

On the control of complex systems through abstractions

Paulo Tabuada
Department of Electrical Engineering



**UNIVERSITY OF
NOTRE DAME**

Embedded devices? Where are they?

The screenshot shows a web browser window with the address bar displaying `http://www.miniusa.com/link/ourcars/performance_handling`. The website header includes the MINI logo and navigation links: HOME | MINI INTERNATIONAL | MINI FINANCIAL SERVICES | OUR VALUES | CONTACT & FAQs | OWNERS' LOUNGE. Below the header is a secondary navigation bar with links: OUR CARS | BUILD YOUR OWN | FIND A DEALER | GET IN THE LOOP | GEAR UP | MOTOR ON.

The main content area features a large article titled "DRIVE-BY-WIRE (ELECTRONIC) THROTTLE". The text explains that MINI uses a computer chip to send an electronic pulse to the engine when the accelerator is pressed, resulting in a faster, more consistent throttle response. It also notes that the computer prevents engine flooding, optimizing acceleration, performance, fuel economy, and reducing emissions.

To the right of the text is a section titled "9 FEATURES THAT GIVE MINIs THEIR EXHILARATING HANDLING" with a sub-section "DRIVE-BY-WIRE" and a progress indicator showing 1 out of 9 items selected.

Below the text is a 3D model of an engine. To the right, a larger 3D model of a MINI car is shown with a semi-transparent blue body, revealing the internal engine and drivetrain components. A "> CLOSE" button is visible in the top left corner of this 3D model's frame.

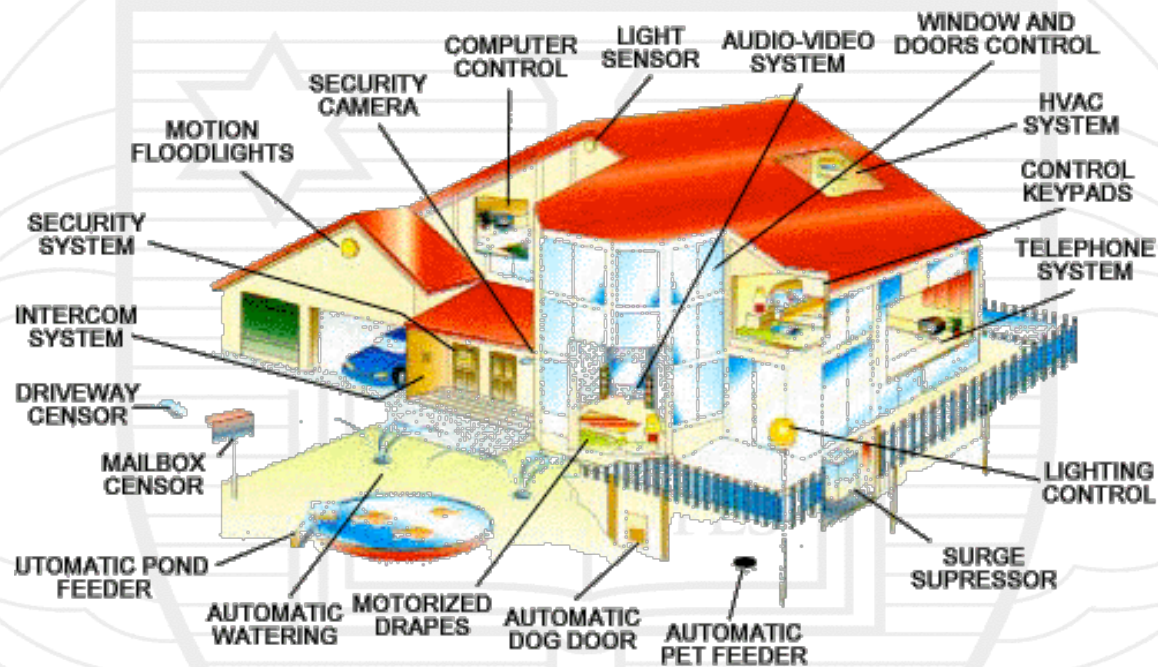
Embedded devices? Where are they?

The screenshot shows a web browser window with the address bar displaying 'http://www.miniusa.com/link/ourcars/performance_handling'. The website header includes the MINI logo and navigation links: HOME | MINI INTERNATIONAL | MINI FINANCIAL SERVICES | OUR VALUES | CONTACT & FAQs | OWNERS' LOUNGE. A secondary navigation bar contains: OUR CARS | BUILD YOUR OWN | FIND A DEALER | GET IN THE LOOP | GEAR UP | MOTOR ON.

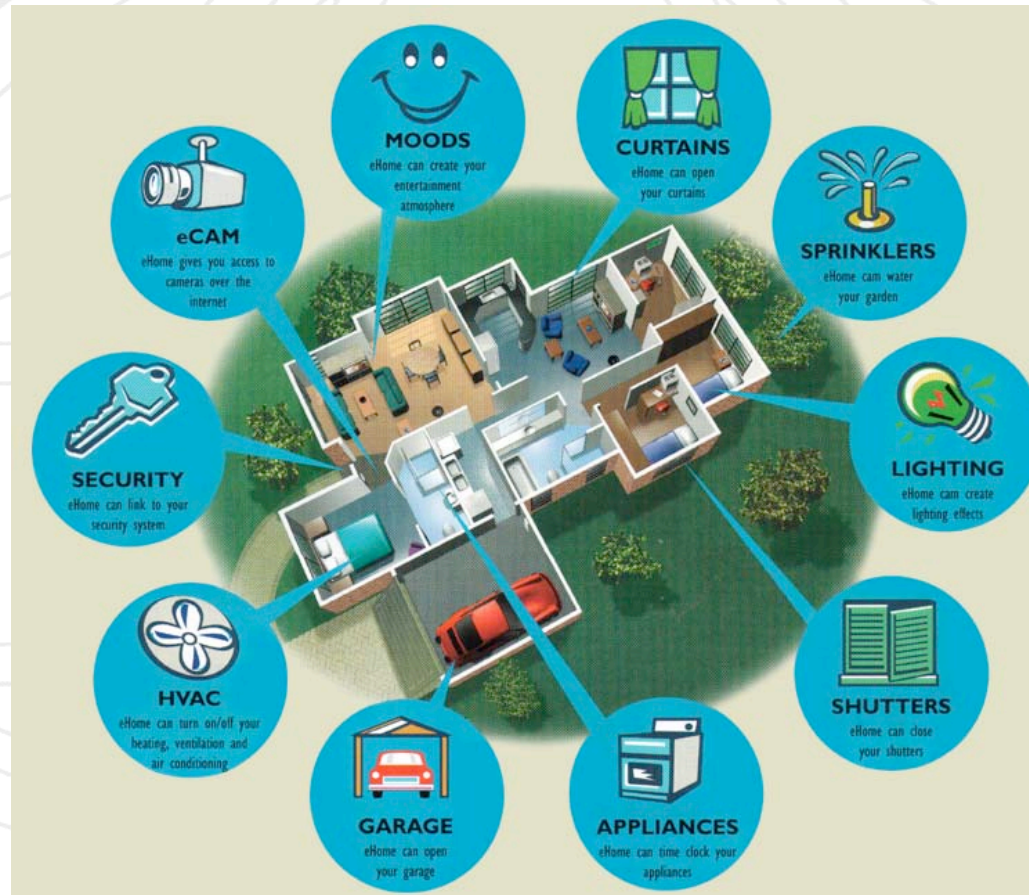
The main content area features a section titled 'ALPHABET BRAKES' with the following text: 'Like many cars, MINIs come with four-channel Anti-Lock Brakes (ABS). Unlike most cars, MINIs also come with Electronic Brakeforce Distribution (EBD) and Cornering Brake Control (CBC). Based on load conditions, EBD helps distribute the brake force to stabilize your MINI from front to back and help prevent nose-dives. The CBC is designed to even out the braking effect in emergency situations and help you keep from going into a skid. Confusing? We know. Just understand that after driving a MINI you might start thinking you have the skills to handle a Formula race car. Even if you don't.'

Below the text is an image of a brake disc assembly. To the right, a section titled '9 FEATURES THAT GIVE MINIs THEIR EXHILARATING HANDLING' includes a sub-section for 'ALPHABET BRAKES' with a progress indicator showing 9 items, with the first 8 being white and the last one blue. A modal window is open over this section, showing a 3D wireframe model of a MINI car with a highlighted orange area on the front wheel hub. The modal window has a '> CLOSE' button in the top left corner.

Embedded devices? Where are they?



Embedded devices? Where are they?



Embedded devices? Where are they?



Pentair Enclosures Medical Packaging Solutions

http://www.medical-enclosures.com/medical/subpage.asp?P=packagi


Apple Travel ND Research MultiMedia Music Banks Varios Cinema

embedded medical - Go... Pentair Enclosures Medi... http://www.validatedsof... http://www.advantech.c...


