1

Control Theory: Basic Concepts

This chapter presents basic concepts of control theory, which will be used in the remaining book.

In section 1.1, we present the general *control/plant* model. In section 1.2, we explain why the introduction of digital sensors and actuators in systems has fundamentally modified the issue of controlled stability. Finally, we introduce the model of *switched* systems, and explains their advantages compared with general systems (section 1.2.3). We then explain in section 1.3 how the notion of *invariant sets* can be used for proving safety and stability properties of controlled systems.

1.1. Model of control systems

A control system is generally divided into a controlled part, called a *plant*, and a *controller*. The plant is generally described as a dynamic time-invariant, possibly uncertain, system governed by equations of the form:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), w(t)) & [1.1] \\ y(t) = g(x(t)), & [1.2] \end{cases}$$

where $x(t) \in \mathbb{R}^n$ is the *system state*, $u(t) \in \mathbb{R}^m$ is the *control input* $y(t) \in \mathbb{R}^p$ is the *output*, $w(t) \in \mathcal{W} \subset \mathbb{R}^q$ is a *disturbance* (or external input) and $\mathcal{W}$ is an assigned compact set. We will refer to $\mathbb{R}^n$ as the *state space* of the system. The general theory of control focuses on *feedback control*: the controller is fed with state signal $x(t)$ coming from the plant, and issues a control input $u(t)$ to the plant. A typical layout of a feedback control system is shown in Figure 1.1. Under general conditions (continuity for u and w, and Lipschitz property for f), the system admits a unique solution $x(t)$ on $\mathbb{R}_{\geq 0}$. Equations [1.1] and [1.2] are often simplified by disregarding $w(t)$, and assuming that $y = x$.

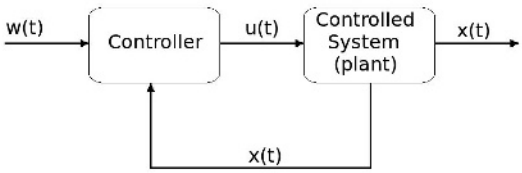

Figure 1.1. *Control/plant model*

An important subclass is the *linear time-invariant* (LTI) framework, for which [1.1] and [1.2] become: $\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Ew(t) \\ y(t) = Cx(t) \end{cases}$
for matrices A, B, C, E of appropriate size with constant coefficients. A *discrete-time LTI* system is a system governed by an equation of the form: $x(t+1) = Ax(t) + Bu(t) + Ew(t)$.

When a system is governed by an equation of the form $\dot{x}(t) = Ax(t)$, where A is a matrix whose eigenvalues have negative real parts, the origin is a *stable equilibrium* point to which the system converges from any initial point of $\mathbb{R}^n$. Given a plant governed by an equation of the form $\dot{x}(t) = Ax(t) + Bu(t)$ with $(A, B) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$, a typical problem of linear control theory is to find a stabilizing controller governed by an equation of the form $u(t) = Kx(t)$ with $K \in \mathbb{R}^{m\times n}$. This essentially amounts to finding coefficient values of K that make the real parts of the eigenvalues of $A + BK$ negative.

1.2. Digital control systems

1.2.1. *Digitization*

With the emergence of digital computers, a control system has to handle data that come from the periodic sampling of signals. In such a context, a control system is said to be *sampled data* or *digital control system*. There, a system described by differential equations (which involve continuous-valued variables that depend on continuous time) is controlled by a discrete-time controller described by difference equations, which involve continuous-valued variables that depend on discrete time. As explained in [ANT 02], a digital control system can be divided into three parts, the plant, the interface, and the controller as shown in Figure 1.2.

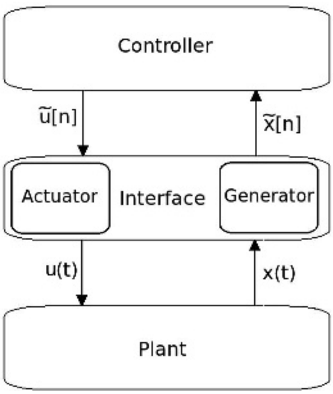

Figure 1.2. *Digital control/plant model (from [ANT 02])*

The system to be controlled (*plant*) is modeled as a time-invariant continuous-time system governed by equations [1.1] and [1.2] where, for the sake of simplicity, we disregard disturbance and assume that the output function is the identity map (i.e. we have $\dot{x} = f(x, u)$ and $y = x$).

