Cyber Physical Systems: Next Generation of Embedded Systems

Janos Sztipanovits
ISIS, Vanderbilt University

27 September, 2010
Outline

- Cyber Physical Systems (CPS)
- Integration of CPS: Dealing with Heterogeneity
- Example: Passivity-based Design
CPS is about Engineered Systems

<table>
<thead>
<tr>
<th>Sectors</th>
<th>Opportunities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health and Biomedical</td>
<td>In-home healthcare delivery. More capable biomedical devices for measuring health. New prosthetics for use within and outside the body. Networked biomedical systems that increase automation and extend the biomedical device beyond the body.</td>
</tr>
<tr>
<td>Smart Grid</td>
<td>Highway systems that allow traffic to become denser while also operating more safely. A national power grid that is more reliable and efficient.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sectors</th>
<th>Goals</th>
</tr>
</thead>
</table>
| Aerospace | • Aircraft that fly faster and further on less energy.  
               • Air traffic control systems that make more efficient use of airspace. |
| Automotive | • Automobiles that are more capable and safer but use less energy.  
                    • Highways that are safe, higher throughput and energy efficient. |
| Defense | • Fleets of autonomous, robotic vehicles  
               • More capable defense systems  
               • Integrated, maneuverable, coordinated, energy efficient  
               • Resilient to cyber attacks |

Energy Internet: When IT Meets ET
Networking and computing delivers unique precision and flexibility in interaction and coordination.

**Computing/Communication**
- Rich time models
- Precise interactions across highly extended spatial/temporal dimension
- Flexible, dynamic communication mechanisms
- Precise time-variant, nonlinear behavior
- Introspection, learning, reasoning

**Integrated CPS**
- Elaborate coordination of physical processes
- Hugely increased system size with controllable, stable behavior
- Dynamic, adaptive architectures
- Adaptive, autonomic systems
- Self monitoring, self-healing system architectures and better safety/security guarantees.
...and the Challenges

Fusing networking and computing with physical processes brings new problems

### Computing/Communication
- Cyber vulnerability
- New type of interactions across highly extended spatial/temporal dimension
- Flexible, dynamic communication mechanisms
- Precise time-variant, nonlinear behavior
- Introspection, learning, reasoning

### Integrated CPS
- Physical behavior of systems can be manipulated
- Lack of composition theories for heterogeneous systems: much unsolved problems
- Vastly increased complexity and emergent behaviors
- Lack of theoretical foundations for CPS dynamics
- Verification, certification, predictability has fundamentally new challenges.
NSF Model of CPS

- A new underlying discipline
- Abstracting from sectors to more general principles
- Apply these to problems in new sectors
- Build a new CPS community

* Jeannette M. Wing
  Assistant Director, CISE, NSF
Assumption About CPS Core:

- Embedded SW
- Embedded HW
- Control
- Sensor Networks
- Formal Methods
- RT SW

New System Science: Physical & Computational
Outline

- Cyber Physical Systems (CPS)
- **Integration of CPS: Dealing with Heterogeneity**
- Example: Passivity-based Design
Key Idea: Manage design complexity by creating abstraction layers in the design flow.

Abstraction layers define platforms.

Abstractions are linked through mapping.

Abstraction layers allow the verification of different properties.

Claire Tomlin, UC Berkeley

Implications of mapping logical-time abstractions in SW to real-time applications:

- \( f \) : reactive program. Program execution creates a mapping between logical-time inputs and outputs.

- \( f_R \) : real-time system. Programs are packaged into interacting components. Scheduler control access to computation and communication resources according to time constraints \( P \)

\[
\forall \rho \in E, \forall \pi \in f_R(\rho), (\rho, \pi) \in P
\]

However, in CPS, essential system properties such as stability, safety, performance are expressed in terms of physical behavior!

- $f$: reactive program. Program execution creates a mapping between logical-time inputs and outputs.

- $f_R$: real-time system. Programs are packaged into interacting components. Scheduler control access to computation and communication resources according to time constraints $P$.

\[
f : [T \rightarrow In] \rightarrow 2^{[T \rightarrow Out]}
\]

\[
f_R : [T_R \rightarrow In] \rightarrow 2^{[T_R \rightarrow Out]}
\]

\[
\forall \rho \in E, \forall \pi \in f_R(\rho), (\rho, \pi) \in P
\]
Heterogeneous Abstraction Layers in CPS:

Goals:

- Compositional verification of essential dynamic properties
  - stability
  - safety
- Robustness against implementation changes and uncertainties
  - fault induced reconfiguration of SW/HW
  - network uncertainties (packet drops, delays)
- Decreased verification complexity

\[ p_R : [T_R \rightarrow In] \rightarrow 2^{[T_R \rightarrow Out]} \]

\[ f_R : [T_R \rightarrow In] \rightarrow 2^{[T_R \rightarrow Out]} \]

\[ f : [T \rightarrow In] \rightarrow 2^{[T \rightarrow Out]} \]

\[ \forall \rho \in E, \Psi_{out}(f_R(\rho)) \subseteq f(\Psi_{in}(\rho)) \]

\[ \forall \rho \in E, \forall \pi \in f_R(\rho), (\rho, \pi) \in P \]
Composition Inside Abstraction Layers 1/2