Instrumentation Console Equipment Enclosures Embedded Controller Systems

Embedded Controller Systems

Embedded controllers are at the heart of virtually all medical equipment - from diagnostic imaging systems, to scanners, to analyzers, and much more. With the demands for processing power and I/O increasing, and the need for clear upgrade paths - OEM's are turning to platforms such as VME and Compact PCI for their next generation of controller products. Our applications engineering team will work with you to identify the power, cooling, mounting and performance requirements for your application. Schroff is the leader in electronic packaging, and we offer a full range of system products including backplanes, chassis, front panels, power supplies, fan trays and more - to address your controller application.



© Copyright Pentair-EP 2005 | medicalinfo@pentair-ep.com



Embedded systems are computational devices with distinctive characteristics:

- They interface the physical world through sensors and actuators;
- They react to physical environment stimuli;
- They require networked and distributed information processing;
- They are frequently part of safety critical applications.

Embedded systems are computational devices with distinctive characteristics:

- They interface the physical world through sensors and actuators;
- They react to physical environment stimuli;
- They require networked and distributed information processing;
- They are frequently part of safety critical applications.

The mixed discrete (computation) and continuous (physical world) nature of embedded devices renders its analysis and design particularly difficult:

- Performance and correctness of operation of continuous controllers critically depend on software/hardware implementations;
- Real-time scheduling of control and communication tasks is aggravated by the reduced computational capabilities of embedded devices;
- Standard verification techniques cannot be applied to embedded software since differential equations cannot, in general, be captured by finite state models.

In today's talk I will try to describe two (three) different lines of research based on the same simple idea:

Vanquish complexity through abstraction.



In today's talk I will try to describe two (three) different lines of research based on the same simple idea:

Vanquish complexity through abstraction.

They are:

1. Synthesis of correct by design embedded control software through finite abstractions of control systems;
 - a) Distributed supervisory control for alternating simulation and bisimulation specifications;
2. Synthesis of real-time schedulers for stabilizing control tasks.

Current practice in embedded software design

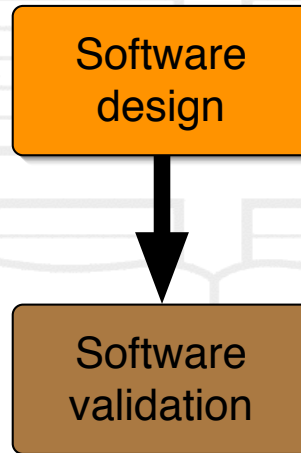
Current practice in embedded software development iteratively combines software design with validation techniques.



Software design

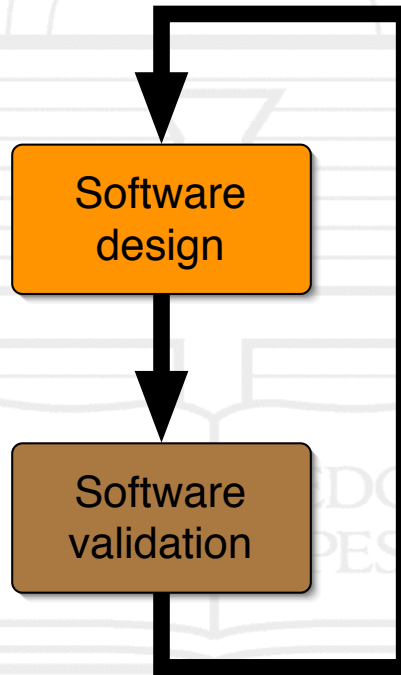
Current practice in embedded software design

Current practice in embedded software development iteratively combines software design with validation techniques.



Current practice in embedded software design

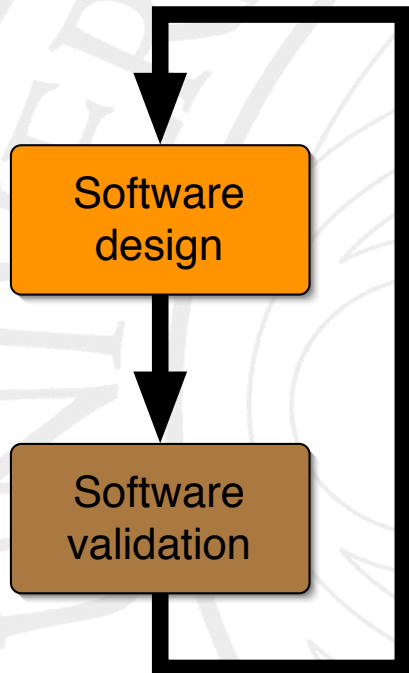
Current practice in embedded software development iteratively combines software design with validation techniques.



Current practice in embedded software design

Current practice in embedded software development iteratively combines software design with validation techniques.

This iterative scheme has several drawbacks:

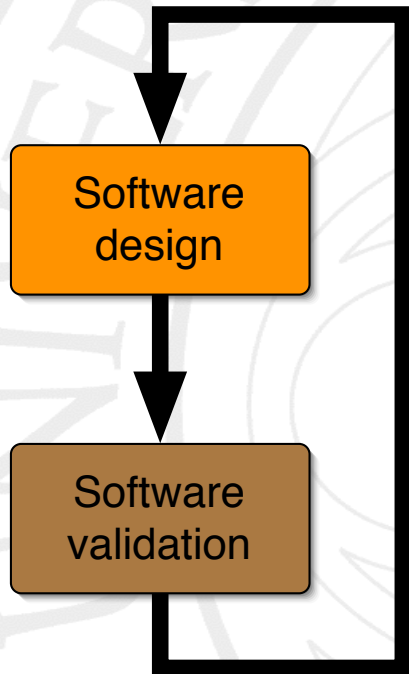


- Validation by extensive simulation and testing increases our confidence in the software but fails to provide adequate guarantees of correct operation and performance;
- Formal verification is currently limited to finite state systems and thus cannot be used to verify properties depending on continuous components;
- Extensive validation is time consuming thus increasing the cost and time-to-market of embedded software.

Current practice in embedded software design

Current practice in embedded software development iteratively combines software design with validation techniques.

This iterative scheme has several drawbacks:



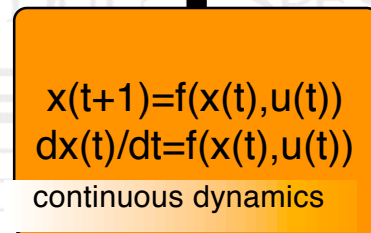
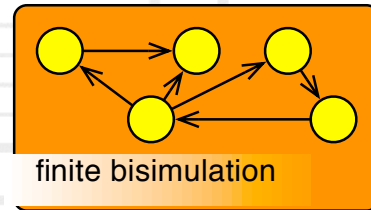
- Validation by extensive simulation and testing increases our confidence in the software but fails to provide adequate guarantees of correct operation and performance;
- Formal verification is currently limited to finite state systems and thus cannot be used to verify properties depending on continuous components;
- Extensive validation is time consuming thus increasing the cost and time-to-market of embedded software.

Some of these disadvantages can be mitigated by adopting a *correct by design* approach to the development of embedded control software.

Synthesizing correct embedded control software

We adopt a three phase approach to the synthesis of correct by design embedded control software:

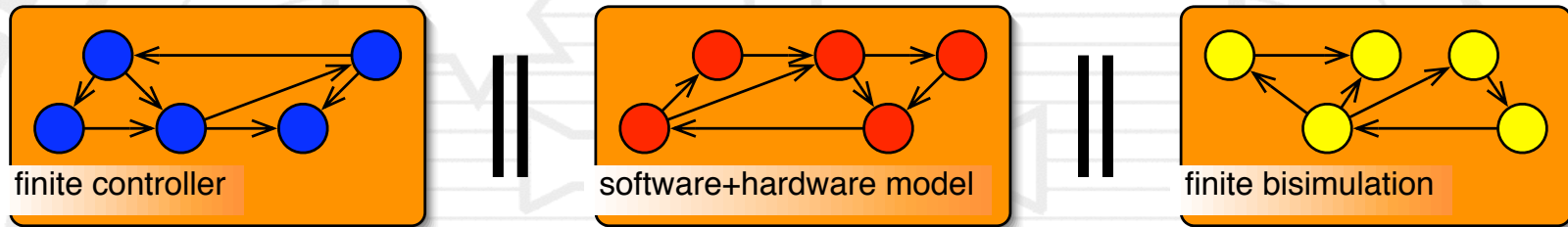
Abstraction



Synthesizing correct embedded control software

We adopt a three phase approach to the synthesis of correct by design embedded control software:

