Synthesis for Cyber-Physical Systems

Paulo Tabuada

Cyber-Physical Systems Laboratory
Department of Electrical Engineering
University of California at Los Angeles

Organization

- Lecture 1: Introduction to Cyber-Physical Systems, models, and relationships
- Lecture 2: Synthesis using exact finite-state abstractions
- Lecture 3: **Synthesis using approximate finite-state abstractions**
- Lecture 4: Playtime with Pessoa (Matthias Rungger)

- In order to enlarge the class of systems admitting finite-state abstractions we need to relax the notions of simulation and bisimulation.

Approximate system relationships

Approximate simulation

- In order to enlarge the class of systems admitting finite-state abstractions we need to relax the notions of simulation and bisimulation.

Definition (Metric system)

A system S is said to be a metric system if the set of outputs Y is equipped with a metric $\mathbf{d} : Y \times Y \rightarrow \mathbb{R}_0^+$.

- Using a metric, we can generalize the notion of simulation to approximate simulation.

Approximate system relationships

Approximate simulation

Definition (Approximate Simulation Relation)

Consider two metric systems S_a and S_b with $Y_a = Y_b$, and let $\varepsilon \in \mathbb{R}_0^+$. A relation $R \subseteq X_a \times X_b$ is an ε-approximate simulation relation from S_a to S_b if the following three conditions are satisfied:

1. for every $x_{a0} \in X_{a0}$, there exists $x_{b0} \in X_{b0}$ with $(x_{a0}, x_{b0}) \in R$;
2. for every $(x_a, x_b) \in R$ we have $\mathbf{d}(H_a(x_a), H_b(x_b)) \leq \varepsilon$;
3. for every $(x_a, x_b) \in R$ we have that:

   $x_a \xrightarrow[a]{u_a} x'_a$ in S_a implies the existence of $x_b \xrightarrow[b]{u_b} x'_b$ in S_b satisfying $(x'_a, x'_b) \in R$.

We say that S_a is ε-approximately simulated by S_b or that S_b ε-approximately simulates S_a, denoted by $S_a \preceq_{\mathcal{S}}^{\varepsilon} S_b$, if there exists an ε-approximate simulation relation from S_a to S_b.

Definition (Approximate Simulation Relation)

Consider two metric systems S_a and S_b with $Y_a = Y_b$, and let $\varepsilon \in \mathbb{R}_0^+$. A relation $R \subseteq X_a \times X_b$ is an ε-approximate simulation relation from S_a to S_b if the following three conditions are satisfied:

1. for every $x_{a0} \in X_{a0}$, there exists $x_{b0} \in X_{b0}$ with $(x_{a0}, x_{b0}) \in R$;
2. for every $(x_a, x_b) \in R$ we have $\mathbf{d}(H_a(x_a), H_b(x_b)) \leq \varepsilon$;
3. for every $(x_a, x_b) \in R$ we have that:

   $x_a \xrightarrow[a]{u_a} x'_a$ in S_a implies the existence of $x_b \xrightarrow[b]{u_b} x'_b$ in S_b satisfying $(x'_a, x'_b) \in R$.

We say that S_a is ε-approximately simulated by S_b or that S_b ε-approximately simulates S_a, denoted by $S_a \preceq_{\mathcal{S}}^{\varepsilon} S_b$, if there exists an ε-approximate simulation relation from S_a to S_b.

- What happens when $\varepsilon = 0$?

Definition (Approximate Simulation Relation)

Consider two metric systems S_a and S_b with $Y_a = Y_b$, and let $\varepsilon \in \mathbb{R}_0^+$. A relation $R \subseteq X_a \times X_b$ is an ε-approximate simulation relation from S_a to S_b if the following three conditions are satisfied:

1. for every $x_{a0} \in X_{a0}$, there exists $x_{b0} \in X_{b0}$ with $(x_{a0}, x_{b0}) \in R$;
2. for every $(x_a, x_b) \in R$ we have $\mathbf{d}(H_a(x_a), H_b(x_b)) \leq \varepsilon$;
3. for every $(x_a, x_b) \in R$ we have that:

   $x_a \xrightarrow[a]{u_a} x_a'$ in S_a implies the existence of $x_b \xrightarrow[b]{u_b} x_b'$ in S_b satisfying $(x_a', x_b') \in R$.

We say that S_a is ε-approximately simulated by S_b or that S_b ε-approximately simulates S_a, denoted by $S_a \preceq_{\mathcal{S}}^{\varepsilon} S_b$, if there exists an ε-approximate simulation relation from S_a to S_b.

- What happens when $\varepsilon = 0$?
- The inequality $\mathbf{d}(H_a(x_a), H_b(x_b)) \leq \varepsilon$ implies $H_a(x_a) = H_b(x_b)$.

Definition (Approximate bisimulation)

Consider two metric systems S_a and S_b with $Y_a = Y_b$, and let $\varepsilon \in \mathbb{R}_0^+$. We say that system S_a is ε-approximately bisimilar to system S_b, denoted by $S_a \cong_{\mathcal{S}}^{\varepsilon} S_b$, if there exists a relation R satisfying:

1. R is an ε-approximate simulation relation from S_a to S_b;
2. R^{-1} is an ε-approximate simulation relation from S_b to S_a.

Definition (Approximate bisimulation)

Consider two metric systems S_a and S_b with $Y_a = Y_b$, and let $\varepsilon \in \mathbb{R}_0^+$. We say that system S_a is ε-approximately bisimilar to system S_b, denoted by $S_a \cong_{\mathcal{S}}^{\varepsilon} S_b$, if there exists a relation R satisfying:

1. R is an ε-approximate simulation relation from S_a to S_b;
2. R^{-1} is an ε-approximate simulation relation from S_b to S_a.

- A word of caution:
  - $S_a \cong_{\mathcal{S}}^{\varepsilon_{ab}} S_b$ and $S_b \cong_{\mathcal{S}}^{\varepsilon_{bc}} S_c$ implies $S_a \cong_{\mathcal{S}}^{\varepsilon_{ab}+\varepsilon_{bc}} S_c$.

Definition (Approximate bisimulation)

Consider two metric systems S_a and S_b with $Y_a = Y_b$, and let $\varepsilon \in \mathbb{R}_0^+$. We say that system S_a is ε-approximately bisimilar to system S_b, denoted by $S_a \cong_{\mathcal{S}}^{\varepsilon} S_b$, if there exists a relation R satisfying:

1. R is an ε-approximate simulation relation from S_a to S_b;
2. R^{-1} is an ε-approximate simulation relation from S_b to S_a.

