www.unisi.it

1st HYCON PhD School on Hybrid Systems

www.ist-hycon.org

Hybrid Systems: Challenges and Opportunities

Alberto Sangiovanni Vincentelli

University of California, Berkeley, USA
and PARADES – Rome, Italy

alberto@eecs.berkeley.edu

HYSCOM
IEEE CSS Technical Committee on Hybrid Systems

Information Society Technologies

Siena, July 19-22, 2005 - Rectorate of the University of Siena

Hybrid Systems: Challenges and Opportunities

Alberto Sangiovanni-Vincentelli
The Edgar L. and Harold H. Buttner Chair of EECS
University of California at Berkeley

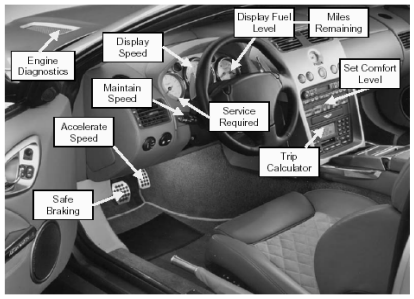
Co-Founder, Chief Technology Advisor and Member of the Board
Cadence Design Systems

Founder and Scientific Director
PARADES (Cadence and ST)
Roma

Outline

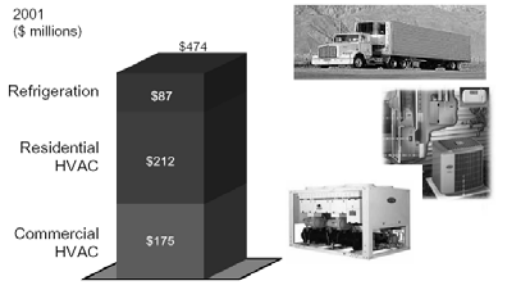
- Motivation
- Embedded Software and Hybrid Systems
- Hybrid Systems and Verification Problems

FUNCTION OF CONTROLS
Typical minivan application

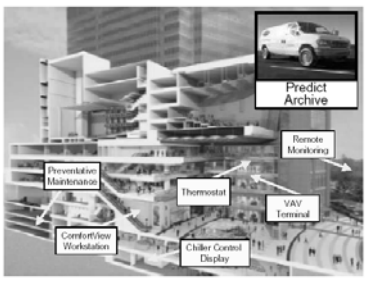


Configure
Sense
Actuate
Regulate
Display
Trend
Diagnose
Predict
Archive

CARRIER CONTROLS BUSINESS
Market segments



FUNCTION OF CONTROLS
Typical commercial HVAC application

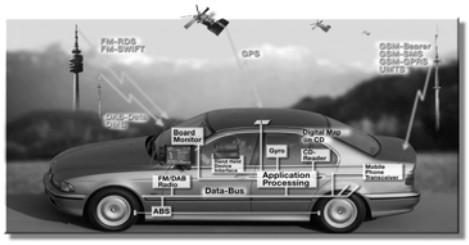


Configure
Sense
Actuate
Regulate
Display
Trend
Diagnose
Predict
Archive

Common Situation in Industry

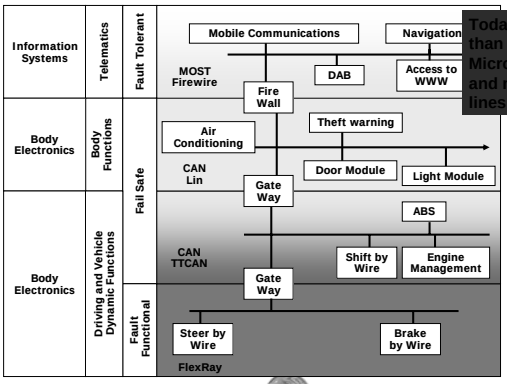
- **Different hardware devices and architectures**
- **Increased complexity**
- **Non-standard tools and design processes**
- **Redundant development efforts**
- **Increased R&D and sustaining costs**
- **Lack of standardization results in greater quality risks**
- **Customer confusion**

Automotive Supply Chain: Car Manufacturers



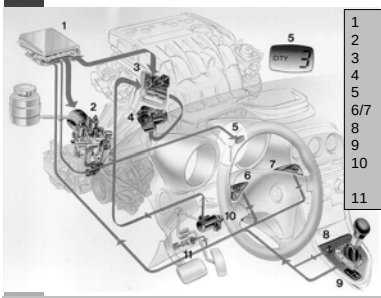
- Product Specification & Architecture Definition (e.g., determination of Protocols and Communication standards)
- System Partitioning and Subsystem Specification
- Critical Software Development
- System Integration

