to} & x_{k+1} = g(x_k,u_k),\ k=0,\ldots,\infty \\ & h(x_k,u_k) \le 0,\ k=0,\ldots,\infty \\ & x_0 = x(0). \end{array} \tag{7.19}$$

We define the set of initial conditions for which this problem has a solution.

$$\mathcal{X}_{0\to\infty} = \{x(0) \in \mathbb{R}^n : \quad \text{Problem (7.19) is feasible and } J^*_{0\to\infty}(x(0)) < +\infty\}. \tag{7.20}$$

For the value function $J^*_{0\to\infty}(x_0)$ to be finite it must hold that

$$\lim_{k\to\infty} q(x_k, u_k) = 0,$$

and because $q(x_k, u_k) > 0$ for all $(x_k, u_k) \neq 0$

$$\lim_{k\to\infty} x_k = 0$$

and

$$\lim_{k\to\infty} u_k = 0.$$

Thus the sequence of control actions generated by the solution of the infinite horizon problem drives the system to the origin. For this solution to exists the system must be, loosely speaking, stabilizable.

Using the recursive dynamic programming approach we can seek the solution of the infinite horizon optimal control problem by increasing N until we observe convergence. If the dynamic programming algorithm converges as $N \to \infty$ then (7.14) becomes the Bellman equation

$$\begin{array}{rll} J^*(x) = \min_u & q(x,u) + J^*(g(x,u)) \\ \text{subj. to} & h(x,u) \leq 0 \\ & g(x,u) \in \mathcal{X}_{0\to\infty} \end{array} \tag{7.21}$$

This procedure of simply increasing N may not be well behaved numerically and it may also be difficult to define a convergence criterion that is meaningful for the control problem. We will describe a method, called *Value Function Iteration*, in the next section.

An alternative is *receding horizon control* which can yield a time invariant controller guaranteeing convergence to the origin without requiring $N \to \infty$. We will describe this important idea later in this chapter.

7.4.1 Value Function Iteration

Once the value function $J^*(x)$ is known, the nonlinear feedback control law $u^*(x)$ is defined implicitly by (7.21)

$$\begin{array}{rll} u^*(x) = \arg\min_u & q(x,u) + J^*(g(x,u)) \\ \text{subj. to} & h(x,u) \leq 0 \\ & g(x,u) \in \mathcal{X}_{0\to\infty}. \end{array} \tag{7.22}$$

It is *time invariant* and guarantees convergence to the origin for all states in $\mathcal{X}_{0\to\infty}$. For a given $x \in \mathcal{X}_{0\to\infty}$, $u^*(x)$ can be found from (7.21) by solving a standard nonlinear programming problem.

In order to find the value function $J^*(x)$ we need to solve (7.21). We can start with some initial guess $\tilde{J}^*_0(x)$ for the value function and an initial guess $\tilde{\mathcal{X}}_0$ for the region in the state space where we expect the infinite horizon problem to converge and iterate. Then at iteration $i + 1$ solve

$$\begin{array}{lll} \tilde{J}^*_{i+1}(x) = \min_u & q(x,u) + \tilde{J}^*_i(g(x,u)) \\ \quad\quad\quad\quad\quad \text{subj. to} & h(x,u) \le 0 \\ & g(x,u) \in \tilde{\mathcal{X}}_i \end{array} \tag{7.23}$$

$$\tilde{\mathcal{X}}_{i+1} = \{x \in \mathbb{R}^n : \ \exists u \ (h(x,u) \le 0, \ \text{and} \ g(x,u) \in \tilde{\mathcal{X}}_i)\}. \tag{7.24}$$

Again, here i is the iteration index and does not denote time. This iterative procedure is called *value function iteration*. It can be executed as follows. Let us assume that at iteration step i we gridded the set $\tilde{\mathcal{X}}_i$ and that $\tilde{J}^*_i(x)$ is known at each grid point from the previous iteration. We can approximate $J^*_i(x)$ at intermediate points via interpolation. For a fixed point $\bar{x}$ the optimization problem (7.23) is a nonlinear programming problem yielding $\tilde{J}^*_i(\bar{x})$. In this manner the approximate value function $\tilde{J}^*_i(x)$ can be constructed at all grid points and we can proceed to the next iteration step $i + 1$.

7.4.2 Receding Horizon Control

Receding Horizon Control will be covered in detail in Chapter 12. Here we illustrate the main idea and discuss the fundamental properties.