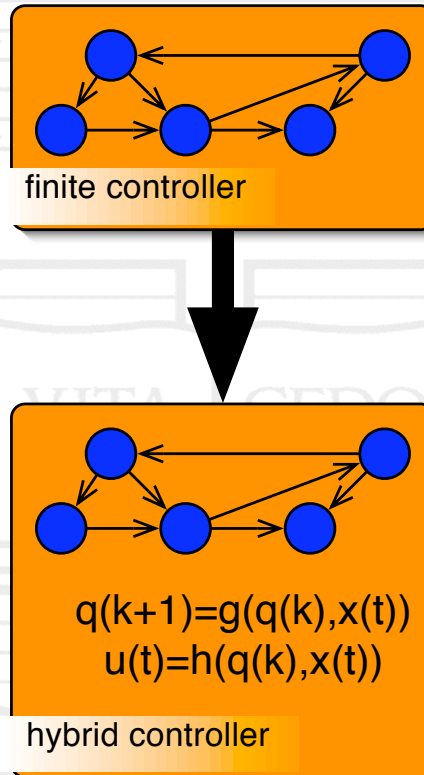
Controller design



Synthesizing correct embedded control software

We adopt a three phase approach to the synthesis of correct by design embedded control software:

Controller refinement



Synthesizing correct embedded control software

We adopt a three phase approach to the synthesis of correct by design embedded control software:

Ultimately, we would like to:

1. Specify the continuous dynamics;
2. Specify the software+hardware platform;
3. Define the specification;

Synthesizing correct embedded control software

We adopt a three phase approach to the synthesis of correct by design embedded control software:

Ultimately, we would like to:

1. Specify the continuous dynamics;
2. Specify the software+hardware platform;
3. Define the specification;
4. Obtain embedded code enforcing the specification for the continuous dynamics on the given software+hardware platform.

A discrete-time control system is defined by a map:

$$f : M \times U \rightarrow M \quad (1.1)$$

describing the state $f(x, u) \in M$ resulting from applying input $u \in U$ at the state $x \in M$. In many situations the resulting state $f(x, u)$ cannot be observed directly but rather through an output map:

$$r : M \rightarrow P \quad (1.2)$$

transforming states $x \in M$ into outputs $r(x) \in P$.

Define bisimulation.

Theorem 1.1 Let $M \times U \xrightarrow{f} M \xrightarrow{r} P$ be a discrete-time control system satisfying any of the following assumptions:

1. f is linear and controllable;
2. f and r are linear and (f, r) is output controllable;
3. f is feedback linearizable or differentially (difference) flat.

Then, any finite partition of the output space induces a finite bisimilar quotient.

Theorem 1.1 Let $M \times U \xrightarrow{f} M \xrightarrow{r} P$ be a discrete-time control system satisfying any of the following assumptions:

1. f is linear and controllable;
2. f and r are linear and (f, r) is output controllable;
3. f is feedback linearizable or differentially (difference) flat.

Then, any finite partition of the output space induces a finite bisimilar quotient.

In general it is difficult to obtain discrete-time models.

State partitions suffer from an intrinsic lack of robustness.

Start with the double integrator:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = u$$

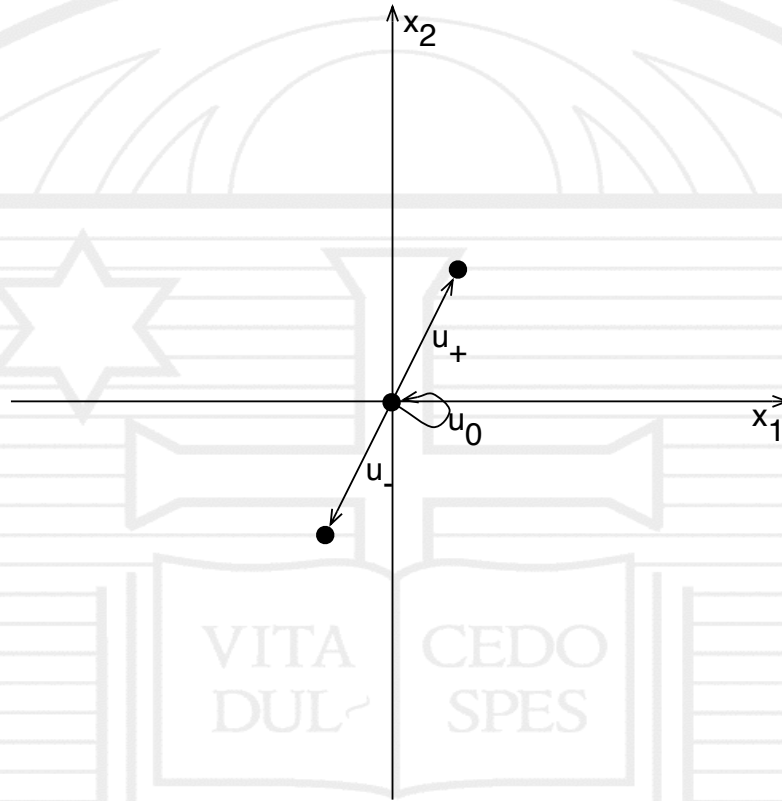
and chose a finite set of input trajectories (or control quanta in the quantized control systems^a terminology):

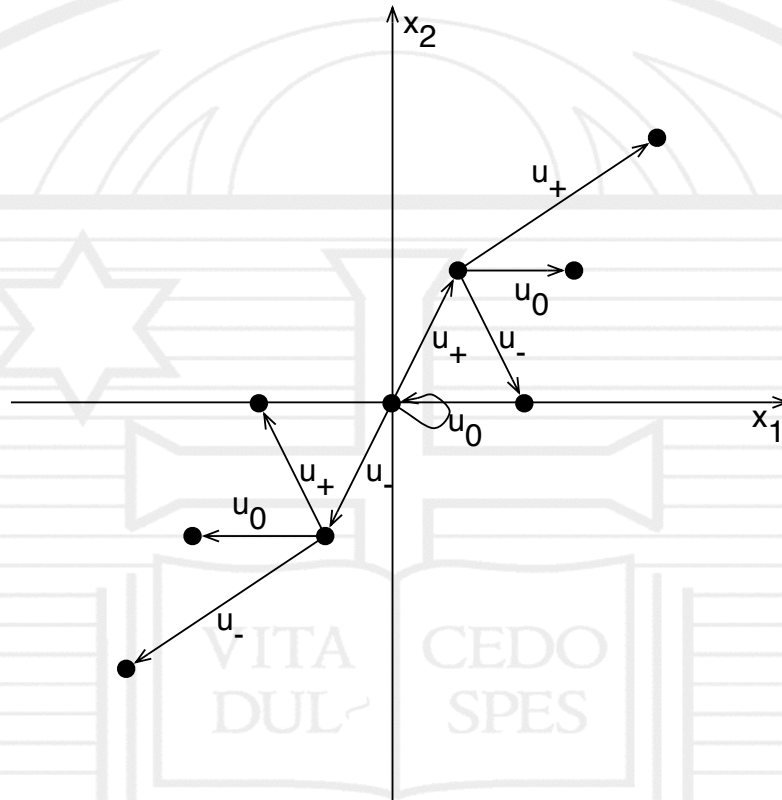
$$\mathbf{u}_- : [0, 1] \rightarrow \mathbb{R} \quad \mathbf{u}_0 : [0, 1] \rightarrow \mathbb{R} \quad \mathbf{u}_+ : [0, 1] \rightarrow \mathbb{R}$$

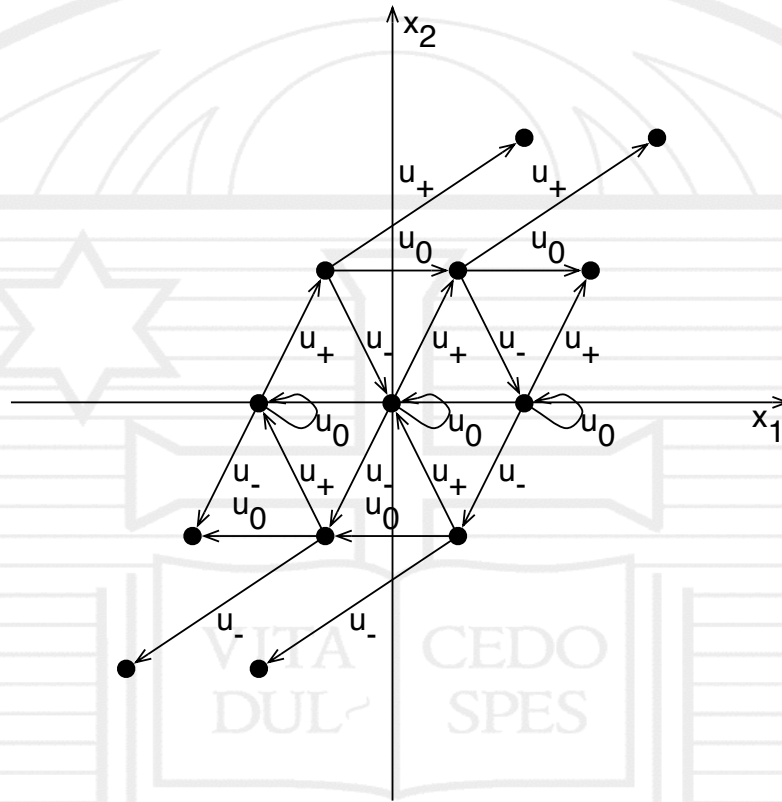
$$\mathbf{u}_-(t) = -1 \quad \mathbf{u}_0(t) = 0 \quad \mathbf{u}_+(t) = 1$$