Electronics for the Car: A Distributed System



Today, more than 80 Microprocessors and millions of lines of code

Automotive Supply Chain: Tier 1 Subsystem Providers



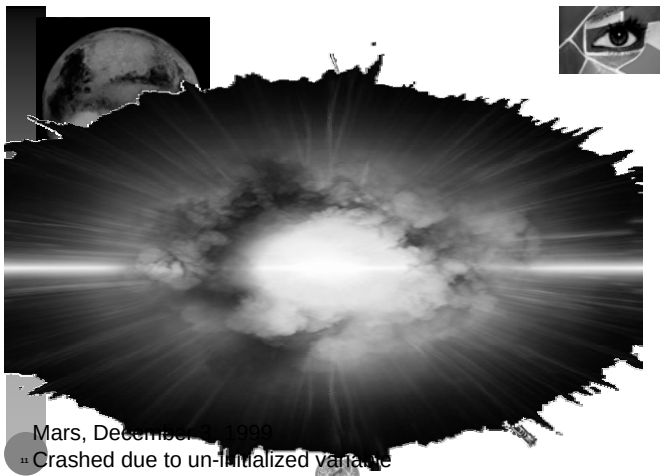
1	Transmission ECU
2	Actuation group
3	Engine ECU
4	DBW
5	Active shift display
6/7	Up/Down buttons
8	City mode button
9	Up/Down lever
10	Accelerator pedal position sensor
11	Brake switch

BOSCH

MAGNETI MARELLI

- Subsystem Partitioning
- Subsystem Integration
- Software Design: Control Algorithms, Data Processing
- Physical Implementation and Production

How Safe is Our Real-Time Software?



Mars, December 3, 1999
Crashed due to un-initialized variable

$4 billion development effort
40-50% system integration & validation cost

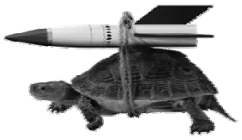
Complexity, Quality, & Time To Market today

	PWT UNIT	BODY GATEWAY	INSTRUMENT CLUSTER	TELEMATIC UNIT
Memory	256 Kb	128 Kb	184 Kb	8 Mb
Lines Of Code	50.000	30.000	45.000	300.000
Productivity	6 Lines/Day	10 Lines/Day	6 Lines/Day	10 Lines/Day*
Residual Defect Rate @ End Of Dev	3000 Ppm	2500 ppm	2000ppm	1000 ppm
Changing Rate	3 Years	2 Years	1 Year	< 1 Year
Dev. Effort	40 Man-yr	12 Man-yr	30 Man-yr	200 Man-yr
Validation Time	5 Months	1 Month	2 Months	2 Months
Time To Market	24 Months	18 Months	12 Months	< 12 Months

*** C++ CODE**

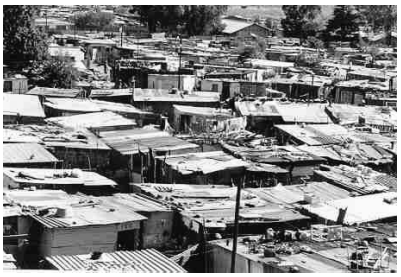
FABIO ROMEO, Magneti-Marelli
DAC, Las Vegas, June 20th, 2001

What About Real Time?



"Make it faster!"

Software Architecture Today



Poor common infrastructure. Weak specialization of functions. Poor resource management. Poor planning.

Design "Practice"

Design Science: Build upon Solid Foundations

Mission of the School

To provide background for research on the design issues necessary for supporting next-generation embedded controllers.

- Model-based design
- Tool-supported methodologies

For

- Real-time
- Fault-tolerant
- Robust
- Secure
- Heterogeneous
- Distributed Software

We are on the line to create a "new systems science" that is at once computational and physical.