- A word of caution:
  - $S_a \cong_{\mathcal{S}}^{\varepsilon_{ab}} S_b$ and $S_b \cong_{\mathcal{S}}^{\varepsilon_{bc}} S_c$ implies $S_a \cong_{\mathcal{S}}^{\varepsilon_{ab}+\varepsilon_{bc}} S_c$.
  - ε-approximate bisimulation is not an equivalence notion!

CyPhyLab UCLA

Definition (Approximate alternating simulation relation)

Let S_a and S_b be metric systems with $Y_a = Y_b$ and let $\varepsilon \in \mathbb{R}_0^+$. A relation $R \subseteq X_a \times X_b$ is an ε-approximate alternating simulation relation from S_a to S_b if the following three conditions are satisfied:

1. for every $x_{a0} \in X_{a0}$ there exists $x_{b0} \in X_{b0}$ with $(x_{a0}, x_{b0}) \in R$;
2. for every $(x_a, x_b) \in R$ we have $\mathbf{d}(H_a(x_a), H_b(x_b)) \leq \varepsilon$;
3. for every $(x_a, x_b) \in R$ and for every $u_a \in U_a(x_a)$ there exists $u_b \in U_b(x_b)$ such that for every $x'_b \in \mathrm{Post}_{u_b}(x_b)$ there exists $x'_a \in \mathrm{Post}_{u_a}(x_a)$ satisfying $(x'_a, x'_b) \in R$.

We say that S_a is ε-approximately alternatingly simulated by S_b or that S_b ε-approximately alternatingly simulates S_a, denoted by $S_a \preceq^{\varepsilon}_{\mathcal{AS}} S_b$, if there exists an ε-approximate alternating simulation relation from S_a to S_b.

Approximate system relationships

Approximate alternating simulation

Definition (Approximate alternating simulation relation)

Let S_a and S_b be metric systems with $Y_a = Y_b$ and let $\varepsilon \in \mathbb{R}_0^+$. A relation $R \subseteq X_a \times X_b$ is an ε-approximate alternating simulation relation from S_a to S_b if the following three conditions are satisfied:

1. for every $x_{a0} \in X_{a0}$ there exists $x_{b0} \in X_{b0}$ with $(x_{a0}, x_{b0}) \in R$;
2. for every $(x_a, x_b) \in R$ we have $\mathbf{d}(H_a(x_a), H_b(x_b)) \leq \varepsilon$;
3. for every $(x_a, x_b) \in R$ and for every $u_a \in U_a(x_a)$ there exists $u_b \in U_b(x_b)$ such that for every $x'_b \in \mathrm{Post}_{u_b}(x_b)$ there exists $x'_a \in \mathrm{Post}_{u_a}(x_a)$ satisfying $(x'_a, x'_b) \in R$.

We say that S_a is ε-approximately alternatingly simulated by S_b or that S_b ε-approximately alternatingly simulates S_a, denoted by $S_a \preceq^{\varepsilon}_{\mathcal{AS}} S_b$, if there exists an ε-approximate alternating simulation relation from S_a to S_b.

- Approximate alternating bisimulation relations can be defined as usual by symmetrizing the above definition.

CyPhyLab UCLA

Approximate system relationships
Example

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

with trajectory $\xi_x(t) = e^{-t}x$.

Approximate system relationships
Example

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

  with trajectory $\xi_x(t) = e^{-t}x$.
- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.

CyPhyLab UCLA

Approximate system relationships

Example

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

with trajectory $\xi_x(t) = e^{-t}x$.

- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.
- Why: consider a pair $(x, x') \in R_\varepsilon$ and a transition $x \longrightarrow x''$ in $S(\Sigma)$.

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

  with trajectory $\xi_x(t) = e^{-t}x$.
- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.
- Why: consider a pair $(x, x') \in R_\varepsilon$ and a transition $x \longrightarrow x''$ in $S(\Sigma)$.
- By definition of $\longrightarrow$ we have $x'' = \xi_x(\tau) = e^{-\tau}x$.

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

  with trajectory $\xi_x(t) = e^{-t}x$.
- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.
- Why: consider a pair $(x, x') \in R_\varepsilon$ and a transition $x \longrightarrow x''$ in $S(\Sigma)$.
- By definition of $\longrightarrow$ we have $x'' = \xi_x(\tau) = e^{-\tau}x$.
- Does there exist a point x''' satisfying $x' \longrightarrow x'''$ and $(x'', x''') \in R_\varepsilon$?

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

with trajectory $\xi_x(t) = e^{-t}x$.

- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.
- Why: consider a pair $(x, x') \in R_\varepsilon$ and a transition $x \longrightarrow x''$ in $S(\Sigma)$.
- By definition of $\longrightarrow$ we have $x'' = \xi_x(\tau) = e^{-\tau}x$.
- Does there exist a point x''' satisfying $x' \longrightarrow x'''$ and $(x'', x''') \in R_\varepsilon$?
$x''' = \xi_{x'}(\tau)$:

Approximate system relationships

Example

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

  with trajectory $\xi_x(t) = e^{-t}x$.
- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.
- Why: consider a pair $(x, x') \in R_\varepsilon$ and a transition $x \longrightarrow x''$ in $S(\Sigma)$.
- By definition of $\longrightarrow$ we have $x'' = \xi_x(\tau) = e^{-\tau}x$.
- Does there exist a point x''' satisfying $x' \longrightarrow x'''$ and $(x'', x''') \in R_\varepsilon$? $x''' = \xi_{x'}(\tau)$:

$$\|x'' - x'''\|$$

Approximate system relationships

Example

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

  with trajectory $\xi_x(t) = e^{-t}x$.
- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.
- Why: consider a pair $(x, x') \in R_\varepsilon$ and a transition $x \longrightarrow x''$ in $S(\Sigma)$.
- By definition of $\longrightarrow$ we have $x'' = \xi_x(\tau) = e^{-\tau}x$.
- Does there exist a point x''' satisfying $x' \longrightarrow x'''$ and $(x'', x''') \in R_\varepsilon$? $x''' = \xi_{x'}(\tau)$:

$$\|x'' - x'''\| = \|\xi_x(\tau) - \xi_{x'}(\tau)\|$$

Approximate system relationships
Example

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

  with trajectory $\xi_x(t) = e^{-t}x$.
- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.
- Why: consider a pair $(x, x') \in R_\varepsilon$ and a transition $x \longrightarrow x''$ in $S(\Sigma)$.
- By definition of $\longrightarrow$ we have $x'' = \xi_x(\tau) = e^{-\tau}x$.
- Does there exist a point x''' satisfying $x' \longrightarrow x'''$ and $(x'', x''') \in R_\varepsilon$?
  $x''' = \xi_{x'}(\tau)$:

$$\|x'' - x'''\| = \|\xi_x(\tau) - \xi_{x'}(\tau)\| = \|e^{-\tau}x - e^{-\tau}x'\|$$