Assume that at time $t = 0$ we determine the control action u_0 by solving the finite horizon optimal control problem (7.3)–(7.5). If $J^*_{0\to N}(x_0)$ converges to $J^*_{0\to\infty}(x_0)$ as $N \to \infty$ then the effect of increasing N on the value of u_0 should diminish as $N \to \infty$. Thus, intuitively, instead of making the horizon infinite we can get a similar behavior when we use a long, but finite horizon N, and repeat this optimization at each time step, in effect moving the horizon forward (*moving horizon* or *receding horizon* control). We can use the batch or the dynamic programming approach.

Batch approach. We solve an optimal control problem with horizon N yielding a sequence of optimal inputs $u^*_0, \ldots, u^*_{N-1}$, but we would implement only the first one of these inputs u^*_0. At the next time step we would measure the current state and then again solve the N-step problem with the current state as new initial condition x_0. If the horizon N is long enough then we expect that this approximation of the infinite horizon problem should not matter and the implemented sequence should drive the states to the origin.

Dynamic programming approach. We always implement the control u_0 obtained from the optimization problem

$$\begin{array}{lll} J^*_{0\to N}(x_0) = & \min\limits_{u_0} & q(x_0,u_0) + \ J^*_{1\to N}(g(x_0,u_0)) \\ & \text{subj. to} & h(x_0,u_0) \le 0, \\ & & g(x_0,u_0) \in \mathcal{X}_{1\to N}, \\ & & x_0 = x(0) \end{array} \tag{7.25}$$

where $J^*_{1\to N}(g(x_0,u_0))$ is the optimal cost-to-go from the state $x_1 = g(x_0, u_0)$ at time 1 to the end of the horizon N.

If the dynamic programming iterations converge as $N \to \infty$, then for a long, but finite horizon N we expect that this receding horizon approximation of the infinite horizon problem should not matter and the resulting controller will drive the system asymptotically to the origin.

In both the batch and the recursive approach, however, it is not obvious how long N must be for the receding horizon controller to inherit these desirable convergence characteristics. Indeed, for computational simplicity we would like to keep N small. We will argue next that the proposed control scheme guarantees convergence just like the infinite horizon variety if we impose a specific terminal constraint, for example, if we require the terminal region to be the origin $\mathcal{X}_f = 0$.

From the principle of optimality we know that

$$J^*_{0\to N}(x_0) = \min_{u_0} \; q(x_0, u_0) + J^*_{1\to N}(x_1). \tag{7.26}$$

Assume that we are at $x(0)$ at time 0 and implement the optimal u_0^* that takes us to the next state $x_1 = g(x(0), u_0^*)$. At this state at time 1 we postulate to use over the next N steps the sequence of optimal moves determined at the previous step followed by zero: $u_1^*, \ldots, u_{N-1}^*, 0$. This sequence is not optimal but the associated cost over the shifted horizon from 1 to $N + 1$ can be easily determined. It consists of three parts: (1) the optimal cost $J^*_{0\to N}(x_0)$ from time 0 to N computed at time 0, minus (2) the stage cost $q(x_0, u_0)$ at time 0 plus (3) the cost at time $N + 1$. But this last cost is zero because we imposed the terminal constraint $x_N = 0$ and assumed $u_N = 0$. Thus the cost over the shifted horizon for the assumed sequence of control moves is

$$J^*_{0\to N}(x_0) - q(x_0, u_0).$$

Because this postulated sequence of inputs is not optimal at time 1

$$J^*_{1\to N+1}(x_1) \le J^*_{0\to N}(x_0) - q(x_0, u_0).$$

Because the system and the objective are time invariant $J^*_{1\to N+1}(x_1) = J^*_{0\to N}(x_1)$ so that

$$J^*_{0\to N}(x_1) \le J^*_{0\to N}(x_0) - q(x_0, u_0).$$

As $q \succ 0$ for all $(x, u) \neq (0, 0)$, the sequence of optimal costs $J^*_{0\to N}(x_0), J^*_{0\to N}(x_1), \ldots$ is strictly decreasing for all $(x, u) \neq (0, 0)$. Because the cost $J^*_{0\to N} \ge 0$ the sequence $J^*_{0\to N}(x_0), J^*_{0\to N}(x_1), \ldots$ (and thus the sequence $x_0, x_1, \ldots$) is converging. Thus we have established the following important theorem.

