2

Models of Dynamical Systems

In this chapter we review some elements of dynamical system modeling. A system model is an *mathematical abstraction* of an actual real-world system: a necessarily simplified representation, attempting to capture those aspects essential to explain the system's behavior. Because *we can only reason about the model*, we quickly adopt the convention that system means the mathematical system model and we state explicitly when we refer to the real-world system it represents. It is very important to keep the abstraction process in mind however. For example, in most cases, our mathematical abstractions of physical or natural systems can only be very rough approximations, in order to obtain tractable models useful for design purposes. In contrast, in communications or computer science for example, models can be developed that represent quite precisely the behavior of the engineered systems (communication protocols, programming language semantics, execution on a digital circuit, etc.). Robustness considerations should guide the types of questions we ask about various models, so that their answers remain relevant once the level of uncertainty about the real-world system is factored in.

2.1 General and Input-Output Systems

Abstractly, a general dynamical system Σ is just a constraint defining a set of possible signals [PW98]. Namely, Σ can defined as a triple $(\mathbb{T}, \mathbb{W}, \mathcal{B})$, where $\mathbb{T}$ is a subset of $\mathbb{R}$ called the *time axis*, $\mathbb{W}$ is a set, and $\mathcal{B}$ is a subset of the function space $\mathbb{W}^{\mathbb{T}}$ called the *behavior*. Note that $\mathbb{W}^{\mathbb{T}}$ denotes the set of maps from $\mathbb{T}$ to $\mathbb{W}$. Note that we call any function space $\mathbb{S}^{\mathbb{T}}$, with $\mathbb{S}$ a set, a *signal space*, and its elements are called *signals*. We are used to describe systems directly with equations, but this is a priori just one way of doing it, and moreover there are typically multiple ways of representing the same behavior of interest with equations. It can be quite interesting to develop a general systems theory from this abstract point of view [PW98], but we will adopt a somewhat more operational perspective and directly deal with some important classes of behavioral descriptions. A continuous-time (CT) system is one for which $\mathbb{T}$ an interval of $\mathbb{R}$ (possibly infinite), and a discrete-time (DT) system is one for which $\mathbb{T} = \mathbb{Z}$ or $\mathbb{T} = \mathbb{N}$. DT systems could be obtained for example by sampling CT systems at a discrete set of points, as discussed in chapter 3. When combining DT and CT models, or several DT systems with different sampling periods, we can always embed the time axes back into $\mathbb{R}$ to reason about the global system.

Given two signal spaces $\mathcal{U} = \mathbb{U}^{\mathbb{T}}, \mathcal{Y} = \mathbb{Y}^{\mathbb{T}}$, an *input-output* system is a subset of $\mathcal{U} \times \mathcal{Y}$, i.e., a binary *relation* $\mathcal{R}_{uy}$ between the sets $\mathcal{U}$ and $\mathcal{Y}$. Then $\mathcal{U}$ is called the input signal space and $\mathcal{Y}$ the output signal space. This is a particular class of behavioral models, with $\mathbb{W} = \mathbb{U} \times \mathbb{Y}$ and $\mathcal{B} := \mathcal{R}_{uy} \subset \mathcal{U} \times \mathcal{Y}$, where we have explicitly (and possibly quite arbitrarily) stated which signals are considered as inputs and which signals are considered as outputs. Hence an input-output system is interpreted as a set of possible input/output signal pairs $(u, y) \in \mathcal{U} \times \mathcal{Y}$. Note that to an input signal u we can associate in general a *set* (possibly empty) of possible output signals $\{y | (u, y) \in R_{uy}\}$. This allows us to take into account for example the presence of unknown or unmodeled *initial conditions*, as well as other sources of nondeterminism in the system such as noise. Hence, a give signal u can produce two different output signals y for the same system started with different initial conditions.

Input-output methods view systems as black boxes that relate external signals, independently of any internal state-space representation of these systems, see Fig. 2.1. They have important applications in linear and

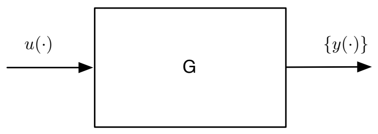


Figure 2.1: A system with input u and output y.

nonlinear control, e.g., in robust control, and are particularly useful for the analysis of large-scale complex and interconnected systems. Pioneering work in this area was done by Zames, Sandberg, Popov, Desoer and others. Standard references on input-output methods include [Wil71, DV09, Saf80].

2.1.1 Signal Spaces

The function spaces $\mathcal{U}, \mathcal{Y}$ of most interest for control purposes are usually the L^p spaces[1], $1 \le p \le \infty$. To define these spaces, we equip a set $\mathbb{S}$ of signal values with a norm $|\cdot|$ (for example, the Euclidean norm on $\mathbb{S} = \mathbb{R}^n$), and we fix a measure μ on $\mathbb{T}$. Then we define, for $p \in [1, \infty)$

$$L^p(\mathbb{T}, \mathbb{S}, \mu) := \left\{ x \in \mathbb{S}^{\mathbb{T}}, \mu - \text{measurable} \middle| \int_{\mathbb{T}} |x(t)|^p \mu(dt) < \infty \right\},$$

and

$$L^\infty(\mathbb{T}, \mathbb{S}, \mu) := \left\{ x \in \mathbb{S}^{\mathbb{T}}, \mu - \text{measurable} \middle| \operatorname*{ess\,sup}_{\mathbb{T}} |x(t)| < \infty \right\}.$$

In the usual contexts where these spaces arise, we have $\mathbb{S} = \mathbb{R}^n$, for some $n \in \mathbb{N}$, and $\mathbb{T} = \mathbb{R}$ or $\mathbb{R}_+$ is fixed with μ the Lebesgue measure, in which case we denote $L^p(\mathbb{T}, \mathbb{S}, \mu) =: L^p_n(\mathbb{T})$, where $\mathbb{T}$ can be omitted if it is understood. Hence we have for example

$$L^p_n(\mathbb{R}_+) = \left\{ f : \mathbb{R}_+ \to \mathbb{R}^n \middle| \int_0^\infty |f(t)|^p dt < \infty \right\},$$

and the space L^∞ consists of functions that are essentially bounded, i.e., bounded on their domain of definition except on a set of Lebesgue measure zero.