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

  with trajectory $\xi_x(t) = e^{-t}x$.
- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.
- Why: consider a pair $(x, x') \in R_\varepsilon$ and a transition $x \longrightarrow x''$ in $S(\Sigma)$.
- By definition of $\longrightarrow$ we have $x'' = \xi_x(\tau) = e^{-\tau}x$.
- Does there exist a point x''' satisfying $x' \longrightarrow x'''$ and $(x'', x''') \in R_\varepsilon$?
  $x''' = \xi_{x'}(\tau)$:

$$\|x'' - x'''\| = \|\xi_x(\tau) - \xi_{x'}(\tau)\| = \|e^{-\tau}x - e^{-\tau}x'\| = \|e^{-\tau}\|\|x - x'\|$$

Approximate system relationships
Example

- All the results in this lecture are generalizations of the following simple idea.
- Consider the dynamical system Σ described by the linear differential equation:

$$\frac{d}{dt}\xi = -\xi, \quad \xi(t) \in \mathbb{R}, t \in \mathbb{R}_0^+ \tag{1}$$

  with trajectory $\xi_x(t) = e^{-t}x$.
- Claim: the relation $R_\varepsilon \subseteq \mathbb{R} \times \mathbb{R}$ defined by $(x, x') \in R_\varepsilon$ iff $\|x - x'\| \leq \varepsilon$ is an ε-approximate simulation relation from $S(\Sigma)$ to $S(\Sigma)$.
- Why: consider a pair $(x, x') \in R_\varepsilon$ and a transition $x \longrightarrow x''$ in $S(\Sigma)$.
- By definition of $\longrightarrow$ we have $x'' = \xi_x(\tau) = e^{-\tau}x$.
- Does there exist a point x''' satisfying $x' \longrightarrow x'''$ and $(x'', x''') \in R_\varepsilon$?
  $x''' = \xi_{x'}(\tau)$:

$$\|x'' - x'''\| = \|\xi_x(\tau) - \xi_{x'}(\tau)\| = \|e^{-\tau}x - e^{-\tau}x'\| = \|e^{-\tau}\|\|x - x'\| \leq \|x - x'\| \leq \varepsilon.$$

CyPhyLab UCLA

- Although we work with control systems in discrete-time we will use the sampling time τ as a design parameter. Therefore, we need to recall continuous-time control systems and the corresponding stability properties.

$$\frac{d}{dt}\xi = A\xi + B\nu \tag{2}$$

with $\xi(t) \in \mathbb{R}^n$, $\nu(t) \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $t \in \mathbb{R}_0^+$.

Approximate finite-state abstractions

Stability of control systems

- Although we work with control systems in discrete-time we will use the sampling time τ as a design parameter. Therefore, we need to recall continuous-time control systems and the corresponding stability properties.

$$\frac{d}{dt}\xi = A\xi + B\nu \tag{2}$$

with $\xi(t) \in \mathbb{R}^n$, $\nu(t) \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $t \in \mathbb{R}_0^+$.

Definition (Input-to-state stability)

A linear control system $(\mathbb{R}^n, \mathbb{R}^m, A, B)$ is said to be input-to-state stable (ISS) when there exist constants $\kappa, \lambda, \rho \in \mathbb{R}^+$ such that for any $x \in \mathbb{R}^n$, any $\nu \in \mathcal{U}$, and any $t \in \mathbb{R}_0^+$, the following inequality is satisfied:

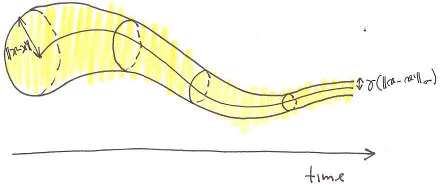
$$\|\xi_{x,\nu}(t)\| \leq \kappa e^{-\lambda t}\|x\| + \rho\|\nu\|.$$

Definition (Input-to-state stability)

A linear control system $(\mathbb{R}^n, \mathbb{R}^m, A, B)$ is said to be input-to-state stable (ISS) when there exist constants $\kappa, \lambda, \rho \in \mathbb{R}^+$ such that for any $x \in \mathbb{R}^n$, any $\nu \in \mathcal{U}$, and any $t \in \mathbb{R}^+$, the following inequality is satisfied:

$$\|\xi_{x,\nu}(t)\| \leq \kappa e^{-\lambda t}\|x\| + \rho\|\nu\|.$$

- For linear control systems, ISS provides "self-similarity" among trajectories:

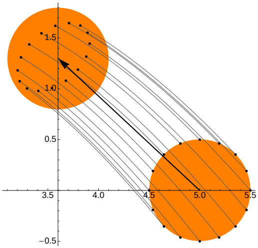


Definition (Input-to-state stability)

A linear control system $(\mathbb{R}^n, \mathbb{R}^m, A, B)$ is said to be input-to-state stable (ISS) when there exist constants $\kappa, \lambda, \rho \in \mathbb{R}^+$ such that for any $x \in \mathbb{R}^n$, any $\nu \in \mathcal{U}$, and any $t \in \mathbb{R}^+$, the following inequality is satisfied:

$$\|\xi_{x,\nu}(t)\| \leq \kappa e^{-\lambda t}\|x\| + \rho\|\nu\|.$$

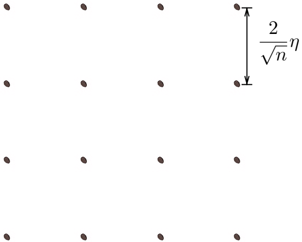
- For linear control systems, ISS provides “self-similarity” among trajectories:



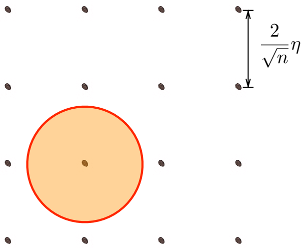
Approximate finite-state abstractions

A simple construction of abstractions

- If trajectories are "self-similar", why don't we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



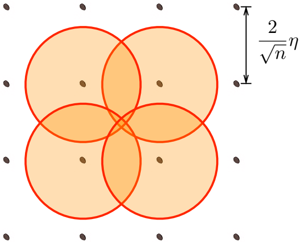
- If trajectories are "self-similar", why don't we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



Approximate finite-state abstractions

A simple construction of abstractions

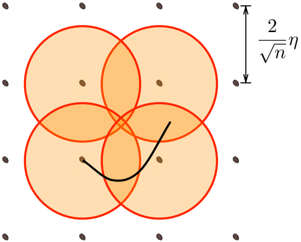
- If trajectories are "self-similar", why don't we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



Approximate finite-state abstractions

A simple construction of abstractions

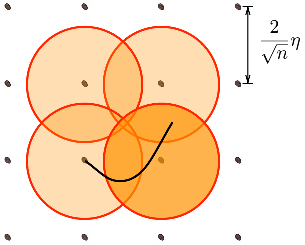
- If trajectories are “self-similar”, why don’t we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



Approximate finite-state abstractions

A simple construction of abstractions

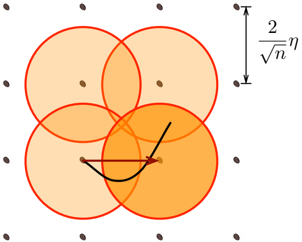
- If trajectories are “self-similar”, why don’t we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



Approximate finite-state abstractions

A simple construction of abstractions

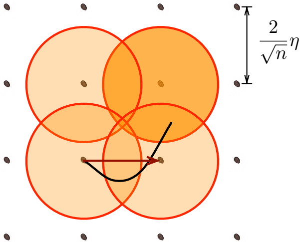
- If trajectories are “self-similar”, why don’t we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



Approximate finite-state abstractions

A simple construction of abstractions

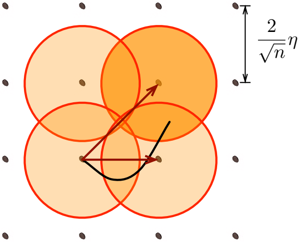
- If trajectories are “self-similar”, why don’t we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



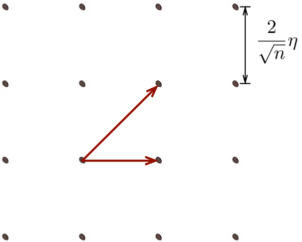
Approximate finite-state abstractions

A simple construction of abstractions

- If trajectories are “self-similar”, why don’t we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



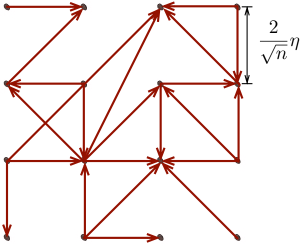
- If trajectories are “self-similar”, why don’t we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



Approximate finite-state abstractions

A simple construction of abstractions

- If trajectories are “self-similar”, why don’t we keep a single trajectory as representative of the infinitely many trajectories starting inside a ball?



Approximate finite-state abstractions

Formalizing the simple construction of abstractions

- We quantize:
  - time using the parameter τ;
  - states using the parameter η;
  - and inputs using the parameter ω.

Approximate finite-state abstractions

Formalizing the simple construction of abstractions

- We quantize:
  - time using the parameter τ;
  - states using the parameter η;
  - and inputs using the parameter ω.

Definition

The system $S_{\tau\eta\omega}(\Sigma) = (X, X_0, U, \longrightarrow, Y, H)$ associated with a linear control system $\Sigma = (\mathbb{R}^n, \mathbb{R}^m, A, B)$ and with quantization parameters $\tau, \eta, \omega \in \mathbb{R}^+$ consists of:

- $X = \left\{x \in \mathbb{R}^n \mid x_i = \ell_i \frac{2}{\sqrt{n}}\eta \text{ for some } \ell_i \in \mathbb{Z} \text{ and } i = 1, 2, \ldots, n\right\}$;
- $U = \left\{u \in \mathbb{R}^m \mid x_i = \ell_i \frac{2}{\sqrt{n}}\omega \text{ for some } \ell_i \in \mathbb{Z} \text{ and } i = 1, 2, \ldots, m\right\}$;
- $x \xrightarrow{u} x'$ if $\xi_{x,\nu} : [0, \tau] \to \mathbb{R}^n$, with $\nu(t) = u \in U$ for $t \in [0, \tau]$, satisfies $\|\xi_{x,\nu}(\tau) - x'\| \leq \eta$;
- $Y = \mathbb{R}^n$;
- $H = \imath : X \hookrightarrow \mathbb{R}^n$.

Approximate finite-state abstractions

ISS Lyapunov functions

- When is $S(\Sigma)$ approximately bisimilar to $S_{\tau\eta\omega}(\Sigma)$?

Approximate finite-state abstractions
ISS Lyapunov functions

- When is $S(\Sigma)$ approximately bisimilar to $S_{\tau\eta\omega}(\Sigma)$?

Definition (ISS Lyapunov function)

Let $(\mathbb{R}^n, \mathbb{R}^m, A, B)$ be a linear control system and consider a function $V : \mathbb{R}^n \to \mathbb{R}$ satisfying the following three properties:

1. V is continuous on $\mathbb{R}^n$ and smooth on $\mathbb{R}^n \backslash \{0\}$;
2. $V(x) \geq 0$ for all $x \in \mathbb{R}^n$;
3. $V(x) = 0$ implies $x = 0$.

The function V is an ISS-Lyapunov function for Σ if there exist constants $\lambda, \sigma \in \mathbb{R}^+$ such that for all $x \in \mathbb{R}^n \backslash \{0\}$, $u \in \mathbb{R}^m$, the following inequality holds:

$$\frac{\partial V}{\partial x}(Ax + Bu) \leq -\lambda V(x) + \sigma \|u\|.$$

- When is $S(\Sigma)$ approximately bisimilar to $S_{\tau\eta\omega}(\Sigma)$?