Theorem 7.1 (Convergence of Receding Horizon Control) *At time step j consider the cost function*

$$J_{j\to j+N}(x_j, u_j, u_{j+1}, \ldots, u_{j+N-1}) = \sum_{k=j}^{j+N} q(x_k, u_k), \;\; q \succ 0 \tag{7.27}$$

and the CFTOC problem

$$\begin{array}{lll} J^*_{j\to j+N}(x_j) = \min_{u_j, u_{j+1}, \ldots, u_{j+N-1}} & J_{j\to j+N}(x_j, u_j, u_{j+1}, \ldots, u_{j+N-1}) \\ \quad\quad\quad\quad\quad\quad \text{subj. to} & x_{k+1} = g(x_k, u_k) \\ & h(x_k, u_k) \le 0, \;\; k = j, \ldots, j+N-1 \\ & x_N = 0 \end{array} \tag{7.28}$$

Assume that only the optimal u_j^* is implemented. At the next time step $j + 1$ the CFTOC problem is solved again starting from the resulting state $x_{j+1} = g(x_j, u_j^*)$. Assume that the CFTOC problem (7.28) has a solution for every state $x_j, x_{j+1}, \ldots$ resulting from the control policy. Then the system will converge to the origin as $j \to \infty$. ■

Thus we have established that a receding horizon controller with terminal constraint $x_N = 0$ has the same desirable convergence characteristics as the infinite horizon controller. At first sight the theorem appears very general and powerful. It is based on the implicit assumption, however, that at every time step the CFTOC problem has a solution. Infeasibility would occur, for example, if the underlying system is not stabilizable. It could also happen that the constraints on the inputs which restrict the control action prevent the system from reaching the terminal state in N steps. In Chapter 12 we will present special formulations of problem (7.28) such that feasibility at the initial time guarantees feasibility for all future times. Furthermore, in addition to asymptotic convergence to the origin we will establish stability for the closed-loop system with the receding horizon controller.

Remark 7.1 For the sake of simplicity in the rest of the book we will use the following shorter notation

$$\begin{aligned}
&J_j^*(x_j) = J_{j\to N}^*(x_j),\ \ j = 0,\ldots,N\\
&J_\infty^*(x_0) = J_{0\to\infty}^*(x_0)\\
&\mathcal{X}_j = \mathcal{X}_{j\to N},\ \ j = 0,\ldots,N\\
&\mathcal{X}_\infty = \mathcal{X}_{0\to\infty}\\
&U_0 = U_{0\to N}
\end{aligned} \tag{7.29}$$

and use the original notation only if needed.

7.5 Lyapunov Stability

While asymptotic convergence $\lim_{k\to\infty} x_k = 0$ is a desirable property, it is generally not sufficient in practice. We would also like a system to stay in a small neighborhood of the origin when it is disturbed slightly. Formally, this is expressed as Lyapunov stability.

7.5.1 General Stability Conditions

Consider the autonomous system

$$x_{k+1} = g(x_k) \tag{7.30}$$

with $g(0) = 0$.

Definition 7.1 (Lyapunov Stability) *The equilibrium point $x = 0$ of system (7.30) is*

- stable *(in the sense of Lyapunov) if, for each $\varepsilon > 0$, there is $\delta > 0$ such that*

$$||x_0|| < \delta \Rightarrow ||x_k|| < \varepsilon,\ \forall k \geq 0 \tag{7.31}$$

- unstable *if not stable*
- asymptotically stable *in $\Omega \subseteq \mathbb{R}^n$ if it is stable and*

$$\lim_{k\to\infty} x_k = 0,\ \forall x_0 \in \Omega \tag{7.32}$$

- globally asymptotically stable *if it is asymptotically stable and* $\Omega = \mathbb{R}^n$
- exponentially stable *if it is stable and there exist constants* $\alpha > 0$ *and* $\gamma \in (0, 1)$ *such that*

$$\|x_0\| < \delta \Rightarrow \|x_k\| \le \alpha\|x_0\|\gamma^k,\ \forall k \ge 0. \tag{7.33}$$

The ε-δ requirement for stability (7.31) takes a challenge–answer form. To demonstrate that the origin is stable, for any value of ε that a challenger may chose (however small), we must produce a value of δ such that a trajectory starting in a δ neighborhood of the origin will never leave the ε neighborhood of the origin.