If $\mathbb{T}$ is a discrete set such as $\mathbb{Z}$ or $\mathbb{N}$ with μ the counting measure, then we denote $L^p(\mathbb{T}, \mathbb{R}^n, \mu) =: \ell^p_n$, so that, in the case $\mathbb{T} = \mathbb{Z}$ for example,

$$\ell^p_n := \left\{ \{x_k\}_{k \in \mathbb{Z}} \middle| \sum_{k=-\infty}^{\infty} |x_k|^p < \infty \right\}, \quad p \in [1, \infty)$$
$$\ell^\infty_n := \left\{ \{x_k\}_{k \in \mathbb{Z}} \middle| \exists M < \infty \text{ s.t. } |x_k| \le M, \forall k \in \mathbb{Z} \right\}.$$

The ℓ^p spaces arise in the context of discrete-time (DT) systems, the L^p spaces in the context of continuous-time (CT) systems.

The L^p and ℓ^p spaces, for $p \in [1, \infty]$, are *normed* vector spaces (see the remark below Theorem 2.1.1). The p-norm $\|\cdot\|_p$ or $\|\cdot\|_{L^p}$ of $u : [0, \infty) \to \mathbb{R}^n$ is defined as

$$\|u\|_p := \left(\int_0^\infty |u(\tau)|^p d\tau \right)^{1/p}, \quad 1 \le p < \infty,$$
$$\|u\|_\infty := \operatorname*{ess\,sup}_{\tau \ge 0} |u(\tau)|,$$

[1] also denoted $L_p, \mathcal{L}^p, \mathcal{L}_p, \mathcal{L}^n_p, \ldots$ in various other references. We follow here the more or less standard notation found in analysis textbooks.

and similarly for other domains of definition $\mathbb{T}$, e.g., $u : \mathbb{R} \to \mathbb{R}^n$ or discrete-time sequences. With these norms, the L^p spaces are in fact *complete*, i.e., Banach spaces. An important consequence is that contractive maps on these spaces have a (unique) fixed point. Moreover, L^2 is a Hilbert space, with inner product

$$\langle f, g \rangle = \int f(t)^T g(t) dt,$$

and similarly for ℓ^2 with

$$\langle x_1, x_2 \rangle = \sum_k x_{1,k}^T \, x_{2,k}.$$

The most important L^p spaces are those obtained for $p = 2$, capturing the notion of energy (the squared 2-norm), $p = \infty$, capturing the worst case behavior of signals, and $p = 1$.

Remark 2.1.1. Sometimes the norm on $\mathbb{R}^n$ used in the definition of L^p_n or ℓ^p_n is assumed to be the p norm for the same value of p. Recall that the p-norm on $\mathbb{R}^n$ is defined as $|x|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$, for a vector x with components x_i. So for example the p-norm on ℓ^p is then $\left(\sum_{k=-\infty}^{\infty} \sum_{i=1}^n |x_{k,i}|^p\right)^{1/p}$. Note that the assumption is in fact made implicitly above when defining L^2 and ℓ^2, which uses the fact that the 2-norm on $\mathbb{R}^n$ comes from the standard inner product (dot product).

Relations Between L^p Spaces

Theorem 2.1.1. *[Hölder, Schwarz and Minkowski Inequalities] For $p, q \in [1, \infty]$, such that*

$$\frac{1}{p} + \frac{1}{q} = 1, i.e., q = \frac{p}{p-1},$$

if $f \in L^p_n$, $g \in L^q_n$, then $f^T g$ defined by $x \mapsto f(x)^T g(x)$ is in L^1, and we have the Hölder inequality

$$\|f^T g\|_1 \leq \|f\|_p \|g\|_q. \tag{2.1}$$

When $p = q = 2$, this inequality is known as the Cauchy-Schwarz inequality. Moreover, if $f, g \in L^p_n$, then $f + g \in L^p_n$ and we have the Minkowski inequality

$$\|f + g\|_p \leq \|f\|_p + \|g\|_p,$$

for all $p \in [1, \infty]$.

Remark 2.1.2. Note that the Minkowski inequality is crucial in establishing that L^p is in fact a vector space, by showing that it is stable under addition.

Remark 2.1.3. The left-hand side (LHS) of the Hölder inequality above is, more explicitly $\int |\sum_{i=1}^n f_i(t) g_i(t)| dt$. However, since the right-hand side (RHS) involves only the components $|f_i(t)|$, we can write the LHS instead as $\int \sum_{i=1}^n |f_i(t) g_i(t)| dt$, obtaining an priori stronger inequality. Similarly, writing the left-hand side $|\int f^T(x) g(x) dt|$ instead of the LHS of (2.1) does not weaken the theorem.

Now suppose that the domain of definition of f, denoted I, has *finite Lebesgue measure* $|I| < \infty$. First, if $f \in L^\infty$, then

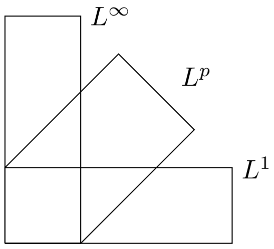
$$\left(\int_I |f(t)|^p \right)^{1/p} \leq \|f\|_\infty |I|^{1/p},$$

so that $f \in L^p$, for all $p \in [1, \infty]$. Next if $1 \leq r < s < \infty$ and $f \in L^s_n$, then by the Hölder inequality (2.1) with $p := s/r$ and $q = p/(p-1)$, we have

$$\int_I |f(t)|_r^r dt = \int_I \sum_{i=1}^n |f_i(t)|^r \cdot 1 \; dt \leq \left(\int_I \sum_{i=1}^n |f_i(t)|^{r(s/r)} dt \right)^{r/s} n^{1/q} |I|^{1/q},$$

so that $\|f\|_r \leq \|f\|_s \, n^{1/qr} |I|^{1/qr}$. In other words, if I has finite measure, then integrability of $\|f\|^p$ for a function f is more restrictive for larger values of p, and we have $L^s(I) \subset L^r(I)$ for $1 \leq r < s \leq \infty$.

Typically for us however, the domain of definition of the signals (say $\mathbb{R}_+$ or $\mathbb{R}$) does not have finite measure, and these inclusions do not hold any more. For example, $t \mapsto 1$ is in $L^\infty(\mathbb{R}_+)$ but not in $L^p(\mathbb{R}_+)$ for any $p < \infty$. In fact, for any $1 < p < \infty$, we have the following Venn diagram, where all drawn subsets are non empty



Note in particular the following fact.