Embedded Systems

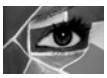
- Computational
  - but not first-and-foremost a computer
- Integral with physical processes
  - sensors, actuators
- Reactive
  - at the speed of the environment
- Heterogeneous
  - hardware/software, mixed architectures
- Networked
  - shared, adaptive

Source: Edward A. Lee

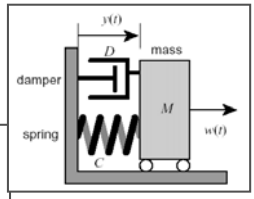
The Long-term Fundamental Research Agenda

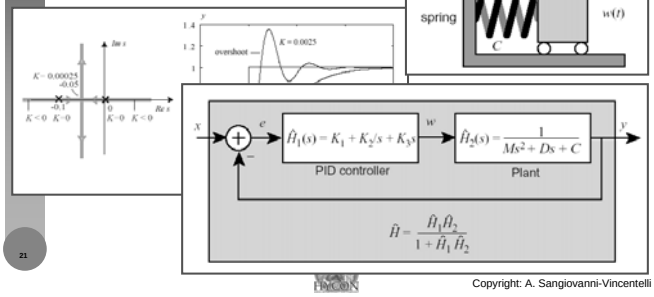
- To create a modern ***computational systems science and systems design practice*** with
  - Concurrency
  - Composability
  - Time
  - Hierarchy
  - Heterogeneity
  - Resource constraints
  - Verifiability
  - Understandability

A Traditional Systems Science – Feedback Control Systems

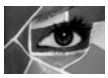


- **Models of continuous-time dynamics**
- **Stability analysis**
- **But not accurate for software controllers**

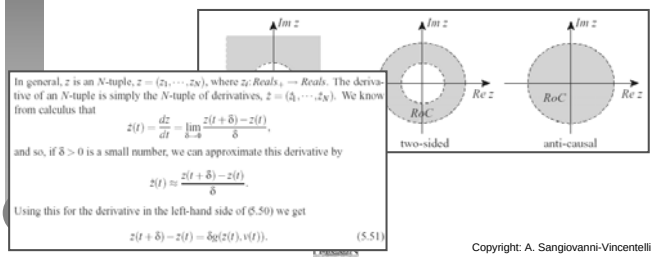




Discretized Model – A Step Towards Software



- **Numerical integration techniques provided ways to get from the continuous idealizations to computable algorithms.**
- **Discrete-time signal processing techniques offer the same sophisticated stability analysis as continuous-time methods.**
- **But it's *still* not accurate for software controllers**



In general, z is an N-tuple, $z = (z_1, \cdots, z_N)$, where z_i: $Reals_+ \to Reals$. The derivative of an N-tuple is simply the N-tuple of derivatives, $\dot{z} = (\dot{z}_1, \cdots, \dot{z}_N)$. We know from calculus that

$$\dot{z}(t) = \frac{dz}{dt} = \lim_{\delta \to 0} \frac{z(t+\delta) - z(t)}{\delta},$$

and so, if $\delta > 0$ is a small number, we can approximate this derivative by

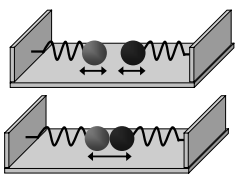
$$\dot{z}(t) \approx \frac{z(t+\delta) - z(t)}{\delta}.$$

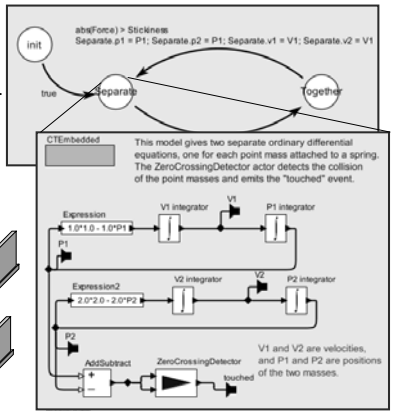
Using this for the derivative in the left-hand side of (5.50) we get

$$z(t+\delta) - z(t) = \delta g(z(t), v(t)). \qquad (5.51)$$

Hybrid Systems – Reconciliation of Continuous & Discrete

But it's *still* not accurate for software controllers

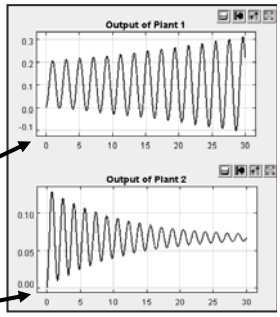


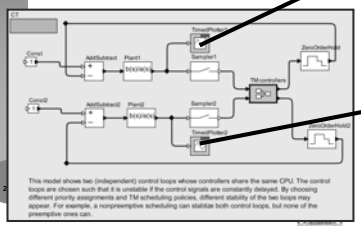


Timing in Software is More Complex Than What the Theory Deals With

An example models two controllers sharing a CPU under an RTOS. Under preemptive multitasking, only one can be made stable (depending on the relative priorities). Under non-preemptive multitasking, both can be made stable.