Remark 7.2 If in place of system (7.30), we consider the time-varying system $x_{k+1} = g(x_k, k)$, then δ in Definition 7.1 is a function of ε and k, i.e., $\delta = \delta(\varepsilon, k) > 0$. In this case, we introduce the concept of "uniform stability." The equilibrium point $x = 0$ is *uniformly stable* if, for each $\varepsilon > 0$, there is $\delta = \delta(\varepsilon) > 0$ (independent from k) such that

$$\|x_0\| < \delta \Rightarrow \|x_k\| < \varepsilon,\ \forall k \ge 0. \tag{7.34}$$

The following example shows that Lyapunov stability and convergence are, in general, different properties.

Example 7.1 Consider the following system with one state $x \in \mathbb{R}$:

$$x_{k+1} = x_k(x_k - 1 - |x_k - 1|). \tag{7.35}$$

The state $x = 0$ is an equilibrium for the system. For any state $x \in [-1, 1]$ we have $(x - 1) \le 0$ and the system dynamics (7.35) become

$$x_{k+1} = 2x_k^2 - 2x_k. \tag{7.36}$$

System (7.36) generates oscillating and diverging trajectories for any $x_0 \in (-1, 1)\{0\}$. Any such trajectory will enter in finite time T the region with $x \ge 1$. In this region the system dynamics (7.35) become

$$x_{k+1} = 0,\ \forall\ k \ge T. \tag{7.37}$$

Therefore the origin is not Lyapunov stable, however the system converges to the origin for all $x_0 \in (-\infty, +\infty)$.

Usually, to show Lyapunov stability of the origin for a particular system one constructs a so called *Lyapunov function*, i.e., a function satisfying the conditions of the following theorem.

Theorem 7.2 *Consider the equilibrium point* $x = 0$ *of system (7.30). Let* $\Omega \subset \mathbb{R}^n$ *be a closed and* *bounded set containing the origin. Assume there exists a function* $V : \mathbb{R}^n \to \mathbb{R}$ *continuous at the origin, finite for every* $x \in \Omega$, *and such that*

$$V(0) = 0 \text{ and } V(x) > 0, \forall x \in \Omega \{0\} \tag{7.38a}$$

$$V(x_{k+1}) - V(x_k) \leq -\alpha(x_k) \; \forall x_k \in \Omega \{0\} \tag{7.38b}$$

where $\alpha : \mathbb{R}^n \to \mathbb{R}$ *is a continuous positive definite function. Then* $x = 0$ *is asymptotically stable in* Ω.

Definition 7.2 *A function* $V(x)$ *satisfying conditions (7.38a)–(7.38b) is called a* Lyapunov Function.

The main idea of Theorem 7.2 can be explained as follows. We aim to find a scalar function $V(x)$ that captures qualitative characteristics of the system response, and, in particular, its stability. We can think of V as an energy function that is zero at the origin and positive elsewhere (condition (7.38a)). Condition (7.38b) of Theorem 7.2 requires that for any state $x_k \in \Omega$, $x_k \neq 0$ the energy decreases as the system evolves to x_{k+1}.

A proof of Theorem 7.2 for continuous dynamical systems can be found in [184]. The continuity assumptions on the dynamical system g is not used in the proof in [247, p. 609]. Assumption in equation (7.38b), however, together with the continuity of $V(\cdot)$ at the origin, implies that $g(\cdot)$ must be continuous at the origin.

Theorem 7.2 states that if we find an energy function which satisfies the two conditions (7.38a)–(7.38b), then the system states starting from any initial state $x_0 \in \Omega$ will eventually settle to the origin.

Note that Theorem 7.2 is only sufficient. If condition (7.38b) is not satisfied for a particular choice of V nothing can be said about stability of the origin. Condition (7.38b) of Theorem 7.2 can be relaxed to allow α to be a continuous positive semi-definite (psd) function:

$$V(x_{k+1}) - V(x_k) \leq -\alpha(x_k), \; \forall x_k \neq 0, \; \alpha \text{ continuous and psd.} \tag{7.39}$$

Condition (7.39) along with condition (7.38a) are sufficient to guarantee stability of the origin as long as the set $\{x_k : V(g(x_k)) - V(x_k) = 0\}$ contains no trajectory of the system $x_{k+1} = g(x_k)$ except for $x_k = 0$ for all $k \geq 0$. This relaxation of Theorem 7.2 is the so called Barbashin-Krasovski-LaSalle principle [183]. It basically means that $V(x_k)$ may stay constant and non zero at one or more time instants as long as it does not do so at an equilibrium point or periodic orbit of the system.