Proposition 2.1.2. *If $f : \mathbb{R}_+ \to \mathbb{R}$ and $f \in L^1_n \cap L^\infty_n$, then $f \in L^p_n$ for $p \in [1, \infty]$.*

Proof. Denote $I := \{t || f(t)| \geq 1\}$. Then since $f \in L^1$, the Lebesgue measure of I is finite and since $f \in L^\infty$ we must then have $\int_I |f(t)|^p dt < \infty$. Now with I^c the complement of I, we have

$$\infty > \int_{I^c} |f(t)|dt \geq \int_{I^c} |f(t)|^p dt, \text{ for all } p \in [1, \infty).$$

Together, these two observations give the conclusion. $\square$

Example 2.1.1. The function $t \mapsto \frac{1}{1+t}$, from $\mathbb{R}_+$ to $\mathbb{R}$, is in L^∞ and L^2 but not in L^1. Finish populating the Venn diagram above with examples of functions, for $p = 2$.

Extended L^p Spaces

Integrability of signals on $\mathbb{R}_+$ or $\mathbb{R}$ is often too strong a requirement to study dynamical systems, e.g., when dealing with unstable systems or finite power signals. For this reason, we are led to consider locally integrable functions, i.e., functions that are integrable on every bounded interval. For $T \geq 0$, define the (linear) truncation operator P_T by

$$(P_T f)(t) := f_T(t) := \begin{cases} f(t), & 0 \leq t \leq T, \\ 0, & t > T, \end{cases}$$

for any function f of time. P_T is a projection since $P_T^2 = P_T$. We say that f_T is obtained by *truncating* f at T. We define the extended space L^{pe} by

$$L^{pe}_n(\mathbb{R}_+) = \{f : \mathbb{R}_+ \to \mathbb{R}^n | \|f_T\|_p < \infty, \forall T \geq 0\}.$$

That is, $f \in L^{pe}$ if $f_T \in L^p$ for all $T \geq 0$.

Example 2.1.2. The function $t \mapsto \frac{1}{1+t}$ is not in L^1 but it is in L^{1e}.

Note the following facts

(i) $\forall f \in L^{pe}$, the map $T \mapsto \|f_T\|_p$ is monotonically increasing.

(ii) $\forall f \in L^p$, $\|f_T\|_p \to \|f\|_p$ as $T \to \infty$.

2.1.2 Some Properties of Input-Output Systems

Causality

Most dynamical system models used in control theory are nonanticipative, i.e., their current behavior cannot depend on the future. This can be defined formally as follows

Definition 2.1.1. A mapping $H : L^{pe} \to L^{pe}$ is *causal* (or nonanticipative) if $P_T H P_T = P_T H$ for all $T \geq 0$, i.e.,

$$(Hu_T)_T = (Hu)_T, \; \forall T \geq 0. \tag{2.2}$$

Example 2.1.3. Let $h \in L^1$ be the impulse response of a linear time-invariant system $H : L^2 \to L^2$ with zero initial conditions, so that $Hu = h * u, \ \forall u \in L^2$, i.e.,

$$(Hu)(t) = \int_{-\infty}^{\infty} h(t-\tau)u(\tau)d\tau, \ \ \forall t \in \mathbb{R}.$$

Then H is causal if and only if $h(t) = 0$ almost everywhere on $(-\infty, 0)$.

In words, Definition 2.1.1 says that if H is causal then the responses to the inputs u and u_T are the same up to time T, i.e., the output does not depend on the future values (for $t > T$) of the inputs. Note that an equivalent definition of a causal system is

$$\forall u_1, u_2 \in L^{pe}, P_T u_1 = P_T u_2 \Rightarrow P_T H u_1 = P_T H u_2. \tag{2.3}$$

That is, two input signals u_1 and u_2 that are identical on $[0, T]$ give identical output responses Hu_1 and Hu_2 on $[0, T]$. Indeed, if H is causal according to definition 2.1.1, then $P_T H u_1 = P_T H P_T u_1 = P_T H P_T u_2 = P_T H u_2$. Conversely, if (2.3) is satisfied, then because $P_T(P_T u) = P_T u$, we have $P_T H P_T u = P_T H u$, i.e., (2.2).

Remark 2.1.4. Note that systems given in a standard state space form, e.g.,

$$\begin{aligned} \dot{x} &= f(x, u, t) \\ y &= h(x, u, t) \end{aligned}$$

are always causal, when viewed as input-output systems $u(\cdot) \mapsto y(\cdot)$.

Induced Norms of Linear Systems

A system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathcal{B})$ is linear if and only if $\mathcal{B}$ is a vector space. In other words, for all $w_1, w_2 \in \mathcal{B}$, for all scalars α_1, α_2, we have $\alpha_1 w_1 + \alpha_2 w_2 \in \mathcal{B}$. Specializing to input-output systems, $H \subset \mathcal{U} \times \mathcal{Y}$ is linear if and only if for all $(u_1, y_1), (u_2, y_2) \in H$, for all scalars α_1, α_2, we have that $(\alpha_1 u_1 + \alpha_2 u_2, \alpha_1 y_1 + \alpha_2 u_2) \in H$. For example, a linear input-output system H with zero initial condition (i.e., such that $u = 0 \implies y = 0$), $\mathcal{U} = (\mathbb{R}^p)^{\mathbb{R}}$ and $\mathcal{Y} = (\mathbb{R}^q)^{\mathbb{R}}$, is of the form

$$(Hu)(t) = \int_{-\infty}^{+\infty} h(t, \tau)u(\tau)d\tau,$$

for $h(\cdot, \cdot)$ a matrix-valued function from $\mathbb{T} \times \mathbb{T}$ to $\mathbb{R}^{q \times p}$. The function h is called the impulse response of H, since $(H\delta(\cdot - t_0))(t) = h(t, t_0)$. If moreover H is *time-invariant*, i.e., shifting the input in time results in an output simply shifted by the same amount, then we can in fact write

$$(Hu)(t) = \int_{-\infty}^{+\infty} h(t-\tau)u(\tau)d\tau = (h * u)(t),$$

for some function $h : \mathbb{R} \to \mathbb{R}^{q \times p}$.