Definition (ISS Lyapunov function)

Let $(\mathbb{R}^n, \mathbb{R}^m, A, B)$ be a linear control system and consider a function $V : \mathbb{R}^n \to \mathbb{R}$ satisfying the following three properties:

1. V is continuous on $\mathbb{R}^n$ and smooth on $\mathbb{R}^n \backslash \{0\}$;
2. $V(x) \geq 0$ for all $x \in \mathbb{R}^n$;
3. $V(x) = 0$ implies $x = 0$.

The function V is an ISS-Lyapunov function for Σ if there exist constants $\lambda, \sigma \in \mathbb{R}^+$ such that for all $x \in \mathbb{R}^n \backslash \{0\}$, $u \in \mathbb{R}^m$, the following inequality holds:

$$\frac{\partial V}{\partial x}(Ax + Bu) \leq -\lambda V(x) + \sigma \|u\|.$$

Theorem

A linear control system Σ is input-to-state stable iff Σ admits an ISS-Lyapunov function.

- For linear systems, existence of an ISS-Lyapunov function also implies the existence of an ISS-Lyapunov function of the form $V(x) = \sqrt{x^T P x}$ for some $P \in \mathbb{R}^{n \times n}$ that is:
  - symmetric ($P^T = P$);
  - positive-definite ($x^T P x > 0$ for all $x \neq 0$).

Approximate finite-state abstractions
ISS Lyapunov functions

- For linear systems, existence of an ISS-Lyapunov function also implies the existence of an ISS-Lyapunov function of the form $V(x) = \sqrt{x^T P x}$ for some $P \in \mathbb{R}^{n \times n}$ that is:
  - symmetric ($P^T = P$);
  - positive-definite ($x^T P x > 0$ for all $x \neq 0$).
- Moreover, Lyapunov functions of this form satisfy several useful inequalities.

Proposition

Let $V : \mathbb{R}^n \to \mathbb{R}_0^+$ be a function of the form $V(x) = \sqrt{x^T P x}$ for some symmetric and positive-definite $P \in \mathbb{R}^{n \times n}$. There exist constants $\underline{\alpha}, \overline{\alpha}, \gamma \in \mathbb{R}^+$ such that for all $x, x', x'' \in \mathbb{R}^n$, the following inequalities are satisfied:

$$\underline{\alpha}\|x\| \leq V(x) \leq \overline{\alpha}\|x\|,$$
$$V(x - x') - V(x - x'') \leq \gamma\|x' - x''\|.$$

$$\underline{\alpha} = \sqrt{\lambda_m(P)}, \quad \overline{\alpha} = \sqrt{\lambda_M(P)}, \quad \gamma = \frac{\lambda_M(P)}{\sqrt{\lambda_m(P)}}$$

Approximate finite-state abstractions

Existence

Theorem

Let $\Sigma = (\mathbb{R}^n, \mathbb{R}^m, A, B)$ be a linear control system admiting aa ISS-Lyapunov function V of the form $V(x) = \sqrt{x^T P x}$ with $P \in \mathcal{SP}(n)$. For any desired precision $\varepsilon \in \mathbb{R}^+$, for any desired time quantization $\tau \in \mathbb{R}^+$, for any desired input quantization $\omega \in \mathbb{R}^+$, and for any space quantization $\eta \in \mathbb{R}^+$ satisfying:

$$\eta \leq \min \left\{ \gamma^{-1} \underline{\alpha} \varepsilon \left(1 - e^{-\lambda\tau} \right) - \gamma^{-1} \lambda^{-1} \sigma \, \omega, \overline{\alpha}^{-1} \underline{\alpha} \varepsilon \right\}, \tag{3}$$

the relation $R_\varepsilon \subseteq X_{\tau\eta\omega} \times X_\tau$ defined by:

$$R_\varepsilon = \{ (x_{\tau\eta\omega}, x_\tau) \in X_{\tau\eta\omega} \times X_\tau \mid V(x_\tau - x_{\tau\eta\omega}) \leq \underline{\alpha}\varepsilon \}$$

is an ε-approximate bisimulation relation between $S(\Sigma)$ and $S_{\tau\eta\omega}(\Sigma)$.

$$\underline{\alpha} = \sqrt{\lambda_m(P)}, \quad \overline{\alpha} = \sqrt{\lambda_M(P)}, \quad \gamma = \frac{\lambda_M(P)}{\sqrt{\lambda_m(P)}}$$

$$\frac{\partial V}{\partial x}(Ax + Bu) \leq -\lambda V(x) + \sigma \|u\|.$$

Approximate finite-state abstractions

Existence

Theorem

Let $\Sigma = (\mathbb{R}^n, \mathbb{R}^m, A, B)$ *be a linear control system admiting aa ISS-Lyapunov function* V *of the form* $V(x) = \sqrt{x^T P x}$ *with* $P \in \mathcal{SP}(n)$*. For any desired precision* $\varepsilon \in \mathbb{R}^+$*, for any desired time quantization* $\tau \in \mathbb{R}^+$*, for any desired input quantization* $\omega \in \mathbb{R}^+$*, and for any space quantization* $\eta \in \mathbb{R}^+$ *satisfying:*

$$\eta \leq \min\left\{\gamma^{-1}\underline{\alpha}\varepsilon\left(1 - e^{-\lambda\tau}\right) - \gamma^{-1}\lambda^{-1}\sigma\,\omega, \overline{\alpha}^{-1}\underline{\alpha}\varepsilon\right\}, \tag{3}$$

the relation $R_\varepsilon \subseteq X_{\tau\eta\omega} \times X_\tau$ *defined by:*

$$R_\varepsilon = \{(x_{\tau\eta\omega}, x_\tau) \in X_{\tau\eta\omega} \times X_\tau \mid V(x_\tau - x_{\tau\eta\omega}) \leq \underline{\alpha}\varepsilon\}$$

is an ε*-approximate bisimulation relation between* $S(\Sigma)$ *and* $S_{\tau\eta\omega}(\Sigma)$*.*

- If we restrict the computation of $S_{\tau\eta\omega}(\Sigma)$ to bounded subsets of $\mathbb{R}^n$ and $\mathbb{R}^m$, X and U become finite.