A similar result as Theorem 7.2 can be derived for *global* asymptotic stability, i.e., $\Omega = \mathbb{R}^n$.

Theorem 7.3 *Consider the equilibrium point* $x = 0$ *of system (7.30). Assume there exists a function* $V : \mathbb{R}^n \to \mathbb{R}$ *continuous at the origin, finite for every* $x \in \mathbb{R}^n$, *and such that*

$$\|x\| \to \infty \Rightarrow V(x) \to \infty \tag{7.40a}$$

$$V(0) = 0 \text{ and } V(x) > 0, \; \forall x \neq 0 \tag{7.40b}$$

$$V(x_{k+1}) - V(x_k) \leq -\alpha(x_k) \ \forall x_k \neq 0 \tag{7.40c}$$

where $\alpha : \mathbb{R}^n \to \mathbb{R}$ *is a continuous positive definite function. Then* $x = 0$ *is globally asymptotically stable.* ■

Definition 7.3 *A function* $V(x)$ *satisfying condition (7.40a) is said to be* radially unbounded.

Definition 7.4 *A radially unbounded Lyapunov function is called a* Global Lyapunov function.

Note that it was not enough just to restate Theorem 7.2 with $\Omega = \mathbb{R}^n$ but we also have to require $V(x)$ to be radially unbounded to guarantee global asymptotic stability. To motivate this condition consider the candidate Lyapunov function for a system in $\mathbb{R}^2$[176]

$$V(x) = \frac{x_1^2}{1 + x_1^2} + x_2^2, \tag{7.41}$$

which is depicted in Figure 7.1, where x_1 and x_2 denote the first and second components of the state vector x, respectively. $V(x)$ in (7.41) is not radially unbounded as for $x_2 = 0$

$$\lim_{x_1 \to \infty} V(x) = 1.$$

For this Lyapunov function even if condition (7.40c) is satisfied, the state x may escape to infinity. Condition (7.40c) of Theorem 7.3 guarantees that the level sets Ω_c of $V(x)$ ($\Omega_c = \{x \in \mathbb{R}^n : V(x) \leq c\}$) are closed.

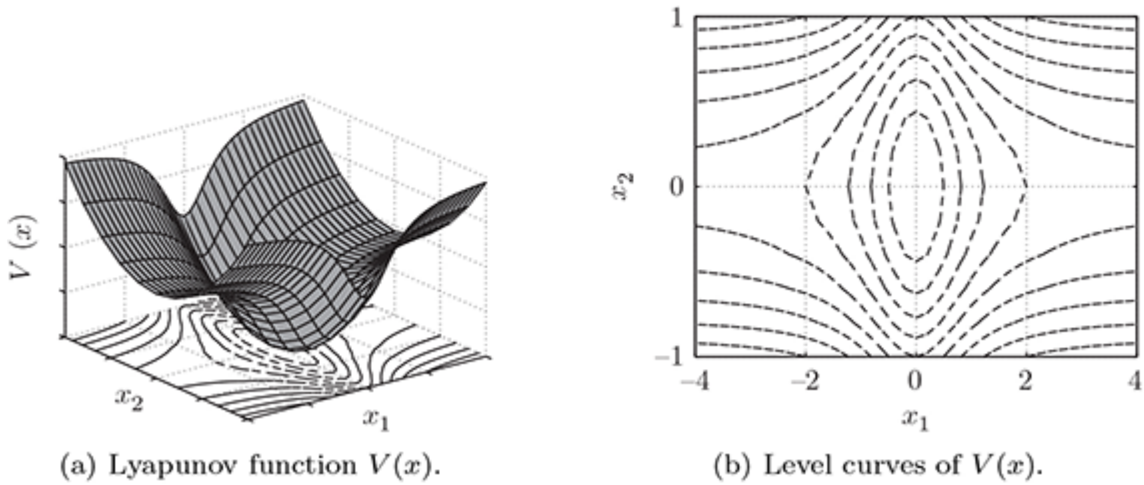

Figure 7.1 Lyapunov function (7.41).