Where is the theory for this?

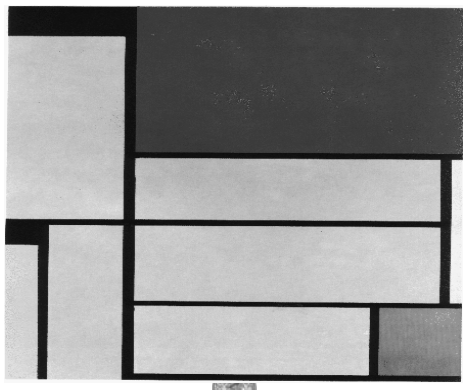




Foundational Theory Research …

- The science of computation has systematically abstracted away the physical world. The science of physical systems has systematically ignored computational limitations.
  Embedded software systems, however, engage the physical world in a computational manner.
- It is time to construct a Hybrid Systems Science that is simultaneously computational and physical.
  Time, concurrency, robustness, continuums, and resource management must be remarried to computation.

Abstraction and Refinement

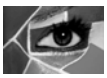


What is a hybrid system?

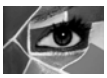
- Combination of discrete and continuous time with a prescribed hierarchy

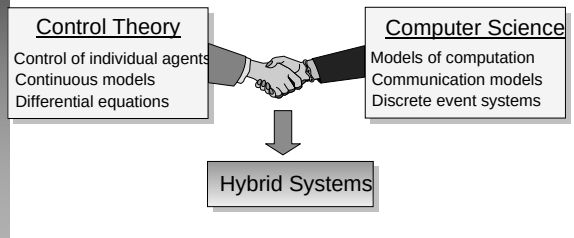
- Models with "heterogeneous components"

Why Hybrid Systems?

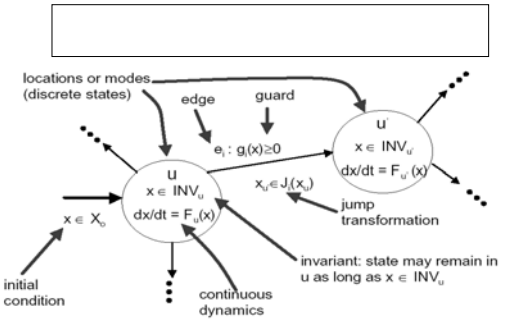


- Modeling abstraction of
  - Continuous systems with phased operation (e.g. walking robots, mechanical systems with collisions, circuits with diodes)
  - Continuous systems controlled by discrete inputs (e.g. switches, valves, digital computers)
  - Coordinating processes (multi-agent systems)
- Important in applications
  - Hardware verification/CAD, real time software
  - Manufacturing, chemical process control,
  - communication networks, multimedia
- Large scale, multi-agent systems
  - Automated Highway Systems (AHS)
  - Air Traffic Management Systems (ATM)
  - Uninhabited Aerial Vehicles (UAV), Power Networks



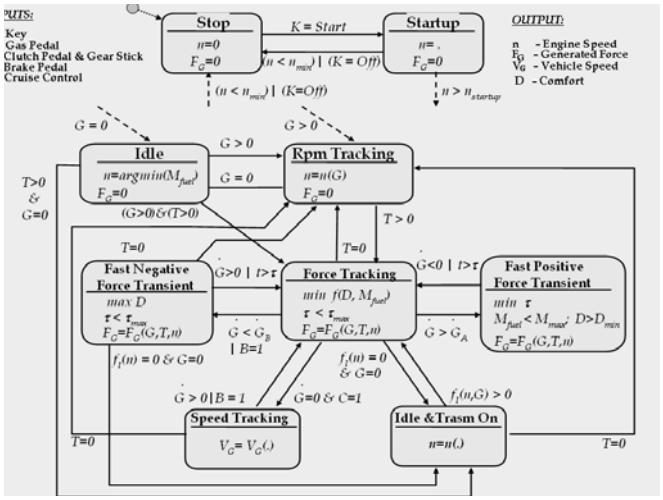


Hybrid Automata (B. De Schutter)