Theorem

Let $\Sigma = (\mathbb{R}^n, \mathbb{R}^m, A, B)$ be a linear control system admiting aa ISS-Lyapunov function V of the form $V(x) = \sqrt{x^T P x}$ with $P \in \mathcal{SP}(n)$. For any desired precision $\varepsilon \in \mathbb{R}^+$, for any desired time quantization $\tau \in \mathbb{R}^+$, for any desired input quantization $\omega \in \mathbb{R}^+$, and for any space quantization $\eta \in \mathbb{R}^+$ satisfying:

$$\eta \leq \min\left\{\gamma^{-1}\underline{\alpha}\varepsilon\left(1 - e^{-\lambda\tau}\right) - \gamma^{-1}\lambda^{-1}\sigma\,\omega, \overline{\alpha}^{-1}\underline{\alpha}\varepsilon\right\}, \tag{3}$$

the relation $R_\varepsilon \subseteq X_{\tau\eta\omega} \times X_\tau$ defined by:

$$R_\varepsilon = \{(x_{\tau\eta\omega}, x_\tau) \in X_{\tau\eta\omega} \times X_\tau \mid V(x_\tau - x_{\tau\eta\omega}) \leq \underline{\alpha}\varepsilon\}$$

is an ε-approximate bisimulation relation between $S(\Sigma)$ and $S_{\tau\eta\omega}(\Sigma)$.

- What if Σ is not ISS?

Approximate finite-state abstractions

Existence

Theorem

Let $\Sigma = (\mathbb{R}^n, \mathbb{R}^m, A, B)$ *be a linear control system admiting aa ISS-Lyapunov function* V *of the form* $V(x) = \sqrt{x^T P x}$ *with* $P \in \mathcal{SP}(n)$. *For any desired precision* $\varepsilon \in \mathbb{R}^+$, *for any desired time quantization* $\tau \in \mathbb{R}^+$, *for any desired input quantization* $\omega \in \mathbb{R}^+$, *and for any space quantization* $\eta \in \mathbb{R}^+$ *satisfying:*

$$\eta \leq \min\left\{\gamma^{-1}\underline{\alpha}\varepsilon\left(1 - e^{-\lambda\tau}\right) - \gamma^{-1}\lambda^{-1}\sigma\,\omega, \overline{\alpha}^{-1}\underline{\alpha}\varepsilon\right\}, \tag{3}$$

the relation $R_\varepsilon \subseteq X_{\tau\eta\omega} \times X_\tau$ *defined by:*

$$R_\varepsilon = \{(x_{\tau\eta\omega}, x_\tau) \in X_{\tau\eta\omega} \times X_\tau \mid V(x_\tau - x_{\tau\eta\omega}) \leq \underline{\alpha}\varepsilon\}$$

is an ε*-approximate bisimulation relation between* $S(\Sigma)$ *and* $S_{\tau\eta\omega}(\Sigma)$.

- What if Σ is not ISS? We design a preliminary controller rendering Σ ISS. This is a simple task for linear control systems.

Approximate finite-state abstractions

Example

- Consider the linear control system defined by:

$$A = \begin{bmatrix} -1 & 1 \\ -8 & 5 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

- Since this system is not ISS, we first design the feedback control law:

$$u = Kx + u' = 7x_1 - 6x_2 + u'$$

rendering the controlled system $(\mathbb{R}^n, \mathbb{R}^m, A + BK, B)$ ISS where:

$$A + BK = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}.$$

Approximate finite-state abstractions

Example

- Consider the linear control system defined by:

$$A = \begin{bmatrix} -1 & 1 \\ -8 & 5 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

- Since this system is not ISS, we first design the feedback control law:

$$u = Kx + u' = 7x_1 - 6x_2 + u'$$

rendering the controlled system $(\mathbb{R}^n, \mathbb{R}^m, A + BK, B)$ ISS where:

$$A + BK = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}.$$

- Using the function $V(x) = \sqrt{x^T P x}$ with:

$$P = \begin{bmatrix} 1 & \frac{1}{16} \\ \frac{1}{16} & 1 \end{bmatrix}$$

as a Lyapunov function we obtain:

$$\gamma = \frac{17}{4\sqrt{15}}, \quad \lambda = \frac{16 - \sqrt{2}}{17}, \quad \underline{\alpha} = \frac{15}{16}, \quad \overline{\alpha} = \frac{17}{16}.$$

CyPhyLab UCLA

Approximate finite-state abstractions

Example

- For simplicity, we assume the only available inputs to be $\{0, \pm 0.25, \pm 0.5\}$;

- For simplicity, we assume the only available inputs to be $\{0, \pm 0.25, \pm 0.5\}$;
- If we choose a sampling time $\tau = 0.25$ and a precision $\varepsilon = 0.1$ we conclude from (3) that η needs to be smaller than 0.017. We choose $\eta = \frac{\sqrt{2}}{100} \approx 0.014$ and restrict Σ to the set $[-1, 1] \times [-1, 1]$.

Approximate finite-state abstractions

Example

- For simplicity, we assume the only available inputs to be $\{0, \pm 0.25, \pm 0.5\}$;
- If we choose a sampling time $\tau = 0.25$ and a precision $\varepsilon = 0.1$ we conclude from (3) that η needs to be smaller than 0.017. We choose $\eta = \frac{\sqrt{2}}{100} \approx 0.014$ and restrict Σ to the set $[-1, 1] \times [-1, 1]$.
- We now consider a safety game with specification set:

$$W = [-0.3, -0.1] \times [-0.1, 0.1]$$

  and compute the maximal controlled invariant subset.