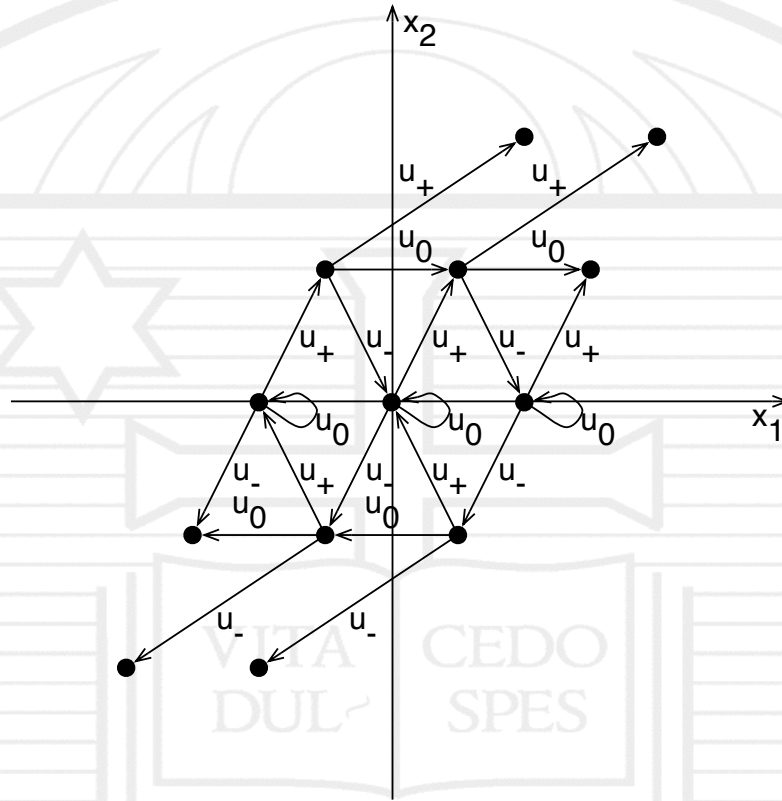
Starting from the origin and applying \mathbf{u}_- , \mathbf{u}_0 and \mathbf{u}_+ we obtain a symbolic description of *some* behaviors of the double integrator.

^aA. Bicchi, A. Marigo, and B. Piccoli. On the reachability of quantized control systems. *IEEE Transactions on Automatic Control*, 4(47):546-563, April 2002.









The question I will try to address is:

When is this symbolic subsystem representative of the whole system behavior?

Define symbolic subsystem.

Once we have a symbolic subsystem T of the transition system T_Σ associated with a control system Σ , we can regard controller synthesis for Σ as a supervisory control problem:



Define symbolic subsystem.

Once we have a symbolic subsystem T of the transition system T_Σ associated with a control system Σ , we can regard controller synthesis for Σ as a supervisory control problem:

Specification: $S \subseteq O^*$;

Assumptions: $T \prec T_\Sigma$ and $\exists T_c$ such that $L(T_c \parallel T) \subseteq S$;

Conclusion: $T'_c = T_c \parallel T$ satisfies $L(T'_c \parallel T_\Sigma) \subseteq S$.

Specification: T_S ;

Assumptions: $T \prec T_\Sigma$, $\exists T_c$ such that $T_c \parallel T \prec T_S$ and ...

Conclusion: $T'_c = T_c \parallel T$ satisfies $T'_c \parallel T_\Sigma \prec T_S$.

Define symbolic subsystem.

Once we have a symbolic subsystem T of the transition system T_Σ associated with a control system Σ , we can regard controller synthesis for Σ as a supervisory control problem:

Specification: $S \subseteq O^*$;

Assumptions: $T \prec T_\Sigma$ and $\exists T_c$ such that $L(T_c \parallel T) \subseteq S$;

Conclusion: $T'_c = T_c \parallel T$ satisfies $L(T'_c \parallel T_\Sigma) \subseteq S$.

Specification: T_S ;

Assumptions: $T \prec T_\Sigma$, $\exists T_c$ such that $T_c \parallel T \prec T_S$ and ...

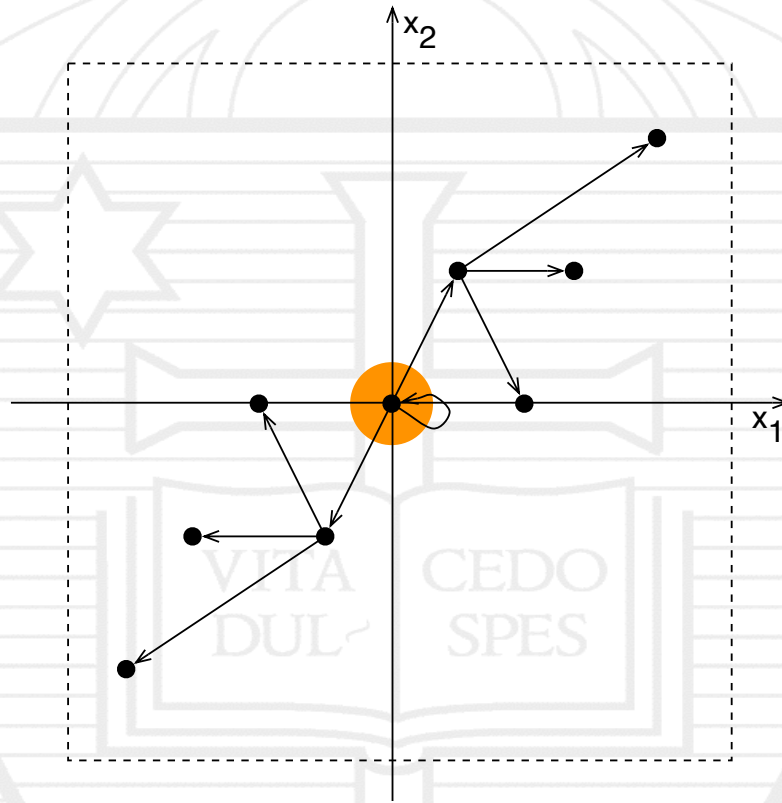
Conclusion: $T'_c = T_c \parallel T$ satisfies $T'_c \parallel T_\Sigma \prec T_S$.

There are, however, two difficulties with the above principles:

- 1) They only provide sufficient conditions;
- 2) Supervisor T'_c can only be used to control T_Σ for initial conditions in $Q \subset \mathbb{R}^n$.

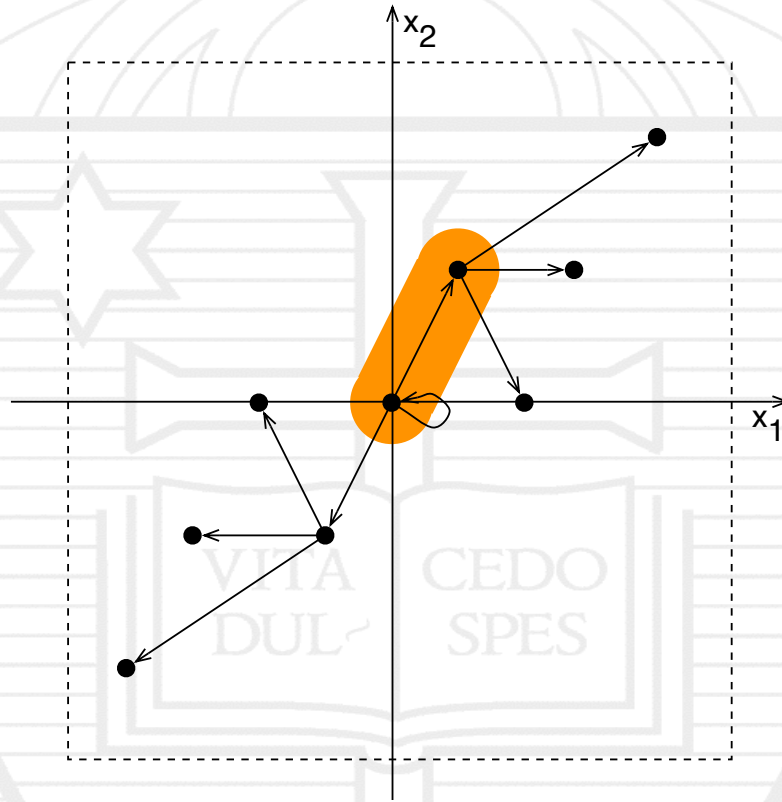
Symbolic models: Continuous-time

The second problem can be solved if we work on a compact and if we can "robustify" T .



