EE291E Hybrid Systems and Intelligent Control
Lecture Notes 2
Mathematical Background: Discrete and Continuous Systems

Claire J. Tomlin

January 17, 2021

The formal definitions of hybrid systems build on a number of concepts from continuous state and discrete state dynamical systems.

A **continuous state dynamical system** in one in which the state x takes on values in $\mathbb{R}^n$, that is $x \in \mathbb{R}^n$. Continuous state systems may evolve in either continuous time (for example, $\dot{x} = f(x, u)$), or discrete time (for example, $x_{k+1} = f(x_k, u_k)$).

A **discrete state dynamical system** is one in which the state q takes on values in a finite set $\{q_1, q_2, q_3, \ldots\}$.

1 Finite Automaton models of discrete state systems

- **Examples:** digital switching circuit, lexical analyzers, digital computer ...
- **Definition [1, 2]:** A *finite automaton* M is a mathematical model of a system represented as

$$(Q, \Sigma, \text{Init}, R) \tag{1}$$

where

  - Q is a finite set of *discrete state variables*
  - $\Sigma = \Sigma_1 \cup \Sigma_2$ is a finite set of *discrete input variables*, where Σ_1 contains the controller's inputs and Σ_2 contains the environment's inputs, which cannot be controlled
  - $\text{Init} \subseteq Q$ is a set of *initial states*
  - $R : Q \times \Sigma \to 2^Q$ is a *transition relation* which maps the state and input space to subsets of the state space and thus describes the transition logic of the finite automaton

- **Definition:** An *execution* of the finite automaton M is defined to be a finite or infinite sequence of states and inputs, written as:

$$(q(\cdot), \sigma(\cdot)) \in (Q \times \Sigma)^\omega \tag{2}$$

where $(\cdot)^\omega$ indicates infinite strings, and for $i \in \mathbb{Z}$:

$$q(0) \in \text{Init and } q(i+1) \in R(q(i), \sigma(i)) \tag{3}$$

- Σ^* is the set of strings of arbitrary length of elements of Σ (we denote by ϵ a string of length 0)
- **Definition:** Consider a finite automaton M with a specified set of final states $F \subseteq Q$. M is said to *accept* a string $s = \sigma(0)\sigma(1)\sigma(2)\ldots\sigma(n) \in \Sigma^*$ if there exists a sequence of states $q = q(0)q(1)q(2)\ldots q(n+1) \in Q^*$ such that:
  - $q(0) \in \text{Init}$

- $q(i+1) \in R(q(i), \sigma(i))$
- $q(n+1) \in F$

- **Definition:** The *language accepted by* M, $L(M)$, is the set of all strings accepted by M
- **Definition:** Two automata, M_1 and M_2, are said to be *equivalent* if $L(M_1) = L(M_2)$
- **Definition:** The finite automaton M may *block* certain input strings if it does not accept certain inputs at certain states
- **Definition:** The finite automaton M may be *non-deterministic*, meaning that it could take different transitions from the same state in response to the same input symbol
- **Remark:** If the transition relation R is a function, that is $|R(q, \sigma)| = 1$ for all $q \in Q$ and all $\sigma \in \Sigma$, then M is deterministic
- **Example:** Consider the finite automaton M in Figure 1. In this case,

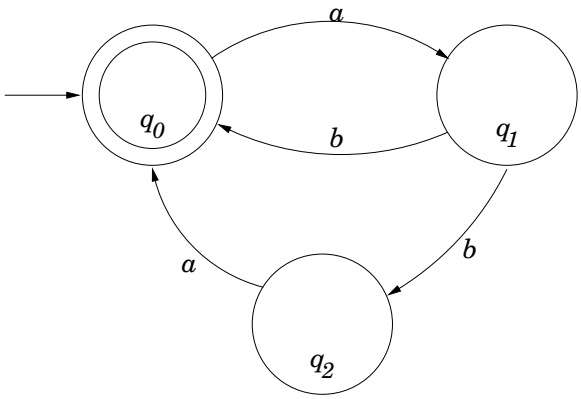

Figure 1: Finite Automaton Example.