**Plant Dynamics Models** ↔ **Controller Models**

**Physical design**

**Software Architecture Models** ↔ **Software Component Code**

**Software design**

**System Architecture Models** ↔ **Resource Management Models**

**System/Platform Design**

**Dynamics:**  $B(t) = \kappa_p(B_1(t),...,B_j(t))$
- **Properties:** stability, safety, performance
- **Abstractions:** continuous time, functions, signals, flows,…

**Assumption:** Effects of digital implementation can be neglected

**Software:**  $B(i) = \kappa_c(B_1(i),...,B_k(i))$
- **Properties:** deadlock, invariants, security,…
- **Abstractions:** logical-time, concurrency, atomicity, ideal communication,…

**Assumption:** Effects of platform properties can be neglected

**Systems:**  $B(t_j) = \kappa_p(B_1(t_i),...,B_k(t_i))$
- **Properties:** timing, power, security, fault tolerance
- **Abstractions:** discrete-time, delays, resources, scheduling,
Controller dynamics is developed without considering implementation uncertainties (e.g. word length, clock accuracy) optimizing performance.

Assumption: Effects of digital implementation can be neglected

Software architecture models are developed without explicitly considering systems platform characteristics, even though key behavioral properties depend on it.

Assumption: Effects of platform properties can be neglected

System-level architecture defines implementation platform configuration. Scheduling, network uncertainties, etc. are introduced time variant delays that may require re-verification of key properties on all levels.
• Consequence of the lack of composability across system layers
  – intractable interactions
  – unpredictable system level behavior
  – full-system verification does not scale
• Approach: simplification strategies
  – Orthogonalization: Use passivity for decoupling stability and implementation induced time variant delays
  – ...

The golden rule of problem solving:
If the going gets tough, the tough changes the problem.
Outline

• Cyber Physical Systems (CPS)
• Integration of CPS: Dealing with Heterogeneity
• Example: Passivity-based Design
Key idea: Passivity-based design of networked control systems provides robustness to time-varying delays

- Various mathematical definitions
  - A passive system only stores and dissipates energy but cannot generate energy of its own
- Passive systems interact in a stable manner
  - When connected in either a parallel or negative feedback manner the overall system remains passive
- Passive control theory applies to
  - Linear and nonlinear systems
  - Continuous and discrete-time systems
- Easier and safer to control
  - Independent joint PD controller for robotic manipulator
  - Asymptotic stability for set-point tracking
Background on Passivity

• Milestones:
  – Wave digital filters (Fettweis, 70’s)
  – Dissipative dynamical systems (Willems, 70’s)
  – Passive structures (Peceli, 80’s)
  – Teleoperation over the Internet (Niemmeyer, 04)
  – Power junctions (Kottenstette, Antsaklis, 08)

• Work at ISIS:
  – Design tool suite for high confidence systems (Eyisi, Hall, Hemingway, Porter, Karsai, Kottenstette, Koutsoukos, Sztipanovits)
Networked Control System

Robot 1
Local Controller

Robot 2
Local Controller

Human Operator
Haptic Paddle

Network Controller

Network
Wave variables were introduced by Fettweis in order to circumvent the problem of delay-free loops and guarantee that the implementation of wave digital filters is realizable.

Wave variables defined by a bilinear transformation under which a stable minimum phase continuous-time system is mapped to a stable minimum phase discrete-time system. The transformation preserves passivity.
Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)

\[ u_{pk}(i) = \frac{1}{\sqrt{2b}}(b\theta_{pk}(i) + \tau_{dck}(i)) \]
\[ v_{c1}(j) = \frac{1}{\sqrt{2b}}(b\theta_{dp1}(j) - \tau_{c1}(j)) \]

Bilinear transform: power and wave vars.

- Bilinear transform (b)
- Power and Wave variables
- Passive down- and up-sampler (PUS, PDS)

- Delays
- Power junction
- Passive dynamical system
Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)

- Bilinear transform (b)
- **Power and Wave variables**
- Passive down- and up-sampler (PUS, PDS)
- Delays
- Power junction
- Passive dynamical system
Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)

- Bilinear transform (b)
- Power and Wave variables
- Passive down- and up-sampler (PUS, PDS)

- Delays
- Power junction
- Passive dynamical system
Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)

\[ u_{pk}(i) = \frac{1}{\sqrt{2b}}(b\theta_{pk}(i) + \tau_{dck}(i)) \]
\[ v_{c1}(j) = \frac{1}{\sqrt{2b}}(b\theta_{dp1}(j) - \tau_{c1}(j)) \]

Bilinear transform: power and wave vars.