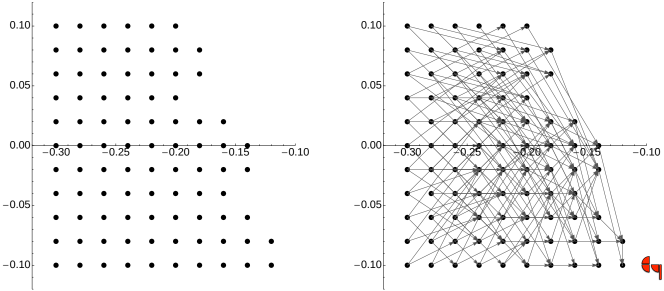
Approximate finite-state abstractions

Example

- For simplicity, we assume the only available inputs to be $\{0, \pm 0.25, \pm 0.5\}$;
- If we choose a sampling time $\tau = 0.25$ and a precision $\varepsilon = 0.1$ we conclude from (3) that η needs to be smaller than 0.017. We choose $\eta = \frac{\sqrt{2}}{100} \approx 0.014$ and restrict Σ to the set $[-1, 1] \times [-1, 1]$.
- We now consider a safety game with specification set:

$$W = [-0.3, -0.1] \times [-0.1, 0.1]$$

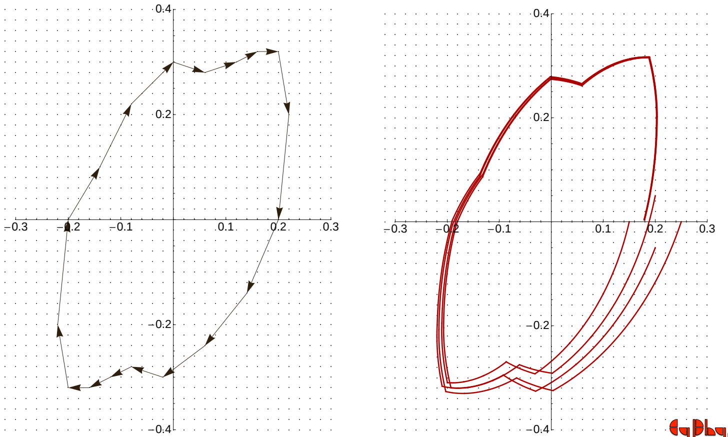
and compute the maximal controlled invariant subset.



- Consider now the specification requiring a periodic orbit visiting the points $(-0.2, 0)$ and $(0.2, 0)$.

Approximate finite-state abstractions
Example

- Consider now the specification requiring a periodic orbit visiting the points $(-0.2, 0)$ and $(0.2, 0)$.



- How about more general classes of systems?

- How about more general classes of systems?
  - The main result also holds for switched linear systems with a common ISS Lyapunov function.

- How about more general classes of systems?
  - The main result also holds for switched linear systems with a common ISS Lyapunov function.
  - When a common ISS Lyapunov function does not exist, we can impose dwell time requirements as part of the specification.

- How about more general classes of systems?
  - The main result also holds for switched linear systems with a common ISS Lyapunov function.
  - When a common ISS Lyapunov function does not exist, we can impose dwell time requirements as part of the specification.
  - The main result also holds for nonlinear systems but one has to use incremental ISS instead of ISS. For linear systems: incremental ISS=ISS=asymptotic stability.

Approximate finite-state abstractions

Extensions of the main result

- How about more general classes of systems?
  - The main result also holds for switched linear systems with a common ISS Lyapunov function.
  - When a common ISS Lyapunov function does not exist, we can impose dwell time requirements as part of the specification.
  - The main result also holds for nonlinear systems but one has to use incremental ISS instead of ISS. For linear systems: incremental ISS=ISS=asymptotic stability.
  - The main result also holds for switched nonlinear systems with the previous provisos.

Approximate finite-state abstractions

Extensions of the main result

- How about more general classes of systems?
  - The main result also holds for switched linear systems with a common ISS Lyapunov function.
  - When a common ISS Lyapunov function does not exist, we can impose dwell time requirements as part of the specification.
  - The main result also holds for nonlinear systems but one has to use incremental ISS instead of ISS. For linear systems: incremental ISS=ISS=asymptotic stability.
  - The main result also holds for switched nonlinear systems with the previous provisos.
  - The main result also holds for all the above classes in the presence of disturbances in the differential equation, e.g., $\dot{\xi} = f(\xi, \nu, \delta)$. In this case we obtain an approximate alternating bisimulation.

CyPhyLab UCLA

- How about more general classes of systems?
  - The main result also holds for switched linear systems with a common ISS Lyapunov function.
  - When a common ISS Lyapunov function does not exist, we can impose dwell time requirements as part of the specification.
  - The main result also holds for nonlinear systems but one has to use incremental ISS instead of ISS. For linear systems: incremental ISS=ISS=asymptotic stability.
  - The main result also holds for switched nonlinear systems with the previous provisos.
  - The main result also holds for all the above classes in the presence of disturbances in the differential equation, e.g., $\dot{\xi} = f(\xi, \nu, \delta)$. In this case we obtain an approximate alternating bisimulation.
- What if we cannot or do not want to design a preliminary controller enforcing ISS?

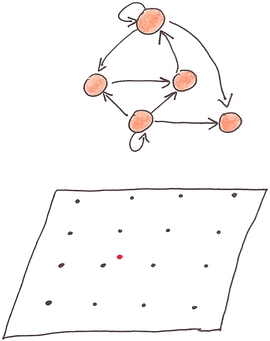
Approximate finite-state abstractions

Extensions of the main result

- How about more general classes of systems?
  - The main result also holds for switched linear systems with a common ISS Lyapunov function.
  - When a common ISS Lyapunov function does not exist, we can impose dwell time requirements as part of the specification.
  - The main result also holds for nonlinear systems but one has to use incremental ISS instead of ISS. For linear systems: incremental ISS=ISS=asymptotic stability.
  - The main result also holds for switched nonlinear systems with the previous provisos.
  - The main result also holds for all the above classes in the presence of disturbances in the differential equation, e.g., $\dot{\xi} = f(\xi, \nu, \delta)$. In this case we obtain an approximate alternating bisimulation.
- What if we cannot or do not want to design a preliminary controller enforcing ISS?
- Under a suitably modified construction, we can still compute a finite-state abstraction $S_{\tau\eta\omega}(\Sigma)$ satisfying $S_{\tau\eta\omega}(\Sigma) \preceq^{\varepsilon}_{\mathcal{AS}} S(\Sigma) \preceq^{\varepsilon}_{\mathcal{S}} S_{\tau\eta\omega}(\Sigma)$.

Controller refinement
A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.

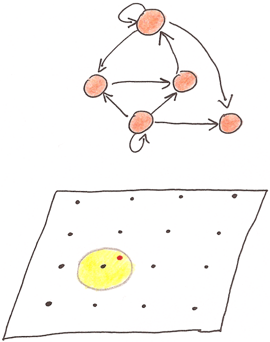


Current time: t_k
Current physical state: x

Controller refinement

A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.