For the set $\mathcal{L}(\mathcal{U}, \mathcal{Y})$ of linear operators between the normed vector spaces $\mathcal{U}$ and $\mathcal{Y}$ with norms $\| \cdot \|_{\mathcal{U}}$ and $\| \cdot \|_{\mathcal{Y}}$, recall that we define the induced norm on $\mathcal{L}(\mathcal{U}, \mathcal{Y})$ by

$$||H|| = \sup_{u \neq 0} \frac{\|Hu\|_{\mathcal{Y}}}{\|u\|_{\mathcal{U}}} = \sup_{\|u\|_{\mathcal{U}}=1} \frac{\|Hu\|_{\mathcal{Y}}}{\|u\|_{\mathcal{U}}}.$$

This provides a norm for linear time-invariant (LTI) systems, which can be computed explicitly for some L^p spaces for $\mathcal{U}$ and $\mathcal{Y}$. For example, consider H a single-input single-output (SISO) causal LTI system with impulse response h. Assume that $\|h\|_1 < \infty$. Then H maps L^∞ to L^∞ and its induced norm as a map of $\mathcal{L}(L^\infty, L^\infty)$ is $\|H\|_{\infty \to \infty} = \|h\|_1$. One part of this statement is easy, namely, since

$$|(Hu)(t)| \leq \int_0^t |h(t-\tau)u(\tau)|d\tau \leq \|u\|_\infty \int_0^t |h(t-\tau)|d\tau \leq \|h\|_1 \|u\|_\infty,$$

we have $\|H\|_{\infty \to \infty} \leq \|h_1\|$. One can show by choosing appropriate input signals that bound is approached arbitrarily closely. We can also consider H as a map in $\mathcal{L}(L^2, L^2)$. Note that since $\|h\|_1 < \infty$, its Fourier transform $\hat{h}$ is well-defined and one can show that the induced norm of H in this case is $\|H\|_{2 \to 2} = \sup_{\omega \in \mathbb{R}} |\hat{h}(j\omega)|$. The reader should refer to standard references on linear systems for more details and the corresponding multiple-input multiple-output (MIMO) results.

L^p Stability

The following definition introduces the important notions of stability and gain for input-output systems.

Definition 2.1.2. An input-output CT system defined by a relation $\mathcal{S} \subset L^{pe}_{n_u} \times L^{pe}_{n_y}$ is called L^p-stable if for all $(u, y) \in S$, $u \in L^p_{n_u} \implies y \in L^p_{n_y}$. It is called finite-gain L^p-stable if there exists $\gamma \geq 0$ such that

$$\inf_{T \geq 0} \left\{ \int_0^T \gamma^p |u(t)|^p - |y(t)|^p dt \right\} > -\infty, \forall (u, y) \in \mathcal{S}. \tag{2.4}$$

If $\mathcal{S}$ is finite-gain L^p-stable, its L^p-gain is the infimum of the set of γ satisfying 2.4. These definitions carry over to the ℓ^p-stability of DT systems, with the integral in (2.4) replaced by a discrete sum.

Remark 2.1.5. The condition (2.4) can be rewritten

$$\inf_{T \geq 0} \{\gamma^p \|u_T\|_p^p - \|y_T\|_p^p\} > -\infty, \forall (u, y) \in \mathcal{S},$$

valid for both the CT and DT case, or

$$\sup_{T \geq 0} \{\|y_T\|_p^p - \gamma^p \|u_T\|_p^p\} < +\infty, \forall (u, y) \in \mathcal{S}.$$

In other words, the set $\{\|y_T\|^p - \gamma^p \|u_T\|^p\}$ for $T \geq 0$ is bounded from above. Hence, for all $(u, y) \in S$, there exists a constant M_{uy} (which can depend on the pair (u, y), to capture transients due to different initial conditions for example) such that

$$\|y_T\|_p^p \leq \gamma^p \|u_T\|_p^p + M_{uy}, \forall T \geq 0, \tag{2.5}$$

fix

and the L^p-gain of $\mathcal{S}$ is the smallest γ satisfying these inequalities. Moreover, if $\mathcal{S}$ is causal and if the constants M_{uy} in condition (2.5) are in fact independent of (u, y), then this condition can be replaced by

$$\|y\|_p^p \leq \gamma^p \|u\|_p^p + M, \;\; \forall (u, y) \in S \text{ s.t. } u \in L^p \;\; (\text{not } L^{pe}!), \tag{2.6}$$

where the T subscripts are dropped. Indeed, (2.5) always implies (2.6). Conversely, if (2.6) holds, then it holds for functions of the form $P_T u_1$ and we can use $P_T H P_T = P_T H$ and the facts below Example 2.1.2 to conclude.

Remark 2.1.6. By considering $u \in L^p$ and taking $T \to \infty$ in (2.5), we see that if a system is finite-gain L^p-stable, then it is L^p-stable.

Example 2.1.4. For H an LTI system, all finite-gain L^p-stability conditions for different values of p are equivalent, i.e., if H is finite-gain L^p-stable for some $p \in [1, \infty]$, then it is finite gain L^p stable for all $p \in [1, \infty]$. In fact, taking H SISO for simplicity, it is L^p-stable if and only if its impulse response h is integrable, i.e., $\|h\|_1 < \infty$. The L^∞-gain is equal to $\|h\|_1 < \infty$, and L^∞-stability corresponds to the standard notion of bounded-input bounded-output (BIBO) stability. The L^1-gain is in fact also equal to $\|h\|_1$, and the L^p-gain is bounded by $\|h\|_1$ for $1 < p < \infty$. In particular, $\|h\|_1 < \infty$ also ensures that the Fourier transform $\hat{h}$ of h is well defined, and the L^2-gain of H is $\sup_{\omega \in \mathbb{R}} |\hat{h}(j\omega)|$.

In some cases, it is useful to define "local" versions of L^p-stability, e.g., by restricting the size of the input signals. Here is one possible way of doing it, bounding the instantaneous values of the input.

Definition 2.1.3. A system $\mathcal{S} \subset L^{pe} \times L^{pe}$ is called small signal L^p-stable if there exists $r > 0$ such that for all $(u, y) \in S$, if $u \in L^p_{n_u} \cap L^\infty_{n_u}$ and $\|u\|_\infty < r$, then $y \in L^p_{n_y}$. It is called small signal finite-gain L^p-stable if there exists $\gamma \geq 0$ and $r > 0$ such that for all $(u, y) \in \mathcal{S}$, there exists $M \geq 0$ such that for all $T \geq 0$, it $\|u_T\|_\infty \leq r$ then $\|y_T\|_p^p \leq \gamma^p \|u_T\|_p^p + M$.

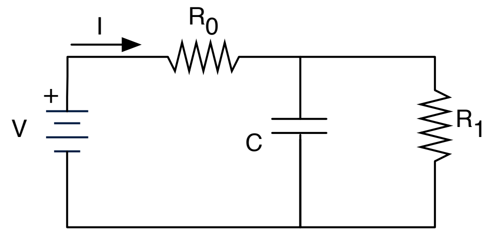


Figure 2.2: An electrical circuit.