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{a, b\}$
- $q_1 = R(q_0, a), q_2 = R(q_1, b), q_0 = R(q_2, a), q_0 = R(q_1, b)$ (note that R is not a function)
- Init $= q_0$ (indicated by the straight arrow in Figure 1)
- $F = q_0$ (indicated by a double circle in Figure 1)
- $L(M) = \{\epsilon, ab, aba, abab, abaaba, \ldots\} = ((ab)^*(aba)^*)^* \subseteq \Sigma^*$
- non-deterministic

- **Example [1]: Coordinating Finite Automata.** Consider the crossroad traffic controller problem as illustrated in Figure 2. The system consists of three components: Avenue A, Boulevard B, and traffic Controller C. We model each of these components

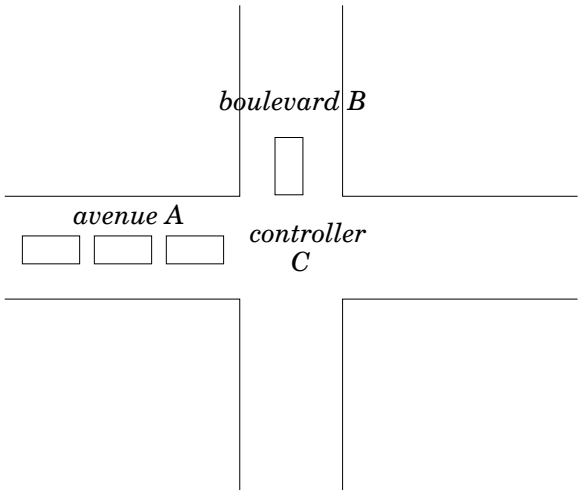

Figure 2: Crossroad Traffic Controller.

as a finite state machine as shown in Figure 3. Each of A, B, and C has 2 states, as shown in Figure 3.

The crossroad system is modeled by the *synchronous composition* of the 3 state machines, written as $M = A \otimes B \otimes C$:

- $Q = \{(stop, stop, go_A), \ldots, (go, go, go_B)\}$ (8 states)
- $\Sigma = \{a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3\}$
- The transition relation for M is given in term of Boolean conjunctions of the respective transition relations for the components, for example: $(go, stop, go_A) = R((stop, stop, go_A), a_2 \wedge (b_1 \vee b_2) \wedge c_1)$
- Init $= \{(stop, stop, go_A), (stop, stop, go_B)\}$
- $L(M)$ is the set of finite sequences of Boolean conjunctions and disjunctions of input symbols

Why is this modeling formalism interesting to us?

Because if we accept this model as reality, and we can prove (verify) that the system never transitions into a state in which $(A : cars_{going}) \wedge (B : cars_{going})$, then the system is *safe*. Likewise, if we can verify that system allows all cars on each road to eventually get through the intersection, then the system is *live*.

Consider the safety problem. It is possible to verify the safety of the system by constructing a *monitor automaton* as shown in Figure 4:

T_1 has the initial state *safe*; the self-loop transition *safe* $\rightarrow$ *safe* marked by a "+" is a *recur* edge, meaning that any infinite path through the transition structure of T_1 which crosses edge *safe* $\rightarrow$ *safe* infinitely often designates a behavior accepted by T_1. The *else* indicates that every behavior (of M) is accepted by T_1 unless it involves $(A : cars_{going}) \wedge (B : cars_{going})$, in which case T_1 moves to state *unsafe*, which disallows