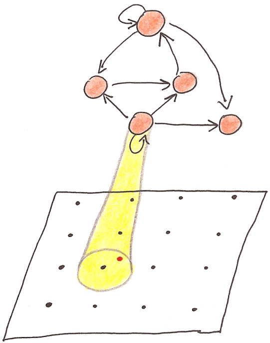


Current time: t_k
Current physical state: x

Controller refinement

A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.

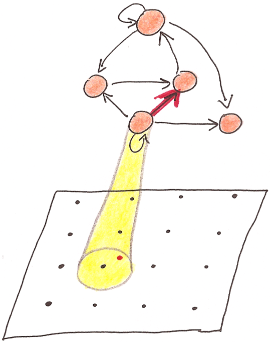


Current time: t_k
Current physical state: x
Current cyber state: q satisfying $(x, q) \in R$

Controller refinement

A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.

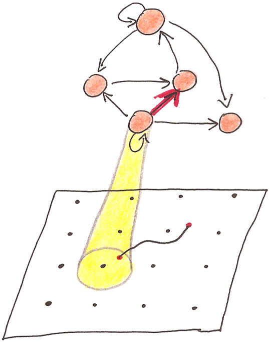


Current time: t_k
Current physical state: x
Current cyber state: q satisfying $(x, q) \in R$
Command from finite-state controller: u

Controller refinement

A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.

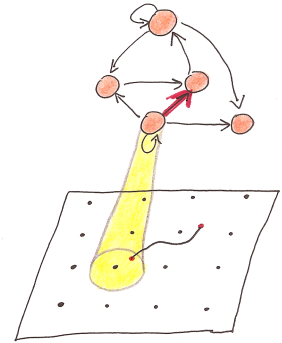


Current time: t_k
Current physical state: x
Current cyber state: q satisfying $(x, q) \in R$
Command from finite-state controller: u
Physical input: $\upsilon(t) = k(\xi_x(t), u), t \in [t_k, t_k + \tau[$

Controller refinement

A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.

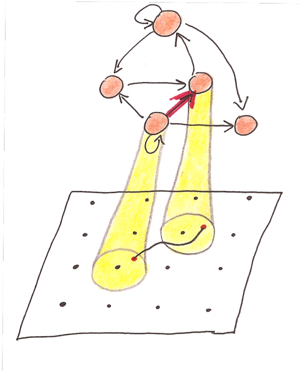


Current time: $t_{k+1} = t_k + \tau$
Current physical state: $\xi_{x\upsilon}(\tau)$
Current cyber state: q satisfying $(x, q) \in R$
Command from finite-state controller: u
Physical input: $\upsilon(t) = k(\xi_x(t), u), t \in [t_k, t_k + \tau[$

Controller refinement

A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.

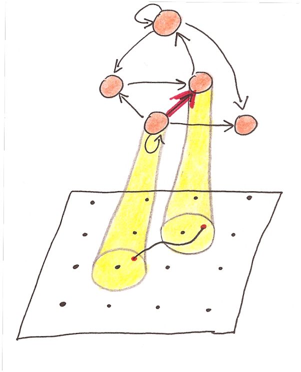


Current time: $t_{k+1} = t_k + \tau$
Current physical state: $\xi_{x\upsilon}(\tau)$
Current cyber state: q' satisfying $(\xi_{x\upsilon}(\tau), q') \in R$
Command from finite-state controller: u
Physical input: $\upsilon(t) = k(\xi_x(t), u), t \in [t_k, t_k + \tau[$

Controller refinement

A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.



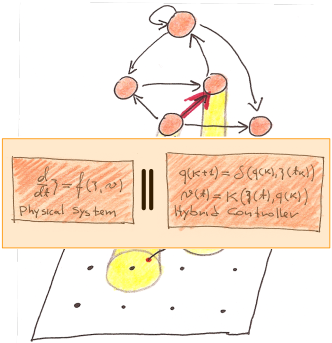
Current time: $t_{k+1} = t_k + \tau$
Current physical state: $\xi_{x\upsilon}(\tau)$
Current cyber state: q' satisfying $(\xi_{x\upsilon}(\tau), q') \in R$
Command from finite-state controller: u
Physical input: $\upsilon(t) = k(\xi_x(t), u), t \in [t_k, t_k + \tau[$

Since the hybrid controller is a formal model for the control software, it is conceptually simple to refine it into actual code for a target platform.

Controller refinement

A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.



Current time: $t_{k+1} = t_k + \tau$
Current physical state: $\xi_{x\upsilon}(\tau)$
Current cyber state: q' satisfying $(\xi_{x\upsilon}(\tau), q') \in R$
Command from finite-state controller: u
Physical input: $\upsilon(t) = k(\xi_x(t), u), t \in [t_k, t_k + \tau[$

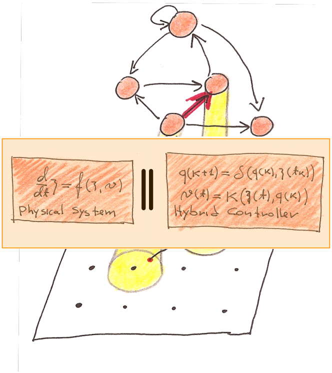
Since the hybrid controller is a formal model for the control software, it is conceptually simple to refine it into actual code for a target platform.

Controller refinement

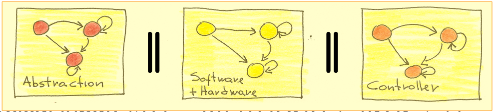
A pictorial description

- The refinement process consists in implementing the supervisory commands issued by the finite-state controller on the physical system.

Current time: $t_{k+1} = t_k + \tau$



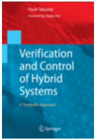
$\cong_\varepsilon$



Since the hybrid controller is a formal model for the control software, it is conceptually simple to refine it into actual code for a target platform.

References and more details

- All the missing details and references can be found in:

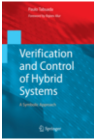


Verification and Control of Hybrid Systems: A Symbolic Approach
Springer, 2009.

- For the latest results:
  http://www.cyphylab.ee.ucla.edu/

References and more details

- All the missing details and references can be found in:



Verification and Control of Hybrid Systems: A Symbolic Approach
Springer, 2009.

- For the latest results:
  http://www.cyphylab.ee.ucla.edu/
- Have fun synthesizing controllers with PESSOA during the hands on session!