Incremental L^p stability

Definition 2.1.4. A system $S \subset L^{pe} \times L^{pe}$ is said to be *incrementally finite-gain L^p-stable* if there exists $\gamma > 0$ such that for all $T > 0$ and for all (u, y_u), $(v, y_v) \in S$,

$$\|(y_u)_T - (y_v)_T\|_p \le \gamma \|u_T - v_T\|_p. \tag{2.7}$$

In this case, the incremental gain of H is the infimum over all γ satisfying (2.7).

Note that an incrementally finite-gain L^p-stable system for which on output y_0 to the input $u \equiv 0$ is in L^p is also finite-gain L^p-stable, since

$$\|y_T\|_p \le \gamma \|v\|_p + \|y_0\|_p,$$

by the triangle inequality. The two notions of L^p-stability and incremental L^p-stability coincide for linear systems, but not for nonlinear systems in general.

2.2 State Space Model Realizations of Input-Output Systems

Input-output models are black-box models. They relate input and output signals independently of any description of the internal mechanisms producing these signals. As a result, they are of fundamental importance to reason about connected and large-scale systems, since the size and complexity of an overall model grows slowly with the number of components in this framework. However, to carry actual system-related computations, it is often necessary to describe a component's internal mechanisms by means of state space models. First, recall the following more-or-less rigorous definition of state.

Definition 2.2.1. The *state* of a system at time t is the information required at time t so that the output for all $t' \ge t$ is determined from this information and the knowledge of the inputs for $t' \ge t$.

State space models can be obtained directly by reasoning from first principles, or by fitting parameters via black-box system identification procedures, which analyze output signals obtained from test input signals. In the later case, the approximate nature of the models is clear, since often no attempt is made to justify the chosen structure of a state space model beyond the fact that it exhibits reasonable predictive power. Most state space models considered in these notes however rely on first principle reasoning. Models of software and digital circuits can be quite precise, even though in this domain abstractions exist as well (e.g., when assuming that digital signals switch instantaneously). But for most physical, analog systems, first principle reasoning typically leaves certain behaviors unmodeled (e.g., high-order dynamics) in order to obtain reasonably tractable models. It is important to make sure that these unmodeled components do not invalidate a system design, either by employing robust design methods or by using a posteriori testing procedures.

Consider for example the electrical circuit shown on Fig. 2.2, where we are interested in the relationship between I as an input signal and V as an output signal [2]. If we did not know the composition of the circuit, we could try various test signals I and observe the behavior of the signal V, in order to fit the parameters of a model. If we assume the circuit to be linear, we could then try to identify the parameters of a Thevenin or

[2]This choice of input and output signals is arbitrary.

Norton equivalent of that circuit. On the other hand, if we have access to the composition of this circuit and know the parameters of its various components, we can directly model the relationships between signals as

$$\frac{dV_C}{dt} = -\frac{1}{R_1 C} V_C + \frac{1}{C} I \tag{2.8}$$

$$V = R_0 I + V_C, \tag{2.9}$$

where V_C is the voltage across the capacitor. It is likely that the physical components do not behave exactly as predicted by these equations and that the knowledge of the values of the capacitances and resistances is not perfect (resistance values are usually associated with a tolerance for example), but for many purposes such a model is precise enough. Here the variable V_C is an internal state variable that is used to explain the input-output relation between I and V.

2.2.1 Linear Time-Invariant State Space Models

Continuous-Time Systems

Generalizing (2.8), (2.9), a CT LTI state space model defines a system with an m-dimensional input u and a p-dimensional output y related by

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{2.10}$$

$$y(t) = Cx(t) + Du(t), \tag{2.11}$$

for $t \in \mathbb{T} = \mathbb{R}_+$ or $\mathbb{R}$ and $x(0) = x_0 \in \mathbb{R}^n$, where A, B, C, D are real matrices of appropriate dimensions. Figure 2.3 gives a block diagram representation of this model. In general, solutions of differential equations $\dot{x} = f(x,t)$ are understood in the sense of Caratheodory, i.e., as absolutely continuous functions satisfying the integral equation

$$x(t) = x(0) + \int_0^t f(x(\tau), \tau) d\tau, \tag{2.12}$$

where $x(0) \in \mathbb{R}^n$ is an "initial condition" parameter. This is necessary for example when the input u in (2.10) is not continuous, so that the notion of derivative of $\dot{x}$ is not well defined at the points of discontinuity. Solutions of (2.12) satisfy the differential equation almost everywhere. A common notation for the system $\mathcal{G}$ defined by (2.10), (2.11) is

$$\mathcal{G} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right].$$

The *transfer matrix* of this linear state space model is

$$G(s) = D + C(sI - A)^{-1}B,$$

defined for all complex numbers s, except possibly at some eigenvalues of A. In addition,

$$G(\infty) := \lim_{|s| \to \infty} G(s) = D.$$

Discrete-Time Systems

DT LTI state space models are defined analogously for $\mathbb{T} \subset \mathbb{Z}$, replacing differential equations by difference equations, to get

$$x[k+1] = Ax[k] + Bu[k], \tag{2.13}$$

$$y[k] = Cx[k] + Du[k]. \tag{2.14}$$

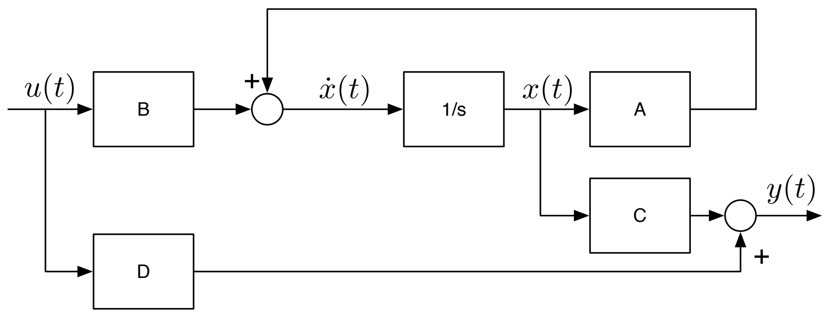


Figure 2.3: An LTI state space model.