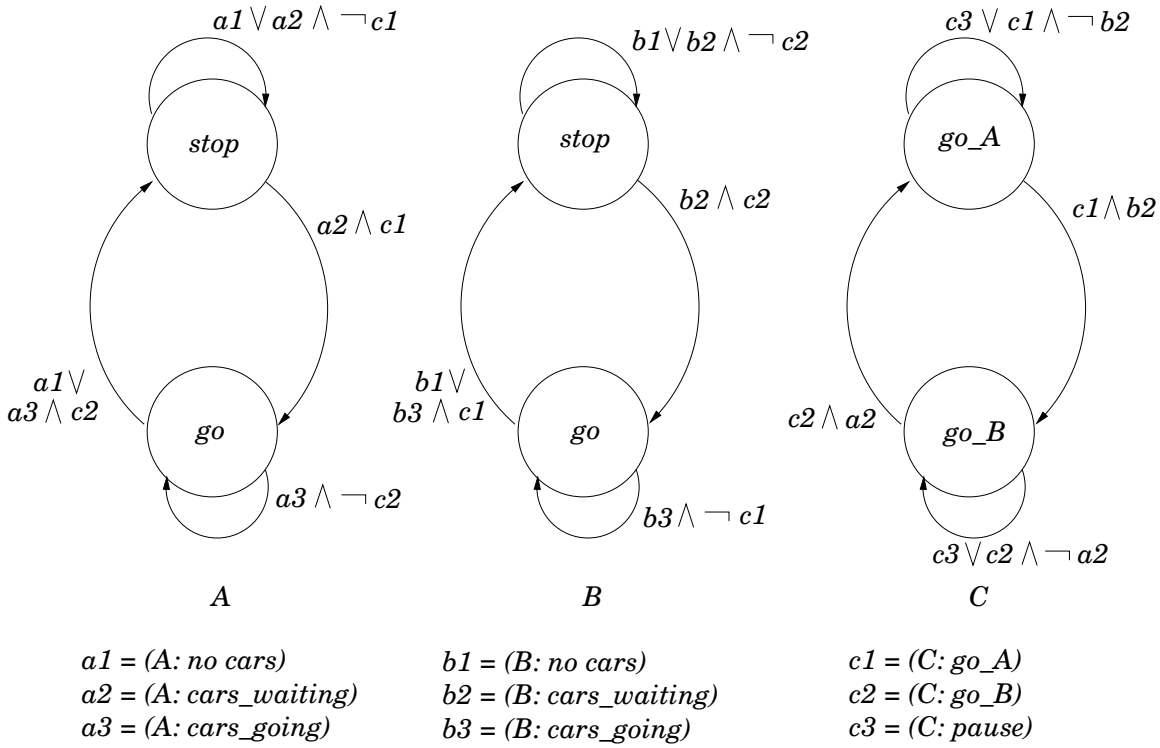

Figure 3: Finite automata models of the crossroad traffic controller example.

an infinite number of recur-edge crossings, and thus disallows the acceptance by T_1 of any behavior which includes that collision event. More formally, the system is safe if

$$L(M) \subset L(T_1) \tag{4}$$

2 Differential Equations

- **Definition:** *Continuous state, continuous time control systems* may be represented as differential equations evolving on a state space X [3]:

$$\dot{x}(t) = f(x(t), v) \tag{5}$$

where

  - $x \in X$ is the state, here $X = R^n$
  - $v \in V = U \times D$ is the space of continuous input variables, where $U = \mathbb{R}^u$ is the set of control inputs and $D = \mathbb{R}^d$ is the set of disturbance inputs
  - $f : \mathbb{R}^n \times U \times D \to \mathbb{R}^n$ is a vector field, assumed to be Lipschitz continuous in x and piecewise continuous in v
  - the initial state $x(0) \in$ Init where Init $\subseteq X$

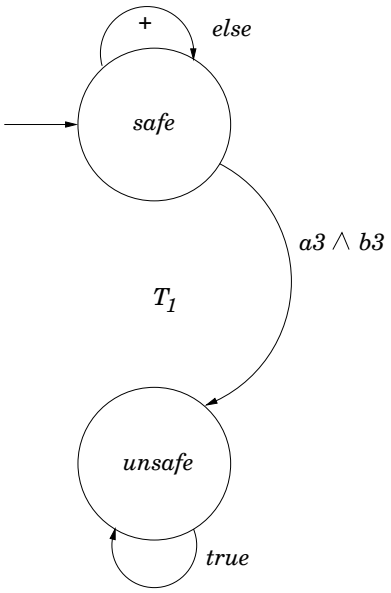

Figure 4: T_1: The monitor automaton which defines the state-invariant task "no cars collide".