The construction of suitable Lyapunov functions is a challenge except for linear systems. First of all one can quite easily show that for linear systems Lyapunov stability agrees with the notion of stability based on eigenvalue location.

Theorem 7.4 *A linear system* $x_{k+1} = Ax_k$ *is globally asymptotically stable in the sense of Lyapunov if and only if all its eigenvalues are strictly inside the unit circle.*

We also note that stability is always "global" for linear systems.

7.5.2 Quadratic Lyapunov Functions for Linear Systems

A simple effective Lyapunov function for linear systems is

$$V(x) = x'Px,\ P \succ 0 \tag{7.42}$$

which satisfies conditions (7.40a)–(7.40b) of Theorem 7.3. In order to test condition (7.40c) we compute

$$V(x_{k+1}) - V(x_k) = x'_{k+1}Px_{k+1} - x'_kPx_k = x'_kA'PAx_k - x'_kPx_k = x'_k(A'PA - P)x_k. \tag{7.43}$$

Therefore condition (7.40c) is satisfied if $P \succ 0$ can be found such that

$$A'PA - P = -Q,\ Q \succ 0. \tag{7.44}$$

Equation (7.44) is referred to as discrete-time Lyapunov equation. The following Theorem [75, p. 211] shows that $P \succ 0$ satisfying (7.44) exists if and only if the linear system is asymptotically stable.

Theorem 7.5 *Consider the linear system $x_{k+1} = Ax_k$. Equation (7.44) has a unique solution $P \succ 0$ for any $Q \succ 0$ if and only if A has all eigenvalues strictly inside the unit circle.*

Thus, a quadratic form $x'Px$ is always a suitable Lyapunov function for linear systems and an appropriate P can be found by solving (7.44) for a chosen $Q \succ 0$ iff the system's eigenvalues lie inside the unit circle. For nonlinear systems, determining a suitable form for $V(x)$ is generally difficult.

For a stable linear system $x_{k+1} = Ax_k$, P turns out to be the infinite time cost matrix

$$J_\infty(x_0) = \sum_{k=0}^{\infty} x'_kQx_k = x'_0Px_0 \tag{7.45}$$

as we can easily show. From

$$J_\infty(x_1) - J_\infty(x_0) = x'_1Px_1 - x'_0Px_0 = x'_0A'PAx_0 - x'_0Px_0 = -x'_0Qx_0 \tag{7.46}$$

we recognize that P is the solution of the Lyapunov equation (7.44). In other words the infinite time cost (7.45) is a Lyapunov function for the linear system $x_{k+1} = Ax_k$.

The conditions of Theorem 7.5 can be relaxed as follows.

Theorem 7.6 *Consider the linear system $x_{k+1} = Ax_k$. Equation (7.44) has a unique solution $P \succ 0$ for any $Q = C'C \succeq 0$ if and only if A has all eigenvalues inside the unit circle and (C, A) is observable.*

We can prove Theorem 7.6 in the same way as Callier and Desoer [75, p. 211] proved Theorem 7.5. In Theorem 7.6 we do not require that the Lyapunov function decreases at every time step, i.e., we allow Q to be positive *semi*definite. To understand this, let us assume that for a particular system state $\bar{x}$, V does not decrease, i.e., $\bar{x}'Q\bar{x} = (C\bar{x})'(C\bar{x}) = 0$. Then at the next time steps we have the rate of decrease $(CA\bar{x})'(CA\bar{x}), (CA^2\bar{x})'(CA^2\bar{x}), \ldots$. If the system (C, A) is observable, then for all $\bar{x} \neq 0$

$$\bar{x}' \left[C \ (CA)' \ (CA^2)' \cdots (CA^{n-1})'\right] \neq 0, \tag{7.47}$$

which implies that after at most $(n - 1)$ steps the rate of decrease will become nonzero. This is a special case of the Barbashin-Krasovski-LaSalle principle. Note that for C square and nonsingular Theorem 7.6 reduces to Theorem 7.5.