2.2.2 Nonlinear Continuous State Space Models

Physical systems exhibiting certain types of behaviors (e.g., multiple equilibrium points, sustained oscillations, chaos, bifurcations, etc.) cannot be modeled by linear systems. In this case, we might want to use a nonlinear state space model, perhaps time-varying, e.g., of the continuous-time form

$$\dot{x} = f(x, u, t) \tag{2.15}$$
$$y = h(x, u, t), \tag{2.16}$$

with $x(0)$ given. Additional assumptions are made on f to obtain reasonable notions of solutions. Similarly, in discrete-time we can model a nonlinear system using the difference equation

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, k), \quad x_0 \text{ given},\\ y_k &= g(x_k, u_k, k). \end{aligned}$$

In general, nonlinear systems can exhibit extremely complex behaviors. In fact, very simple one dimensional discrete recurrences can exhibit chaotic behavior, for example the logistic map

$$x_{k+1} = r x_k (1 - x_k),$$

for certain values of r. As a result, one must generally start by identifying special classes of interesting nonlinear systems relevant for a problem of interest and amenable to analysis, and develop analysis and synthesis tools dedicated to these special cases. Moreover, an often adequate solution from the computational and robustness point of view is to try to approximate a nonlinear system by a linear time-invariant one plus some perturbation term that is only approximately characterized. This is the basic principle of robust control methods.

2.2.3 Nondeterminism

The previous linear and nonlinear models produce essentially a unique output once an initial condition and an input are given. More realistic models attempt to directly include some notion of perturbation, either in a stochastic or a deterministic framework. In such models, dynamics can be perturbed, and output measurements can be noisy. One way of capturing uncertainty is by using differential inclusions rather than differential equations as in (2.15), (2.16). More precise information can be given in the form of probabilistic transition systems and output maps. See also definition 2.2.2. For example, we can have a DT model

$$x_{k+1} = f(x_k, u_k, w_k, k), \quad x_0 \text{ given}, \tag{2.17}$$
$$y_k = g(x_k, u_k, v_k, k), \tag{2.18}$$

where w_k and v_k are uncertain vectors perturbing the dynamics and observations respectively.

2.2.4 Transition Systems and Automata

In computer science, system models (for software, hardware and physical systems) are often expressed in the form of *transition systems*. Many variations of such models exist, depending on the problem of interest, and the terminology also tends to differ slightly depending on the application area. In particular, transition systems are sometimes also called *automata*, but many authors reserve the latter term for certain specific types of transition systems (e.g., finite number of states and transitions, resulting in a finite-state machine, presence of initial and final or marked states). From the computational point of view, the situation is roughly as follows: in engineering (in particular control engineering here), LTI models are preferred because many things about them can be computed using linear algebra (and more generally convex optimization methods such as semi-definite programming). Computer science tends to focus on finite finite automata for which many algorithms are also available, e.g., based on the graph representation of the automata or more advanced data structures such as Binary Decision Diagrams (BDDs).

The theoretical investigations of the properties of automata, e.g., related to the language generated by an automaton, to reachability, safety and liveness properties, or various notions of compositions, have focused on somewhat different issues than classical control theory, and these ideas are useful in the study of NECS. Moreover, differential and difference equations have traditionally been used in control engineering to model *time-driven processes*. In the analysis of complex systems involving say human operators and computers, it is important to also be able to efficiently model asynchronous occurrences of discrete events, resulting in Discrete-Event Systems [CL08] and *event-driven processes*, or hybrid systems combining time-driven and event-driven dynamics. Finally, automata-based models are convenient to study general finite state models, whereas the models presented so far are mostly used for continuous state spaces. Note however that difference equation models are useful to study certain discrete models with special structure as well, e.g., queueing systems and even general types of finite state Hidden Markov Models [EAM94].

We start with the following extremely general notion of transition system, a constrain it later as computational aspects become important.

Definition 2.2.2. A (Moore) transition system TS is a tuple $(X, X_0, U, \longrightarrow, Y, L)$ consisting of

- a set of states X (not necessarily finite or even countable)
- a set of initial states $X_0 \subseteq X$
- a set of inputs U (not necessarily finite or even countable)
- a transition relation $\longrightarrow \subseteq X \times U \times X$
- a set of outputs Y
- a set-valued output map $L : X \to 2^Y$.

The transition system is called *finite* if X and Y are finite. The system starts in one of the initial states (non-determinism is introduced if X_0 consists of more than one state), and evolves according to the transition relation. Namely, a transition $(x, u, x') \in\longrightarrow$, denoted in the following $x \xrightarrow{u} x'$, means that the system initially in state x is subject to an input or event u and moves to the state x'. When the transition system is finite, we can represent it as a graph with nodes associated to states and edges associated to triplets of the transition relation and labeled by the input value. It is sometimes useful to have a special input label, denoted τ, with $U = U' \cup \{\tau\}$ and $\tau \notin U'$, to denote transitions occurring in the absence of an observable input, called silent transitions (these transitions are similar to ϵ-transitions in the standard terminology for nondeterministic automata). The set U' then denotes the observable inputs. Note that we allow non-determinism in the dynamics, since there can be several states $x' \in X$ associated to a given pair (x, u) by the transition relation. In the graph representation, this means that we allow several edges labeled with the same input value u starting from a node x. Nondeterministic choices in particular are useful to model the parallel execution of independent activities, unknown or unpredictable environments (e.g., a human user), as well as for abstraction purposes and to leave certain aspects of the system unspecified, e.g., in early design phases to leave several options for the possible final behaviors.

The set-valued map L means that in state x, there is a *set* of possible observable outputs $L(x)$, with possible non-determinism introduced to model sensor uncertainty for example. If L is a single-valued map (deterministic observations), with output y in state x, then we use the standard notation $L(x) = y$ instead of $L(x) = \{y\}$. The case of full state observation corresponds to $L(x) = x$, i.e., L is the identity map. Allowing non-determinism in the output map is done for convenience, but is not strictly necessary. It produces models similar to (2.17), (2.18) and to the way Hidden Markov Models (HMM) are formulated in the probabilistic setting for example. However, we could have pushed all the non-determinism in the transition relation, so that starting from a transition $x \xrightarrow{u} x'$ with say $L(x') = \{y_1, y_2\}$, we split the transition into two transitions $x \xrightarrow{u} x'_1$ and $x \xrightarrow{u} x'_1$ with $L(x'_1) = y_1$ and $L(x'_2) = y_2$.

Remark 2.2.1. The name Moore transition system is not really standard, and is used here by analogy with the definition in the finite state case of Moore automaton, which is just a finite state automaton with outputs associated to states. With input-output transition systems in general, there is an alternative way of introducing outputs, namely by associating outputs to transitions rather than states. Both conventions are used (including, e.g., in the theory of controlled Markov chains). Then the transition relation becomes a relation on $X \times U \times Y \times X$, denoted $x \xrightarrow{u/y} x'$, and in the graphical representation the edges are now labeled by input-output pairs. The corresponding automata are usually called *Mealy automata*. It is not hard to see that the two types of representations are equivalent.