The *controller* is a discrete event system modeled as a deterministic automaton. The action of the controller can be described by equations of the form:

$$\begin{cases} \tilde{s}[n] = \delta(\tilde{s}[n-1], \tilde{x}[n]) \\ \tilde{u}[n] = \phi(\tilde{s}[n]), \end{cases}$$

where δ is the state transition function of the controller and ϕ is the output function of the controller. Tildes are used to indicate that the particular signal is made up of symbols. The index n is here analogous to a time index in that it specifies the order of the symbols in the sequence. An argument in brackets, for example, $\tilde{x}[n]$, represents the nth symbol from a set. The input signal $\tilde{x}$ and the output signal $\tilde{u}$ associated with the controller are a sequence of symbols, rather than continuous-time signals. Note that there is *no delay* in the controller: the state transition, from $\tilde{s}[n-1]$ to $\tilde{s}[n]$, and the controller symbol, $\tilde{u}[n]$, occur immediately after the occurrence of plant symbol $\tilde{x}[n]$.

The controller and plant cannot communicate directly because each utilizes different types of signals. Thus, an *interface* is required that can convert continuous-time signals to sequences of symbols and vice versa. The interface consists of a memoryless map γ called *actuator*, and a memoryless map α called *generator*. The actuator converts a controller symbol $\tilde{u}[n]$ to a constant plant input of the form $u(t) = \gamma(\tilde{u}[n])$. Since the plant input, u, can only take on certain constant values, where each value is associated with a particular controller symbol, the plant input signal $u(t)$ is *piecewise constant*, and may change only when a controller symbol occurs. Such a piecewise continuous command signal issued by the actuator is illustrated in Figure 1.3. The generator is a function α that maps the real-valued state vector $x(t)$ of the plant into a plant symbol of the form $\tilde{x}[n] = \alpha(x(t))$. Note that $\tilde{x}$ does not change continuously, but only when a *plant event* occurs. There are two different models of plant event: in the *state-triggered* model, a plant event occurs when the plant state x crosses the boundary of two predefined state regions; in the *time-triggered* model, a plant event occurs periodically when the signal $\tilde{x}$ issued by the generator corresponds to a *periodic sampling* of the plant output x, as illustrated in Figure 1.4.

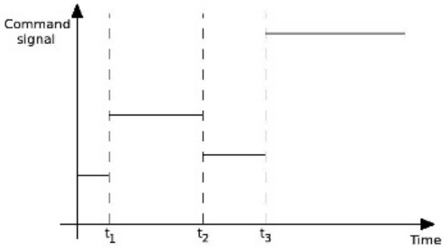

Figure 1.3. *Staircase command signal $u(t)$ issued by the actuator as it receives controller symbols $\tilde{u}[1], \tilde{u}[2], \ldots$ at time t_1, t_2, ... (from [ANT 02])*

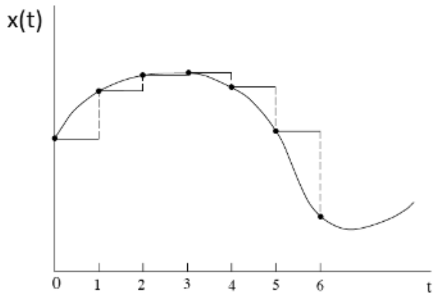

Figure 1.4. *Controller symbols $\tilde{x}[1], \tilde{x}[2], \ldots$ produced by the generator by sampling of the plant output signal $x(t)$ (time-triggered plant event model) (from [ANT 02])*

Note that, since it is assumed that there is no delay in the controller, the command signal $u(t)$ issued by the actuator is *synchronized* with the signal $x(t)$ issued by the generator. In the time-triggered model, the command $u(t)$ is therefore itself *periodic*. (In Figure 1.4, the stair length is constant and equal to τ.)

1.2.2. *Quantization*

Digitization also has an effect sometimes known as *quantization* (see, e.g., [PAT 05]). Suppose that the signal u now takes its values on a *finite* domain U, instead of a dense (possibly bounded) domain or an infinite discrete domain. This means that, in Figure 1.3, the plant input signal $\mathbf{u}(t)$ is a staircase signal that can take only a *finite* number of values. In such a situation, there are many systems (even LTI systems) for which there is no control function that ensures stabilization, that is convergence to a unique equilibrium point (see, e.g., [BRO 00]). The controller can only achieve *practical stability*, that is convergence into a bounded set instead of a single point for general stability The goal is then to synthesize controllers that are capable of steering the system to within sufficiently small neighborhoods of the equilibrium. The size of the final set within which the trajectories are confined is a measure of performance of the controlled dynamics. Hence, for a quantized system, the notion of *minimal invariant* set (once a proper notion of size has been defined) is useful for describing zones of practical stability.