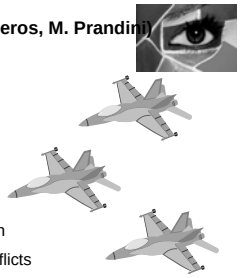


Problem Formulation: Engine Control

- Engine Control Problem subdivided in “modes”, i.e., a mode corresponds to a particular set of external inputs (e.g., position of accelerator pedal, gear)
- Each mode corresponds to different plant models and cost function for control
- Important to decouple mode diagram from other considerations, e.g., control algorithm, requirements



Air Traffic Management (J. Lygeros, M. Prandini)

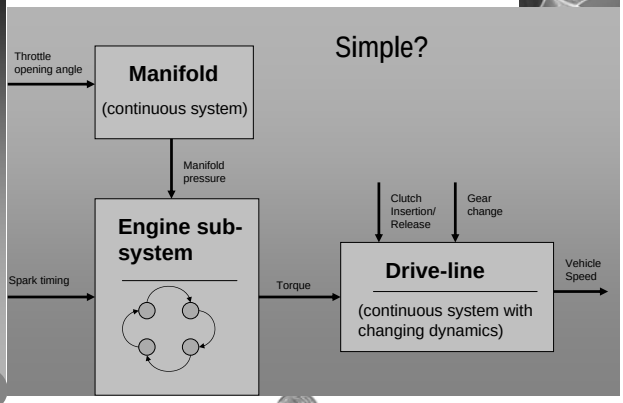


- Large number of semiautonomous agents
- Coordinate to
  - Make efficient use of common resource
  - Achieve a common goal
- Individual agents have various modes of operation
- Agents optimize locally, coordinate to resolve conflicts
- System architecture is hierarchical and distributed
- Safety critical systems

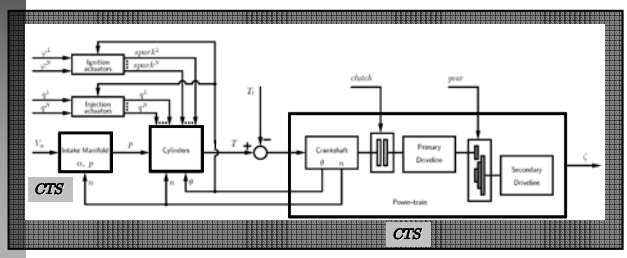
Challenge: Develop models, analysis, and synthesis tools for designing and verifying the safety of multi-agent systems

Stochastic Hybrid Systems are an essential modeling tool!

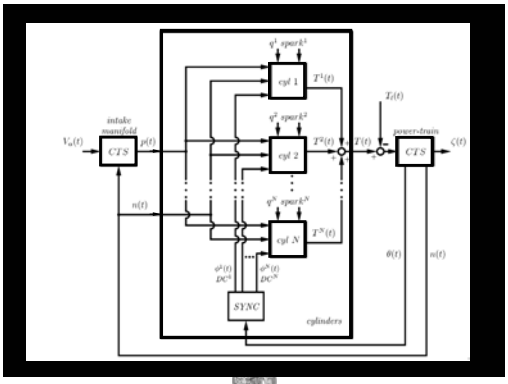
Model of Power-train (Application Day)



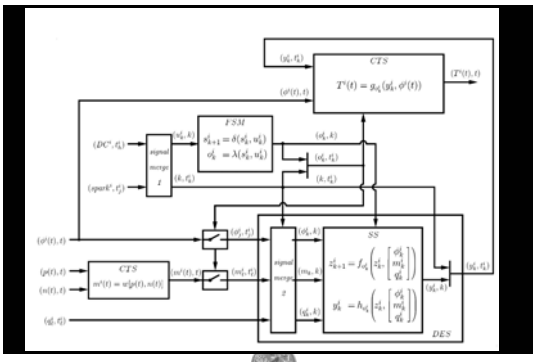
Engine and Drive-line



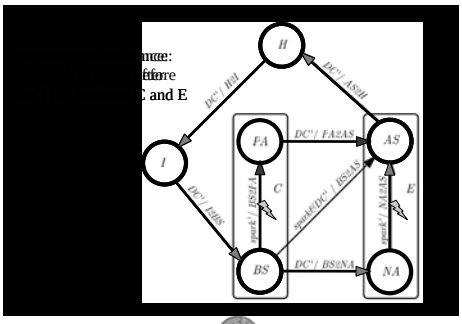
Engine and Drive-line



Single Cylinder Hybrid Model



FSM for a single cylinder



Hybrid Model vs Mean-Value Model: Control of Hybrid Systems (B. Piccoli)