Exercise 1. Show that Moore and Mealy transition systems are equivalent ways of describing input-output behaviors, and in fact both can be represented as automata with trivial output set, i.e., with Y a singleton. Again, this can be exploited to develop general computational tools but is not necessarily convenient for modeling purposes. Hint: starting say with a Moore transition system, convert it to an equivalent Mealy transition system, and then reinterpret $U \times Y$ as the new input set.

Note again the generality of the definition 2.2.2. For example, one can represent a system described by a differential inclusion or a controlled differential equation using such a transition system. Consider a time-invariant control system obeying a given controlled differential equation. Define the set of inputs as the set of pairs $(T, u|_{[0,T]})$, where $u|_{[0,T]})$ denotes the restriction of an admissible input signal to the interval $[0, T]$. Then two states x and x' are related by a transition if there exists $T \geq 0$ and an input signal $u|_{[0,T]}$ driving the state from x to x'. Note here that we have an uncountable number of states and generally an uncountable number of transitions out of every state. By fixing T in the input set to a fixed value, one also obtains a discrete-time, regularly sampled version of the continuous-time system. Of course, by using this device we are forgetting the nice algebraic and computational properties potentially associated with the differential equation, hence no progress has been made unless we can simplify again the resulting transition system, say to a computationally tractable finite system. This idea is explored later when we discuss notions of equivalence and approximation of transition systems, in particular simulation and bisimulation relations.

Also, keep in mind that the output set Y in definition 2.2.2 can be quite general. In particular, it can consist of atomic logic propositions that are true for the associated states. L is then called a *labeling function*. Then, given Φ a propositional logic formula, the state x verifies Φ if the evaluation of $L(x)$ makes the formula true (denoted $L(x) \models \Phi$). This view can be exploited to specify and verify properties along the possible trajectories of a system modeled as a transition system, an idea discussed in a later chapter and known as *model-checking* [BK08].

Remark 2.2.2. Another widely used term is *Kripke structure*. Normally, a Kripke structure is a finite state Moore transition system with a single label in the input vocabulary (i.e., the input labels are irrelevant) and with L a labeling function (i.e., the outputs are logic propositions).

We finish this section on transition systems with a few general definitions. For $x \in X$ and $u \in U$, define the sets of direct u-successors and direct successors of x as

$$Post(x, u) = \{x' \in X | x \xrightarrow{u} x'\}, \quad Post(x) = \bigcup_{u \in U} Post(x, u).$$

Similarly, the sets of direct u-predecessors and direct predecessors of x are defined as

$$Pre(x, u) = \{s' \in X | x' \xrightarrow{u} x\}, \quad Pre(x) = \bigcup_{u \in U} Pre(x, u).$$

The notation is extended to sets in a straightforward way. For $S \subset X$,

$$Post(S, u) = \bigcup_{x \in S} Post(x, u), \quad Post(S) = \bigcup_{x \in S} Post(x),$$

$$Pre(S, u) = \bigcup_{x \in S} Pre(x, u), \quad Pre(S) = \bigcup_{x \in S} Pre(x).$$

A state x of a transition system is called *terminal* is it does not have any outgoing transition, i.e., if $Post(x) = \emptyset$. For reactive systems, the types of systems we are interested in for control applications and are supposed to run continuously, this is generally something that must be detected and avoided. In such a state, the system is said to be in *deadlock*. Finally, we have the following notions of determinism.

Definition 2.2.3. A transition system is called *input-deterministic* if $L|_{X_0}$ is injective, and $|Post(x, u)| \leq 1$ for all states $x \in X$ and inputs $u \in U$. Hence observing the initial output and all the subsequent inputs allows us to reconstruct the state trajectory with certainty. It is called *output-deterministic* if $L|_{X_0}$ is injective and $|Post(x) \bigcap \{x' \in X | L(x') = A\}| \leq 1$ for all states $x \in X$ and output sets $A \subset Y$. In other words, for any state x and inputs u', u'' with $x \xrightarrow{u'} x'$ and $x \xrightarrow{u''} x''$ and the same observations $L(x') = L(x'')$, we must have $x' = x''$. Hence if L is a single valued map, observing the sequence of outputs allows us to reconstruct the state trajectory with certainty. Similarly if L is a labeling function and we can observe in each state all the propositions that are true.

2.2.5 Timed and Hybrid Automata

For all practical purposes, the transition systems of definition 2.2.2 cannot be manipulated at this level of generality. Indeed, in general, as modeling expressiveness increases, so do the associated computational costs. Because the definition places no restriction on the state space or the transitions, essentially any type of system can be modelled in this framework, but one cannot develop computational tools to answer basic questions about or even just store such general models. Finite transition systems can be manipulated relatively easily on a computer, but are often insufficient for control purposes. In particular, it is not immediately clear how to record the precise timing of asynchronous events using state transition systems with a finite or even countable state space. When used directly for modeling purposes, finite state transition systems usually only record the ordering of events, in *untimed models* or perhaps regularly sampled models with fixed sampling period. Note however that such models are sufficient for some purposes, e.g., reachability analysis. Moreover it turns out that certain types of hybrid transition systems with state variables evolving both in continuous-time and via instantaneous discrete updates (events) are for various purposes equivalent to finite transition systems, a topic that we revisit later when we discuss simulation relations between automata. In this case, the original hybrid system formalism is used for modeling purposes, and assuming that certain restrictions are enforced on the type of admissible models, the model can sometimes be automatically transformed into a (usually very large) finite state system, adapted to computer-based analysis.

A well-known class of hybrid models is the class of *timed automata* introduced by Alur and Dill [AD94]. Other models of timed automata exist, see, e.g., the references in [CL08, chapter 5]. Timed automata are finite transition systems with additional state variables called *clocks*, which are used to control the timing of the transitions. With such models, questions such as transition before a certain deadline, the number of events in a given time interval or the time spent in a particular state can be considered. Clearly, such questions are of critical importance in real-time systems such as control systems. Note that for synchronous discrete-time systems, standard discrete transition systems are sufficient. The advantage of timed automata is the possibility to model continuous-time systems in an efficient way, i.e., without employing say a very fine time discretization that would lead to a very large transition system. These models are also particularly useful for asynchronous systems.