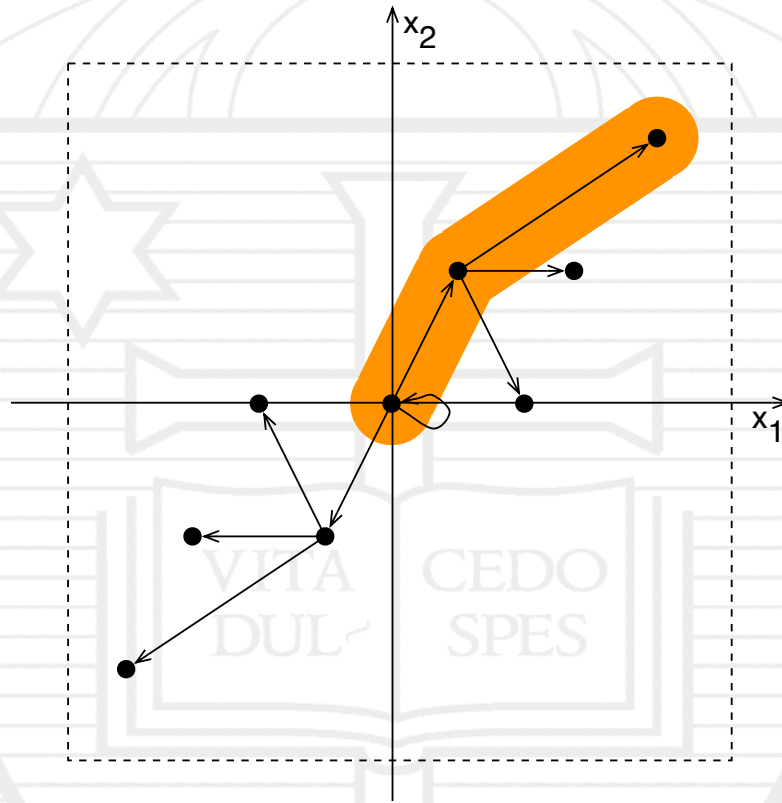
Symbolic models: Continuous-time

The second problem can be solved if we work on a compact and if we can "robustify" T .



Symbolic models: Continuous-time

The second problem can be solved if we work on a compact and if we can "robustify" T .



Let us call a linear control system $\dot{x} = Ax + Bu$ feedback stabilizable if there exists a linear feedback $u = Kx$ such that $\dot{x} = Ax + BKx$ is stable. In this case there exists also a Lyapunov function V satisfying $\dot{V} = \frac{\partial V}{\partial x}(Ax + BKx) \leq 0$.



Let us call a linear control system $\dot{x} = Ax + Bu$ feedback stabilizable if there exists a linear feedback $u = Kx$ such that $\dot{x} = Ax + BKx$ is stable. In this case there exists also a Lyapunov function V satisfying $\dot{V} = \frac{\partial V}{\partial x}(Ax + BKx) \leq 0$.

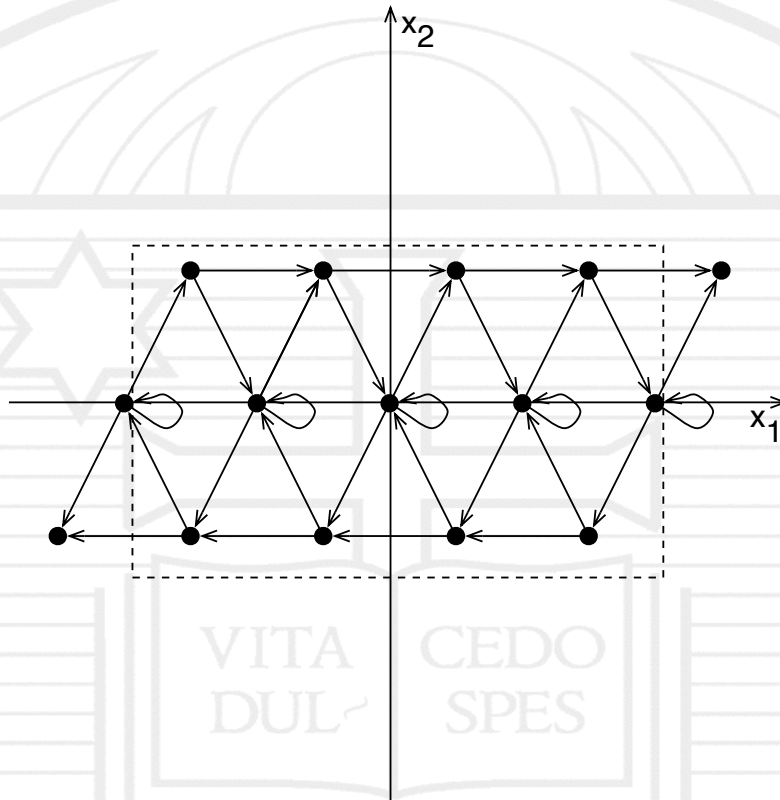
Theorem 1.2 *Let $T_\Sigma = (Q_\Sigma, Q_\Sigma^0, \longrightarrow_\Sigma, O_\Sigma, H_\Sigma)$ be the transition system associated with a linear control system Σ . If Σ is feedback stabilizable, then for:*

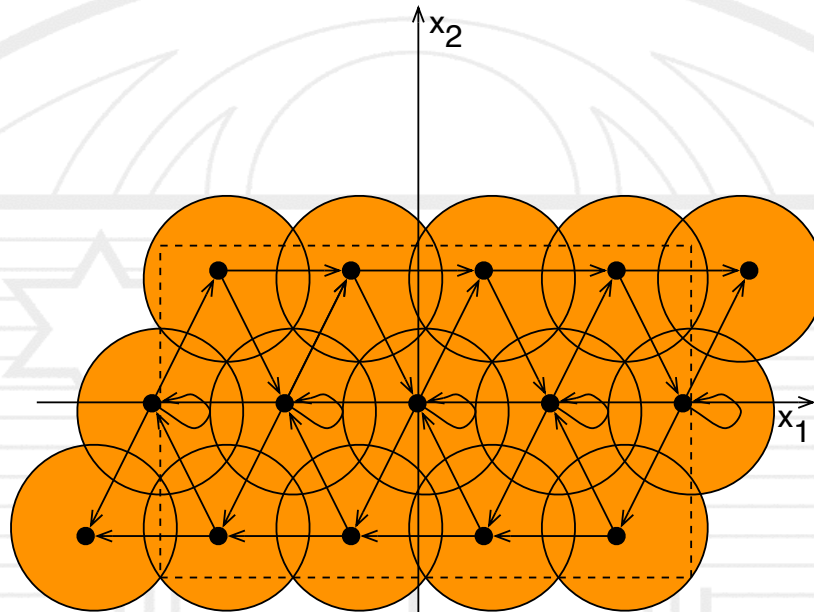
1. *any stabilizing linear controller K and corresponding Lyapunov function V ;*
2. *any symbolic sub-system $T = (Q, Q^0, \longrightarrow, O, H)$ of T_Σ ;*
3. *any bounded subset Q'_Σ of Q_Σ containing Q ,*