1.2.3. *Switching*

A *switched system* is a digital quantized control system that consists of a finite family of continuous subsystems and a rule that controls the switching between them. More precisely, we have the following definition:

DEFINITION 1.1.– *A* switched system *is a quadruple $\mathcal{S} = (\mathbb{R}^n, U, \mathcal{U}, \mathcal{F})$, where $\mathbb{R}^n$ is the state space; $U = \{1, \ldots, N\}$ is the finite set of modes; $\mathcal{U}$ is the set of piecewise constant functions from $\mathbb{R}_{\geq 0}$ to U, continuous from the right; and $\mathcal{F} = \{f_1, \ldots, f_N\}$ is a collection of smooth vector fields indexed by U.*

A *switching signal* of $\mathcal{S}$ is a function $\mathbf{u} \in \mathcal{U}$. A piecewise $\mathcal{C}^1$ function $\mathbf{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is said to be a *trajectory* of $\mathcal{S}$ if it is continuous and there exists a switching signal $\mathbf{u} \in \mathcal{U}$ such that, at each $t \in \mathbb{R}_{\geq 0}$ where the function $\mathbf{u}$ is continuous, $\mathbf{x}$ is continuously differentiable and satisfies:

$$\dot{\mathbf{x}}(t) = f_{\mathbf{u}(t)}(\mathbf{x}(t)).$$

The times at which the switching signal changes its values are called the *switching instants*. The scheme of switched systems is represented in Figure 1.5. It is easy to see that a quantized discrete-time LTI system is a particular subclass of switched systems (for which the function $f_{\mathbf{u}(t)}(x(t))$ is of the form $Ax(t) + B\mathbf{u}(t)$). However, the class of switched systems is much more general.

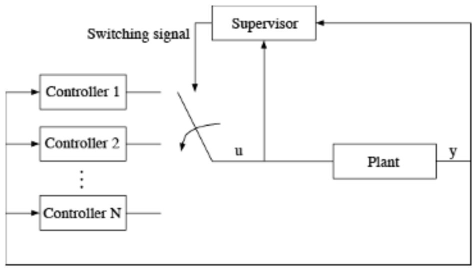

Figure 1.5. *Scheme of a switching controller*

In recent years, control techniques based on switching between different controllers, as shown in Figure 1.5, have been used in order to achieve stability and improve transient response. The importance of such control methods also arises from the existence of systems that cannot be stabilized by a single continuous feedback law (see [BRO 83]). In contrast, even if the different components of a switched system working in their proper mode have no (common) equilibrium, it is still possible to control the global system in order to make its behavior similar to those of conventional stable systems near equilibrium (see, e.g., [BUI 05]). Switched systems have thus found numerous applications switching power converters and many other fields (see [LIB 99]).

Caveat

Note, however, it is possible for a switched system to be unstable even when all the subsystems are stable around a common equilibrium point. This is true even when the subsystems are linear, as illustrated in the following example (see [ANT 02]). Consider the switched system $\dot{x}(t) = A_u x(t)$, where $x \in \mathbb{R}^2$, $u \in \{1, 2\}$, and $A_1 = \begin{pmatrix} -1 & 100 \\ 10 & -1 \end{pmatrix}$, $A_2 = \begin{pmatrix} -1 & -10 \\ 100 & -1 \end{pmatrix}$, with a (state-triggered) switching signal that applies A_1 (respectively A_2) when x is in the second and fourth (respectively first and third) quadrants. Both A_1 and A_2 are stable since their eigenvalues $\lambda_{1,2} = -1 \pm j\sqrt{1{,}000}$ have negative real parts. However, their trajectories are unstable (see Figure 1.6). Such a phenomenon takes place because the intervals between the switchings of the dynamics decrease to 0 as time goes to infinity. This can be avoided by imposing a minimum duration (called *dwell time*) between two switching instants. This can be easily enforced for the class of *sampled* switched systems that we will study in this book for which switchings occur with a fixed period τ.

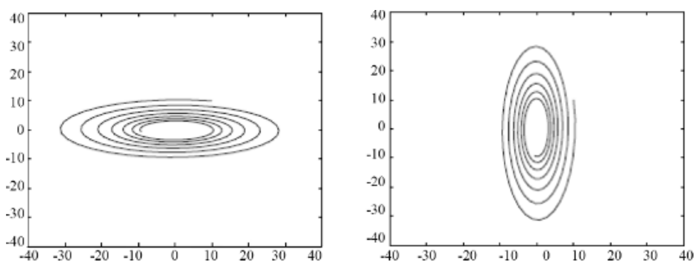

Figure 1.6. *Unstable trajectory of switched system consisting of stable subsystems (from [ANT 02])*

1.3. Control of switched systems using invariant sets

We now consider the problem of synthesizing controllers for switched systems. This results in finding a switching signal that controls the system in order to satisfy some given properties. We focus on the safety and stability properties. We explain that the controller synthesis problem is related to the construction of controlled invariant sets.