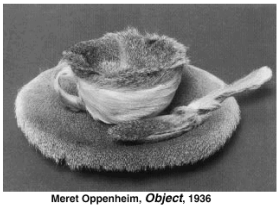
- Mean-Value Model: accurate over a longer time window
  - regulation control problems
  - low performance transient problems
- Hybrid Model: cycle accurate
  - transient control problems
  - stability of delay-sensitive control algorithms
  - high performance control algorithms

The Hybrid System Nightmare

“Femme se coiffant”
Pablo Ruiz Picasso
1940

Validating Designs (G. Pappas)

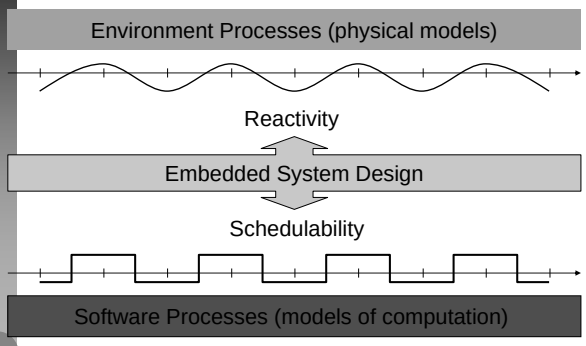
- By construction
  - property is inherent.
- By verification
  - property is provable syntactically.
- By simulation
  - check behavior for all inputs.
- By intuition
  - property is true. I just know it is.
- By assertion
  - property is true. Wanna make something of it?
- By intimidation
  - Don’t even try to doubt whether it is true



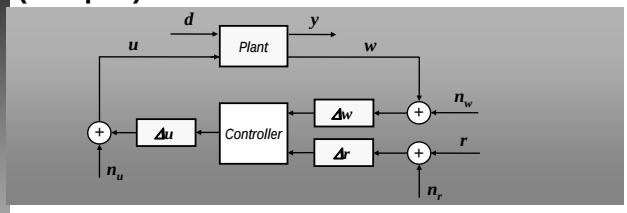
Meret Oppenheim, ***Object,*** 1936

It is generally better to be higher in this list

A Hybrid Systems Science



Effects of controller implementation in the controlled plant performance (Palopoli)

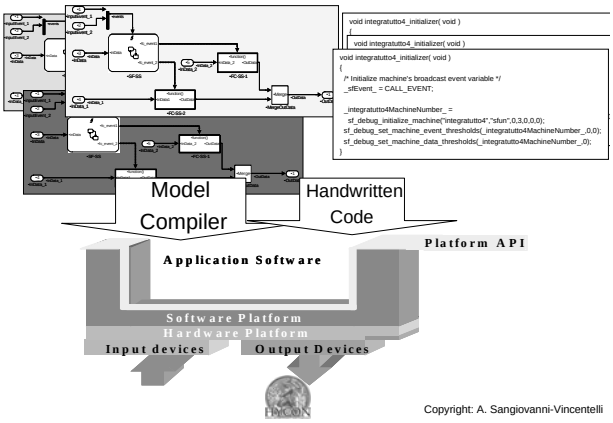


- modeling of implementation non-idealities:
  - $\Delta u, \Delta r, \Delta w$: time-domain perturbations
    - control loop delays, sample & hold , etc.
  - n_u, n_r, n_w :value-domain perturbations
    - quantization error, computation imprecision, etc.

Control Algorithm Implementation

- The control algorithms are mapped to the target platform to achieve the best performance/cost trade-off
- In most cases the platform can accommodate in software the control algorithms, if not:
  - New platform services or
  - New hardware components or
  - New control algorithms
- Float to fixed transformation

Synthesis of Control Algorithms



Conclusions

- New System Theory: marrying physical with abstract (control vs. computer science)
- Hybrid Systems powerful mechanism for the design of embedded systems
- Hybrid Systems are quite complex: need a different theory
- Hybrid Systems are increasingly finding important applications