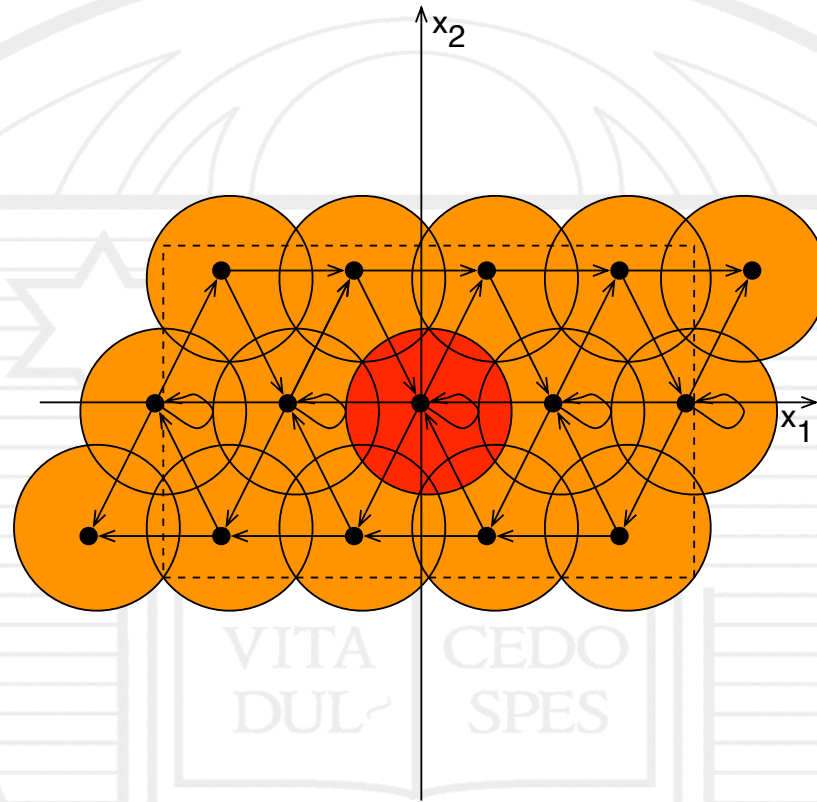
there exists a real number $\mu \in \mathbb{R}$ such that $R \subseteq Q \times Q'_\Sigma$ defined by:

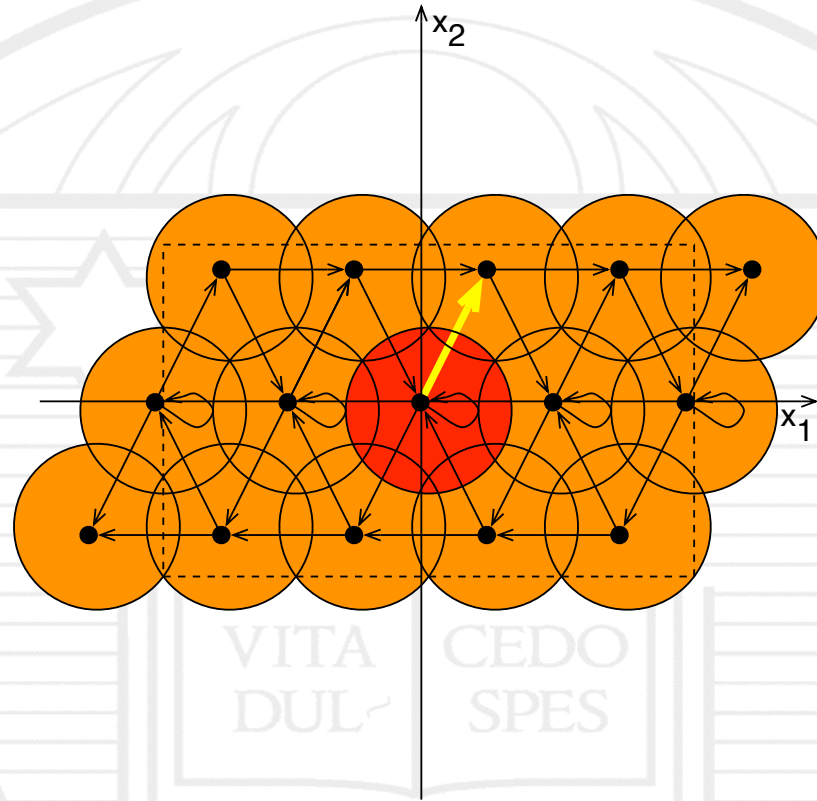
$$(q, x) \in R \text{ when } V(x - q) \leq \mu$$

is a simulation relation from T to T'_Σ satisfying $R(Q) = Q'_\Sigma$, where $T'_\Sigma = (Q'_\Sigma, Q'_\Sigma, \longrightarrow, O, H'_\Sigma)$ with $q \in H'_\Sigma(x)$ when $V(x - q) \leq \mu$.

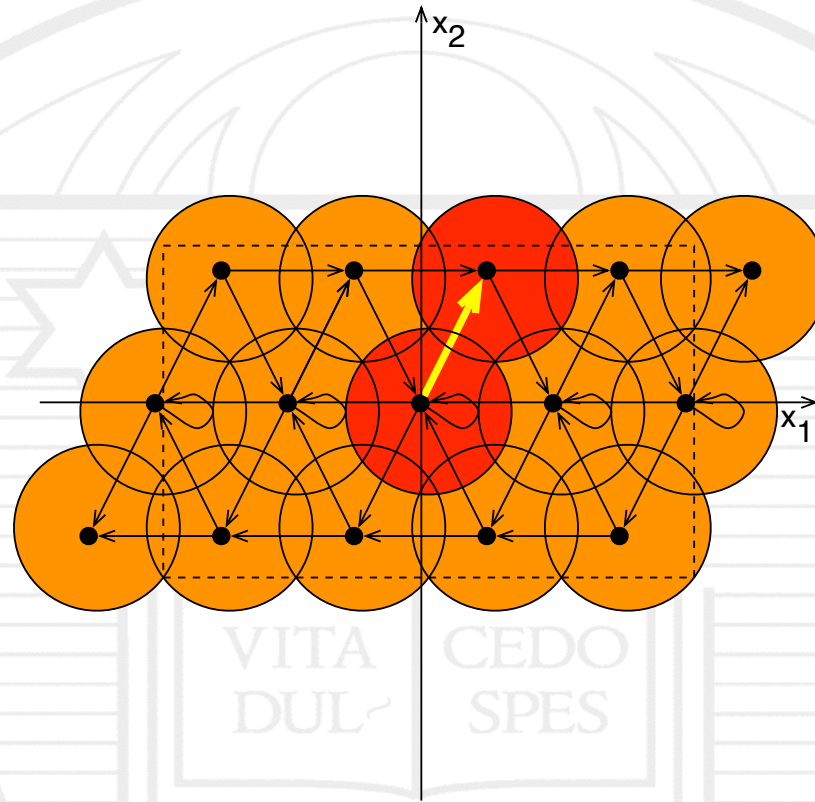


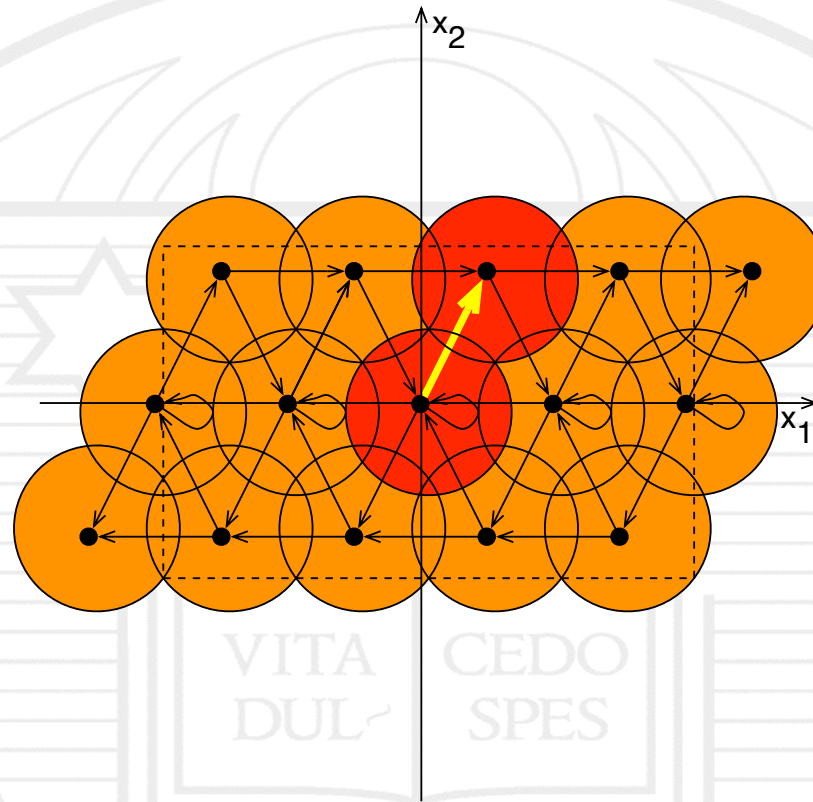






Symbolic models: Continuous-time





We can do even better if we strengthen stability to an asymptotic stability.
 In this case V satisfies $\dot{V} = \frac{\partial V}{\partial x}(Ax + BKx) < -\alpha V$ for $\alpha > 0$ and this implies:

$$V(t) < V(0)e^{-\alpha t}$$

Asymptotic stability allows for a decrease in the "uncertainty" measured by the Lyapunov function. This can be captured by working with a countable version of T :

Definition 1.3 Let T_Σ be the transition system induced by a linear control system Σ . For any sub-system $T = (Q, Q^0, \longrightarrow, O, H)$ of T_Σ , $T_{\mathbb{N}_0}$ denotes the transition system defined by $T_{\mathbb{N}_0} = (Q \times \mathbb{N}_0, Q^0 \times \mathbb{N}_0, \longrightarrow_{\mathbb{N}_0}, O \times \mathbb{N}_0, H_{\mathbb{N}_0})$ where $(q, n) \longrightarrow_{\mathbb{N}_0} (q', n')$ if $q \longrightarrow q'$ in T and $n' = n + 1$, and $H_{\mathbb{N}_0}(q, n) = \{(q, n)\}$.