Similarly, we can analyze the controlled system $x_{k+1} = Ax_k + Bu_k$ with $u_k = Fx_k$ and the infinite time cost

$$J_\infty(x_0) = \sum_{k=0}^{\infty} x_k' Q x_k + u_k' R u_k \tag{7.48}$$

with $Q = C'C$ and $R = D'D$ with $\det(D) \neq 0$. We can rewrite the cost as

$$J_\infty(x_0) = \sum_{k=0}^{\infty} x_k'(Q + F'RF)x_k = \sum_{k=0}^{\infty} x_k' \begin{bmatrix} C \\ DF \end{bmatrix}' \begin{bmatrix} C & DF \end{bmatrix} x_k \tag{7.49}$$

for the controlled system $x_{k+1} = (A+BF)x_k$. The infinite time cost matrix P can now be found from the Lyapunov equation

$$(A + BF)'P(A + BF) - P = \begin{bmatrix} C \\ DF \end{bmatrix}' \begin{bmatrix} C & DF \end{bmatrix}. \tag{7.50}$$

According to Theorem 7.6 the solution P is unique and positive definite iff $(A+BF)$ is stable and $\left[\begin{pmatrix} C \\ DF \end{pmatrix}, (A + BF)\right]$ is observable. This follows directly from the observability of (C, A). If (C, A) is observable, then so is $(C, A + BF)$ because feedback does not affect observability. Observability is also not affected by adding the observed outputs DFx.

From (7.44) it follows that for stable systems and for a chosen $Q \succ 0$ one can always find $P \succ 0$ solving

$$A'PA - P + Q \preceq 0. \tag{7.51}$$

This *Lyapunov inequality* shows that for a stable system we can always find a P such that $V(x) = x'Px$ decreases at a desired "rate" indicated by Q. We will need this result later to prove stability of receding horizon control schemes.

7.5.3 1/∞ Norm Lyapunov Functions for Linear Systems

For $p = \{1, \infty\}$ the function

$$V(x) = \|Px\|_p$$

with $P \in \mathbb{R}^{l \times n}$ of full column rank satisfies the requirements (7.40a), (7.40b) of a Lyapunov function. It can be shown that a matrix P can be found such that condition (7.40c) is satisfied for the system $x_{k+1} = Ax_k$ if and only if the eigenvalues of A are inside the unit circle. The number of rows l necessary in P depends on the system. The techniques to construct P are based on the following theorem [177, 237].

Theorem 7.7 *Let $P \in \mathbb{R}^{l \times n}$ with* rank(P) = n *and* $p \in \{1, \infty\}$. *The function*

$$V(x) = ||Px||_p \tag{7.52}$$

is a Lyapunov function for the discrete-time system

$$x_{k+1} = Ax_k, \; k \geq 0, \tag{7.53}$$

if and only if *there exists a matrix $H \in \mathbb{R}^{l \times l}$, such that*

$$PA = HP, \tag{7.54a}$$

$$||H||_p < 1. \tag{7.54b}$$

An effective method to find both H and P was proposed by Christophersen and Morari in [88].

To prove the stability of receding horizon control, later in this book, we will need to find a $\tilde{P}$ such that

$$\|\tilde{P}Ax\|_\infty - \|\tilde{P}x\|_\infty + \|Qx\|_\infty \leq 0, \;\; \forall x \in \mathbb{R}^n. \tag{7.55}$$

Note that the inequality (7.55) is equivalent to the Lyapunov inequality (7.51) when the squared two-norm is replaced by the 1– or ∞–norm. Once we have constructed a P and H to fulfill the conditions of Theorem 7.7 we can easily find $\tilde{P}$ to satisfy (7.55) according to the following lemma.

Lemma 7.1 *Let P and H be matrices satisfying conditions (7.54), with P full column rank. Let $\sigma = 1 - ||H||_\infty$, $\rho = ||QP^{\#}||_\infty$, where $P^{\#} = (P'P)^{-1}P'$ is the left pseudoinverse of P. Then, the square matrix*

$$\tilde{P} = \frac{\rho}{\sigma} P \tag{7.56}$$

satisfies condition (7.55).

Proof: Since $\tilde{P}$ satisfies $\tilde{P}A = H\tilde{P}$, we obtain $-\|\tilde{P}x\|_\infty + \|\tilde{P}Ax\|_\infty + \|Qx\|_\infty = -\|\tilde{P}x\|_\infty + \|H\tilde{P}x\|_\infty + \|Qx\|_\infty \leq (\|H\|_\infty - 1)\|\tilde{P}x\|_\infty + \|Qx\|_\infty \leq (\|H\|_\infty - 1)\|\tilde{P}x\|_\infty + \|QP^{\#}\|_\infty \|Px\|_\infty = 0$. Therefore, (7.55) is satisfied. ■