Clock variables in timed automata are simply state variables verifying the differential equation $\dot{x} = 1$, with the additional possibility of *resetting* these variables during transitions between discrete states. These discrete states are also called *modes*. *Hybrid automata* greatly generalize timed automata by allowing general differential and algebraic equations in each mode, together with instantaneous mode switches as in timed automata. Hybrid automata are extremely expressive models but as a result many computational issues arise, such as the undecidability of certain reachability questions, the possibility of generating non-physical behaviors such as Zeno behaviors, etc. Hence hybrid automata should be used with care, because it is not always clear what is gained by describing a system as a hybrid automaton if computational issues are not discussed. In other words, trivializing the modeling problem generally comes at a high computational price. Still, hybrid automata form a popular class of hybrid system models and we give a formal definition below. The various elements of the model are often further constrained to try to improve the computational properties.

Definition 2.2.4. A hybrid automaton is a tuple

$$(Q, X, E, U, Y, f, \phi, \text{Inv}, \text{Guard}, \rho, q_0, X_0),$$

where

- Q is a set of discrete states, also called modes or locations
- X is a continous state space, e.g., $\mathbb{R}^n$
- E is a finite set of input events
- U is a set of admissible continuous controls
- f is a vector field for each mode: $f : Q \times X \times U \to X$
- ϕ is a discrete state transition function $\phi : Q \times X \times E \to Q$
- Inv is a set defining an invariant condition for each mode, $\text{Inv} \subseteq Q \times X$
- Guard is a set defining a guard condition for each transition $\text{Guard} \subseteq Q \times Q \times X$
- ρ is a reset function for each transition and input event: $\rho : Q \times Q \times X \times E \to X$
- $L : Q \times X \to 2^Y$ is an output or labeling function
- q_0 is an initial discrete state
- x_0 is an initial continuous state.

Guards are condition under which a transition is allowed. Invariants are conditions that must be satisfied by the continuous state is a particular mode. Violation of this condition forces the state to take one of the transitions allowed by the guard conditions. Of course, certain relationships must be enforced between guards and invariants in order to have a well-posed model. Finally, we can have additional sources of non-determinism in the hybrid automaton model, for example by defining discrete transitions as $\phi : Q \times X \times E \to 2^Q$, reset functions as $\rho : Q \times Q \times X \times E \to 2^X$, and allowing sets of initial conditions. Note that it is straightforward to *unfold* a hybrid automaton and view it as a transition system as in definition 2.2.2, at least as long as one allows uncountably many states and transitions.

For timed automata, in addition to the restrictions on the vector fields mentioned above, there are restrictions on the form of the guard and invariant conditions. Namely, these conditions must be of the form

$$\text{cond} ::= x < c \,\Big|\, x \leq c \,\Big|\, x > c \,\Big|\, x \geq c \,\Big|\, \text{cond} \wedge \text{cond}.$$

Moreover, the reset functions are only allowed to set clock variables back to 0.

Computational Tools

Two popular computer software tools for the manipulation and model-checking of timed automata are UPPAAL [UPP] and KRONOS [Yov97]. Various software tools for the manipulation and verification of properties of hybrid systems also exist.

2.3 Composing Systems

2.3.1 Composing Input-Output Systems

Composing input-output systems is straightforward, and the methods available in this framework tend to be compositional as well, in the sense that

- the complexity of the system grows modestly when we compose subsystems. For example, composing two systems with n inputs and n outputs in series produces a new system with n inputs and outputs. The complexity of characterizing the relationships between the new global inputs and outputs often grows reasonably with the number of subsystems (e.g., linearly). For example, in the previous series connection, if the two subsystems have gain bounded by γ_1 and γ_2 then we immediately know that the connected system has gain bounded by $\gamma_1\gamma_2$.
- input-output analysis methods tend to focus on properties that are preserved under composition, e.g., passivity.

Note however that input-output connections typically rely on an idealization which consists in assuming that the behavior of a component is not modified when its output signals are used as inputs of another component. In practice, a component's behavior does depend on the load it is driving. However, engineers typically try to limit this phenomenon as much as possible, precisely to allow modular designs. For example, electronic systems have high input impedance and low output impedance for this purpose, and buffer circuits are available.

Review series, parallel and feedback connections, including the corresponding operations on the transfer functions for LTI systems. We will discuss in the next chapter how to connect continuous-time and discrete-time systems.

2.3.2 Composing State-Space and Transition Systems

The behavior of state-space models with respect to composition is much worse than for input-output models, with state space sizes that tend to grow exponentially with the number of subsystems. Note that for continuous state-space systems, the dimensions of the subsystems add. But discretizing the state space $X \subset \mathbb{R}^d$ of a continuous system for computational purposes leads to a number of state that grows exponentially with the dimension d.

We consider the following standard notion of composition of transition systems [BK08], called composition by handshaking, by synchronous message passing, or simply parallel composition. Note that other useful types of compositions exist.

Definition 2.3.1. Let $TS_i = (X_i, X_{0,i}, U_i, \longrightarrow_i, Y_i, L_i), i = 1, 2$ be two transition systems and $H \subset U_1 \cap U_2$. The transition system $TS_1||_H TS_2$ is defined as

$$TS_1||_H TS_2 = (X_1 \times X_2, U_1 \cup U_2, \rightarrow, X_{0,1} \times X_{0,2}, Y_1 \cup Y_2, L),$$

where $L((x_1, x_2)) = L_1(x_1) \cup L_2(x_2)$ and where the transition relation $\rightarrow$ is defined by the rules

- interleaving for $u \notin H$:

$$\frac{x_1 \xrightarrow{u} x_1'}{(x_1, x_2) \xrightarrow{u} (x_1', x_2)} \qquad \frac{x_2 \xrightarrow{u} x_2'}{(x_1, x_2) \xrightarrow{u} (x_1, x_2')}$$

- handshaking for $u \in H$:

$$\frac{x_1 \xrightarrow{u} x_1' \qquad x_2 \xrightarrow{u} x_2'}{(x_1, x_2) \xrightarrow{u} (x_1', x_2')}$$

In other words, in the system $TS_1||_H TS_2$, the transition systems synchronize on the inputs in H. We abbreviate the notation to $TS_1||TS_2$ if $H = U_1 \cap U_2$. The operation $||_H$ is obviously commutative, but also associative. Hence the composition of more than 2 transition systems $TS_1||_H TS_2||_H \cdots ||_H TS_n$ is defined immediately, for $H \subseteq U_1 \cap \cdots \cap U_n$. Note however that associativity does *not* hold if H is not the same in the different operations, i.e.

$$TS_1||_H(TS_2||_{H'}TS_3) \neq (TS_1||_H TS_2)||_{H'}TS_3$$

in general for $H \neq H'$.

Remark 2.3.1. Recall also that for automata the product construction has implications in terms of basic language operations, in particular for language intersection and union (the two differ only by the choice of marked or final states in the product automaton).