Defining $\sigma = e^{-\alpha\tau} < 1$ where τ is the duration of the input trajectories used to construct symbolic subsystem T we have the following "graded" version of Theorem 1.2.

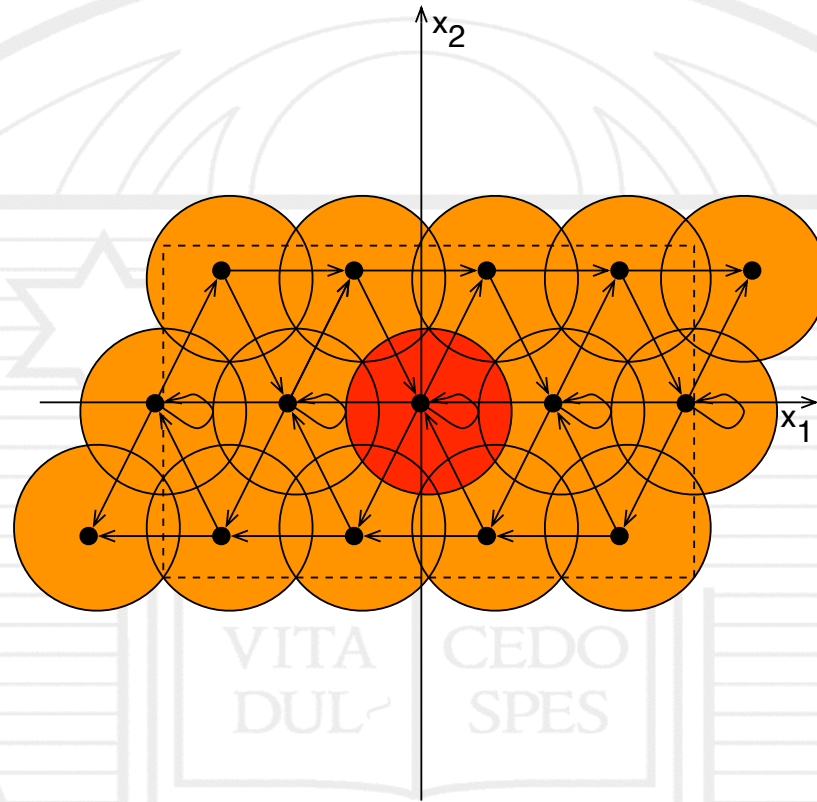
Theorem 1.4 Let $T_\Sigma = (Q_\Sigma, Q_\Sigma^0, \longrightarrow_\Sigma, O_\Sigma, H_\Sigma)$ be the transition system associated with a linear control system Σ . If Σ is asymptotically feedback stabilizable, then for:

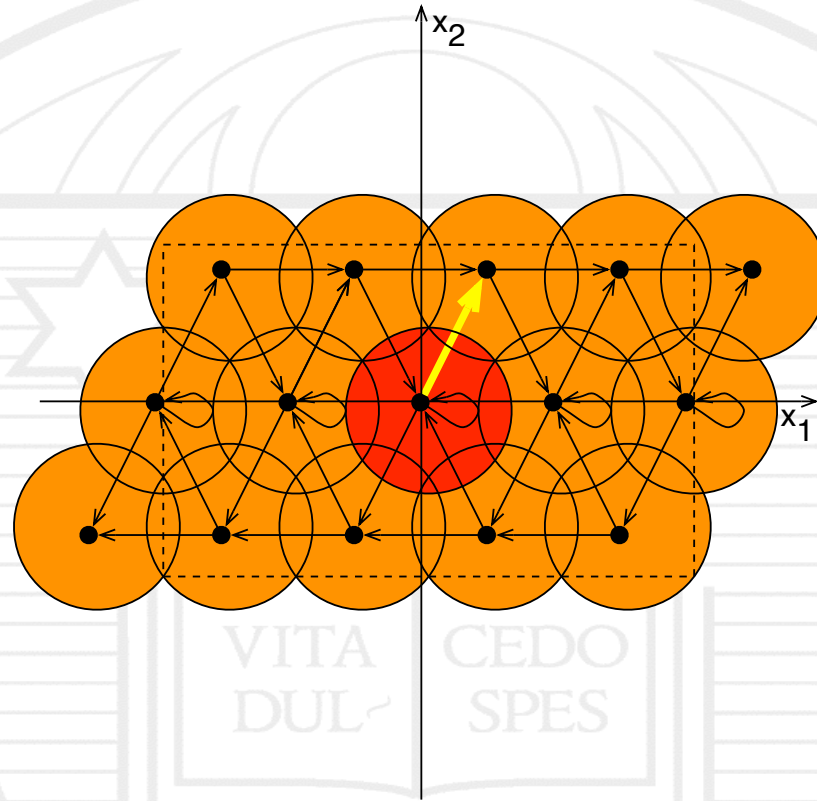
1. any asymptotically stabilizing linear controller K and corresponding Lyapunov function V satisfying $\dot{V} \leq -\alpha V$;
2. any symbolic sub-system $T = (Q, Q^0, \longrightarrow, O, H)$ of T_Σ ;
3. any bounded subset Q'_Σ of Q_Σ containing Q ,

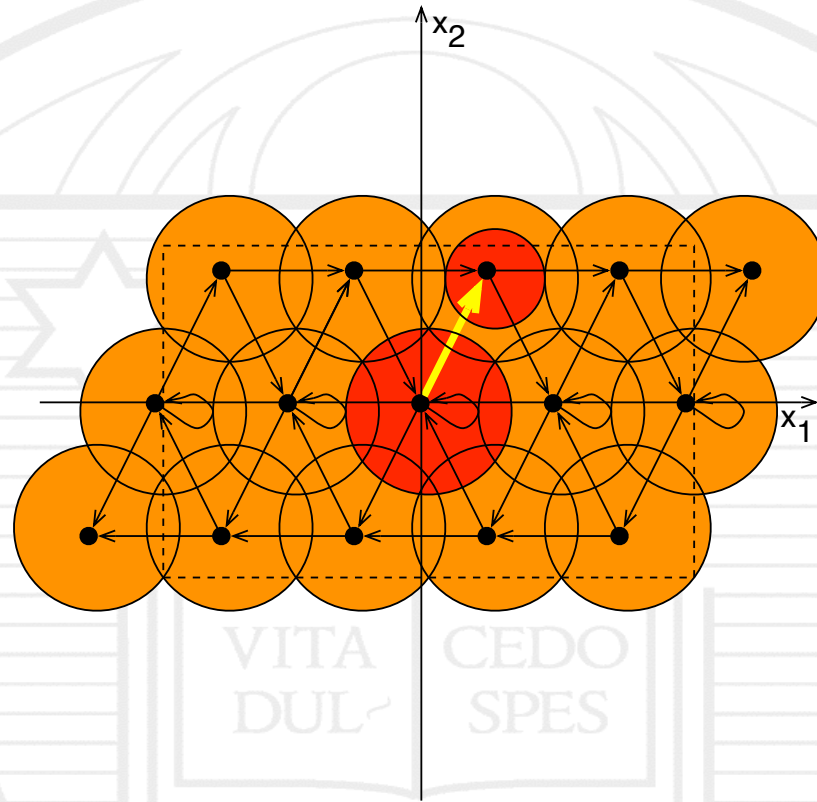
there exists a real number $\mu \in \mathbb{R}$ such that $R \subseteq (Q \times \mathbb{N}_0) \times Q_\Sigma$ defined by:

$$((q, n), x) \in R \text{ when } V(x - q) \leq \mu\sigma^n$$

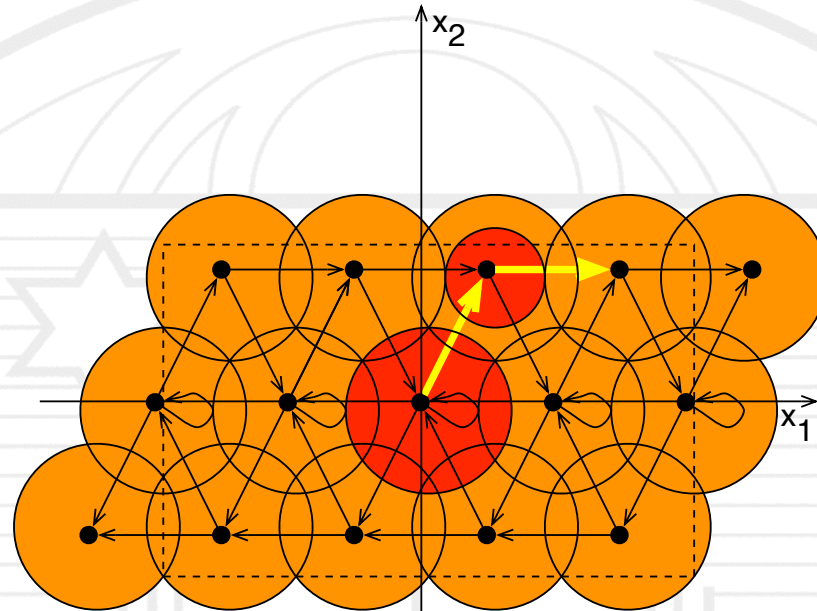
is a simulation relation from $T_{\mathbb{N}_0}$ to T'_Σ satisfying $R(Q) = Q'_\Sigma$, where $T'_\Sigma = (Q'_\Sigma, Q'_\Sigma, \longrightarrow_\Sigma, O \times \mathbb{N}_0, H'_\Sigma)$ with $(y, n) \in H'_\Sigma(x)$ when $V(x - y) \leq \mu\sigma^n$.