Definition 1 (Trajectory) *A* trajectory *(execution, solution in the sense of Caratheodory) of (5) over an interval* $[\tau, \tau'] \subseteq \mathbb{R}$ *is a map:* $(x(\cdot), v(\cdot)) : [\tau, \tau'] \to \mathbb{R}^n \times V$ *such that*

$$x(\tau') = x(\tau) + \int_{\tau}^{\tau'} f(x(s), v(s))ds \tag{6}$$

Definition 2 (Lipschitz Continuous) *The function* $f : \mathbb{R}^n \to \mathbb{R}^n$ *is said to be Lipschitz continuous if there exists a* $\lambda > 0$ *such that for all* $x_1, x_2 \in \mathbb{R}^n$*:*

$$||f(x_1) - f(x_2)|| < \lambda ||x_1 - x_2|| \tag{7}$$

If f is Lipschitz continuous in x, then it is continuous is x. The converse is not necessarily true; however, if f has bounded partial derivative in x, then it is Lipschitz continuous.

- **Theorem 1: Local Existence and Uniqueness.** Assume $f(x, v)$ is piecewise continuous in v and Lipschitz continuous in x, for $x \in \{x \in \mathbb{R}^n :\| x - x_0 \| \leq r, r > 0\}$ and for $||f(x_0)|| \leq h, h \geq 0$. Then there exists $\delta > 0$ such the system (5) has a unique solution (6) on $[0, \delta]$.

- **Theorem 2: Global Existence and Uniqueness.** Assume $f(x, v)$ is piecewise continuous in v and Lipschitz continuous in x, and that $\| f(x_0) \| < h$ for $h > 0$, $x \in \mathbb{R}^n$. Then the system (5) has a unique solution (6) $\forall t$.

- **Examples:** Consider systems which violate some of the conditions:

  - f discontinuous in x: $\dot{x} = -\text{sgn}(x)$. The solution starting at $x_0 = 0$ is undefined for all $t > 0$.

- f continuous but not Lipschitz in x: $\dot{x} = x^{1/3}$. Solution is not unique ($x(t) = (2t/3)^{3/2}, x(t) = 0$ are both valid solutions for $x_0 = 0$)
- f Lipschitz but not globally Lipschitz in x: $\dot{x} = -x^2$. The solution for $x_0 = -1$ is $x(t) = 1/(t-1)$ and is not defined when $t = 1$.

- **Theorem 3: Continuous dependence on initial conditions.** Assume $f(x, v)$ satisfies the conditions of Theorem 2, and let x and y be two solutions of (5) starting at x_0 and y_0 respectively. Then for all $\epsilon > 0$ there exists $\delta(\epsilon, T) > 0$ such that

$$\| x_0 - y_0 \| \leq \delta \Rightarrow \| x(t) - y(t) \| \leq \epsilon \tag{8}$$

  for all $t \in [0, T]$, T finite.

- **Remarks:** These theorems provide conditions for a continuous system to be *non-blocking* (solutions exist locally), *deterministic* (solutions are unique), *non-Zeno* (solutions can be extended over arbitrarily long horizons).

References

[1] R. P. Kurshan. *Computer Aided Verification of Coordinating Processes: The Automata Theoretic Approach*. Princeton University Press, Princeton, N. J., 1994.

[2] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Menlo Park, 1979.

[3] S. S. Sastry. *Nonlinear Systems*. Springer Verlag, 1999.