- Bilinear transform (b)
- Power and Wave variables
- Passive down- and up-sampler (PUS, PDS)

- Delays
- Power junction
- Passive dynamical system
Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)

- Bilinear transform (b)
- Power and Wave variables
- Passive down- and up-sampler (PUS, PDS)
- Delays
- Power junction
- Passive dynamical system
Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)

- Bilinear transform (b)
- Power and Wave variables
- Passive down- and up-sampler (PUS, PDS)

- Delays
- Power junction
- Passive dynamical system
DSML for passivity-based design: PaNeCS

- Developed by Emeka Eyisi using the Model Integrated Computing (MIC) tools, (GME, UDM).
- PaNeCS Meta-model
  - **Main Components**
    - Plant Subsystem
    - Controller Subsystem
    - PowerJunction Subsystem
    - Network Subsystem
Plant Subsystem

• Components
  – Plant
    • Discrete-Time LTI
      \[ x(k + 1) = Ax(k) + Bu(k) \]
      \[ y(k) = Cx(k) + Du(k) \]
  – BilinearTransformP
  – PassiveDownSampler
  – PassiveUpSampler

• Interconnections
  – Plant_Bilinear
  – Bilinear_To_DownSampler
  – UpSampler_To_Bilinear
- **Components**
  - DigitalController
  - BilinearTransformC
  - Reference Input
  - ZeroOrderHold

- **Interconnections**
  - Controller_Bilinear
  - ZOH_Controller
  - Input_ZOH
PowerJunction Subsystem

• Components
  – PowerInputPowerOutput (Plant connection to PowerJunction)
  – PowerOutputPowerInput (Controller connection to PowerJunction)
Network Subsystem

- Network representation
  - Defines parameters for the network
  - Ability to introduce network disturbance for simulation purposes
Control Design Aspect

- Provides visualization of the control modeling layer indicating flow of control and sensor signals.
- Components represent control design concepts.
- Visible Components
  - Plant Subsystem
  - Controller Subsystem
  - Powerjunction
Platform Aspect

- Provides visualization of the physical platform layer indicating the flow of data packets over the network.
- Components represent physical entities
- Visible components
  - Plant Subsystem
  - Controller subsystem
  - Wireless network
• OCL Implementation
  – Connection between BilinearTransformC and DigitalController

Description: There must be one connection between the DigitalController block and the BilinearTransformC block

Equation:

```
self.connectionParts("Controller_Bilinear").size() = 1
```
Passivity Analysis

• In order to achieve the desirable properties of passive systems
  – Analyze the networked control system
• Analysis of the NCS
  – Component Analysis
  – System-level Analysis
Component Analysis

- Analyze individual components of the NCS
  - Only Plant and Controller Components
- Designed Model Interpreter Tool integrated in GME visits each tool and invokes the analysis function.

\[
\begin{bmatrix}
A^TPA - P - \bar{Q} & A^TPB - \bar{S} \\
(A^TPB - S)^T & -\bar{R} + B^TPB
\end{bmatrix} \leq 0
\]

\[
\bar{Q} - C^TQC, \quad \bar{S} - C^TS + C^TQD
\]

\[
\bar{R} - D^TQD + (D^TS + S^TD) + R
\]

\[
\exists \varepsilon > 0, \quad Q = -\varepsilon I, \quad R = 0, \quad S = \frac{1}{2} I
\]

- CVX semi-definite programming tool (SDP) used in a Matlab script to solve LMI.

Kottenstette, Antsaklis 2008
System-level Analysis

• Due to the “correct-by-construction” approach
  – Network as a whole ensure global robustness by a combination of
    • Individual components satisfaction of passivity constraints.
    • Passive Composition constraints encoded in the modeling language.
• Reduction in the analysis burden of verifying passivity.
PaNeCS Design Flow

NCS Modeling

PaNeCS GME
- structural constraints
- component passivity analysis

SL/tt model generator

PaNeCS 2tt
- model transformation

Simulink TrueTime
- behavior simulation

Experimental Setup

- Two CrustCrawler robotic arms
  - 4 DOF with AX-12 smart servos at each joint
- Novint haptic paddle
- Five networked Windows PCs with Matlab/Simulink
Experiment 1: Nominal Case

$x$-$y$-$z$ coordinates and angle of joint 2 of reference, robot 2, and robot 3
Experiment 1: Time Delay

Time Delay Between Robot 2 and Power Junction
Angle of joint 3 and y coordinate of reference, robot 2, and robot 3
Experiment 2: Intermittent Wireless Connection

Angle of joint 3 and y coordinate of reference, robot 2, and robot 3
Summary

- CPS “mindset” leads to new results:
  - Composition in heterogeneous CPS requires decoupling among design concerns (usually captured as different abstraction layers)
  - Decoupling requires significant effort, but the benefits are also significant: this is the primary tool for decreasing complexity
  - There is a performance tradeoff; in safety critical systems it is still the right choice!
- Active research area with significant industry pull
- Follow: www.cps-vo.org