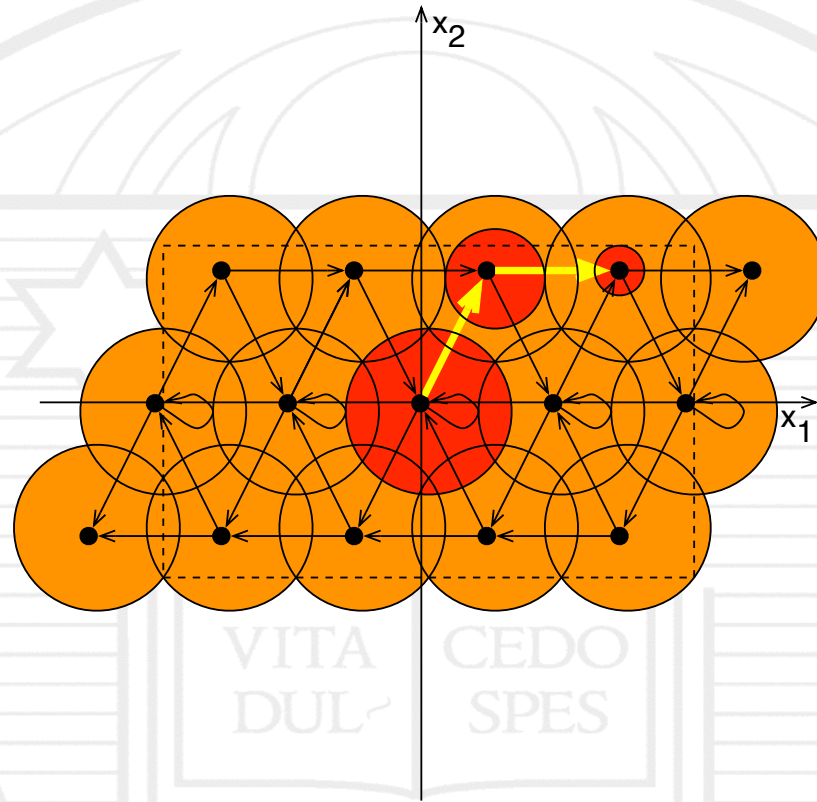




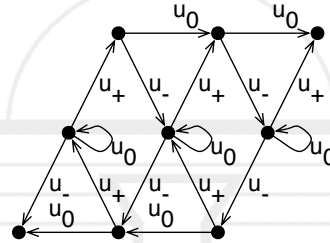


Symbolic models: Continuous-time



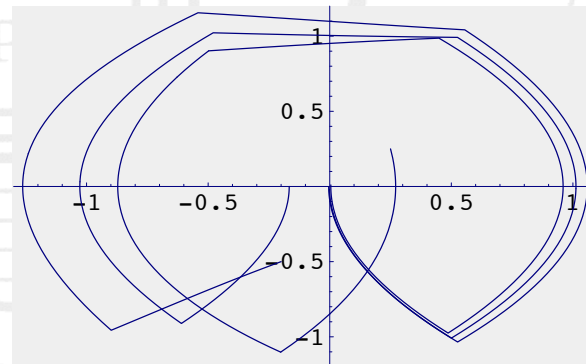
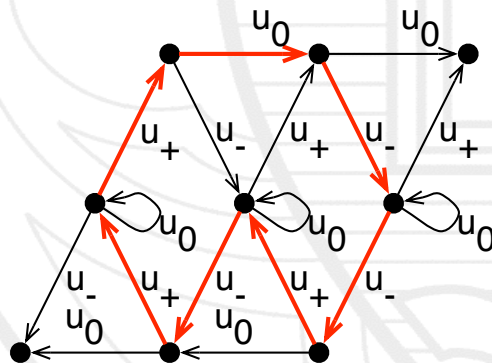


Let us reconsider the double integrator Σ and its symbolic subsystem:

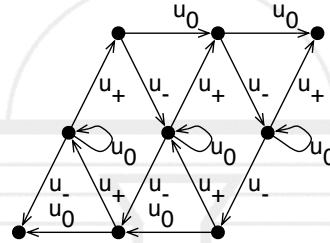


Σ can be stabilized by $u = -x_1 - x_2$ and control Lyapunov function $V = x_1^2 + x_1x_2 + x_2^2$ satisfies $\dot{V} = -V$.

Enforcing the string $u_-u_+u_+u_0u_-u_-u_+$ results in:

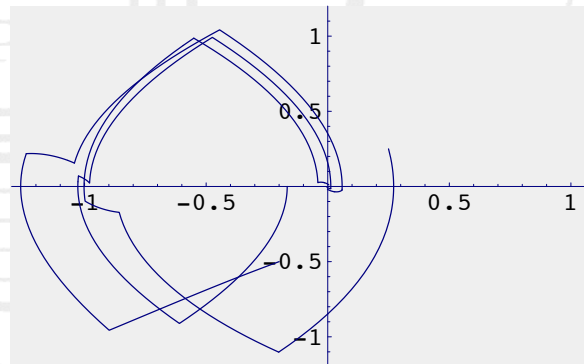
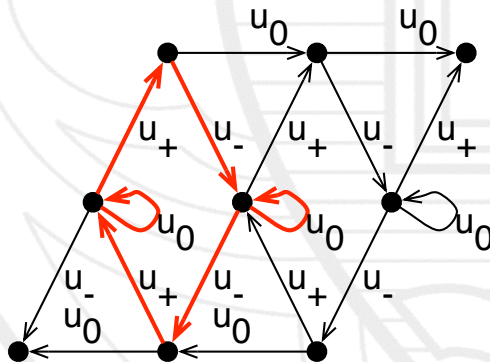


Let us reconsider the double integrator Σ and its symbolic subsystem:



Σ can be stabilized by $u = -x_1 - x_2$ and control Lyapunov function $V = x_1^2 + x_1x_2 + x_2^2$ satisfies $\dot{V} = -V$.

Enforcing the string $u_-u_+u_0u_+u_-u_0u_0$ results in:



Linearity + stability \Rightarrow control based on symbolic subsystems is possible, up to a certain resolution;



Linearity + stability \Rightarrow control based on symbolic subsystems is possible, up to a certain resolution;

Nonlinear systems? Incremental stability and Input-to-State stability;



Linearity + stability \Rightarrow control based on symbolic subsystems is possible, up to a certain resolution;

Nonlinear systems? Incremental stability and Input-to-State stability;

How to choose the symbolic subsystem? Under study, for linear systems it suffices to choose u_- , u_0 and u_+ per input;

Linearity + stability \Rightarrow control based on symbolic subsystems is possible, up to a certain resolution;

Nonlinear systems? Incremental stability and Input-to-State stability;

How to choose the symbolic subsystem? Under study, for linear systems it suffices to choose u_- , u_0 and u_+ per input;

Bisimulation? Under study, possible using the "right" notion of "approximate" or "tolerance" bisimulation and for certain classes of input trajectories. Related to work by^a Girard and Pappas, and^b Caspi and Benveniste.

^aApproximation metrics for discrete and continuous systems. *Technical Report MS-CIS-05-10, Dept. of CIS, University of Pennsylvania*, May 2005.

^bToward an Approximation Theory for Computerised Control. *2nd International Workshop on Embedded Software*, 2002.

Simple examples of safety controllers

Controllable linear model obtained by time-discretization of a double integrator model of each wheel. *Note that the unicycle kinematic model can be brought to this form by dynamic feedback linearization.*

Observation map defines a 10×10 grid on the state space.