1.3.1. *Controlled invariants*

Given a dynamic system, a subset $\mathcal{I}$ of the state space is said to be *invariant* if it has the following property: if it contains the system state at some time, then it will contain it also in the future [BLA 99][1]. We have:

$$x(t) \in \mathcal{I} \Rightarrow x(t') \in \mathcal{I}, \text{ for all } t' \geq t.$$

The concept of invariance can be easily extended to the case in which a control input is present. In this case, we say that a set R is *controlled invariant* if, for all initial conditions chosen in R, we can keep the trajectory inside $\mathcal{I}$ *by means of a proper switching signal*. Let us now explain why controlled invariants are useful for proving *safety* and *stability* properties of a switched system.

1.3.2. *Safety control problem*

The safety property is typically encoded as a subset S of the continuous state space, called *safe set*. In a simple formulation, S is a box set given by the minimum and maximum values tolerated for each state variable. The associated safety properties suffice to describe typical requirements of direct-current to direct-current (DC-DC) power converters such as voltage regulation, current limitation, maximal current and voltage ripple.

Safety control problem: given a safe set S, determine whether a switching signal $\mathbf{u}$ exists such that if $x(0) \in S$, then $x(t) \in S$ for $t \geq 0$.

1 This property is often called "positively invariant" instead of just "invariant" in the literature.

Several approaches [ASA 00, TOM 00] have been proposed to solve the safety control problem. The idea of these approaches is to obtain a *controlled invariant* W that is included into S for an appropriate switching signal $\mathbf{u}$. If such a set W exists and if the initial state is in W, then the system is ensured to stay in W, hence in the safe set S. In [ASA 00], an abstract algorithm is proposed to synthesize controlled invariants using a backward iterative computation of reachable states. Furthermore, the set W computed is the *maximal controlled invariant* subset of S (it contains all other controlled invariant included into S). In [TOM 00], the controller synthesis problem is formulated as a game between controller and disturbance. We can then find Hamilton–Jacobi equations whose solutions describe the boundaries of the maximal safe set, and derive an associated maximally permissive controller. In Chapter 3, we will discuss methods to synthesize safety controllers that are adapted to the simpler context of sampled switched systems that we consider here.

1.3.3. *Stability control problem*

Given a certain region R, many controlled invariants subsets of R exist. If, instead of looking for maximal invariant subsets, we look for finding invariants as small a size as possible around a given operating point, we get a characterization of a controller with the smallest deviation from the point, and obtain a steady-state behavior with "minimum ripple" (see [SEN 03]). When periodic solutions of the system exist, we should be able to synthesize a *stability controller* that makes the trajectories converge to such periodic solutions of the system, also called *limit cycles*.

Stability control problem: given a region R, determine a switching signal $\mathbf{u}$ that makes the trajectories starting in R converge to a subregion as small as possible, ideally a limit cycle.

In Chapter 4, we will discuss a method based on a procedure of state-space decomposition, and iterated computation of forward reachable states for synthesizing stability controllers.

1.3.4. *Other controllers*

We will also give some hints to solve the problem of synthesizing *robust safety* controllers that maintain the plant in a safety region in the presence of disturbance or uncertainty, as well as *reachability controllers*, which drive the plant in finite time from an initial operating region to a desired operating region (see Chapter 6).

1.4. Notes

The common approach for stability analysis of dynamic systems is based on Lyapunov's method, which relies on the concept of a *Lyapunov function* or generalized energy function. Essentially, a Lyapunov function for an equilibrium point x_e of the system $\dot{x} = f(x)$ is a differentiable function $V(x)$ that has a strict minimum as x_e, and so that its derivative $\dot{V}(x) = \frac{\partial V(x)}{\partial x} \cdot f(x)$ along the system trajectories is negative in some neighborhoods of the equilibrium. Various converse theorems establish the existence of a Lyapunov function whenever the equilibrium point is stable (in the appropriate sense). There are fundamental connections between the notion of Lyapunov function and that of invariance. Precisely, given a Lyapunov function, its level sets are the boundaries of invariant sets. In this book, we will not use Lyapunov functions, but focus on invariant sets.

The context of section 1.2 is mainly taken from [ANT 02].