Composition in Heterogeneous Systems

Janos Sztipanovits
ISIS, Vanderbilt University

12 October, 2009
Outline

- Effects of heterogeneity
- Passivity-based design
- Toward a high-confidence model-based design tool chain
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
Abstraction layers: SW-RTS

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 \( P \).

\[
\forall \rho \in E, \forall \pi \in f_R(\rho), (\rho, \pi) \in P
\]
Abstraction layers: PHY-SW-RTS

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

\[
\forall \rho \in E, \forall \pi \in f_R(\rho), (\rho, \pi) \in P
\]
**Dynamics:** $B(t) = \kappa_p (B_1(t), \ldots, B_j(t))$
- **Properties:** stability, safety, performance
- **Abstractions:** continuous time, functions, signals, flows,…

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

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

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

**Systems:** $B(t_j) = \kappa_p (B_1(t_i), \ldots, B_k(t_i))$
- **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 introduce time variant delays that may require re-verification of key properties on all levels.
Composition and Heterogeneity

• 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
  – …
Outline

• Effects of heterogeneity
• Passivity-based design
• Toward a high-confidence model-based design tool chain
Physical layer: 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)
  – Passive structures (Peceli, 80’s)
  – Teleoperation over the Internet (Niemmeyer, 04)
  – Power junctions (Kottenstette, Antsaklis, 08)

• Work at ISIS:
  – Design tool suite and extension through applications (Eyisi, Hall, Porter, Kottenstette, Koutsoukos, Sztipanovits)
Networked Control System

Diagram:
- Robot 1
  - Local Controller
- Robot 2
  - Local Controller
- Network
- Human Operator
  - Haptic Paddle
  - Network Controller
• 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)

\[
u_{pk}(i) = \frac{1}{\sqrt{2b}}(b\theta_{pk}(i) + \tau_{dc2}(i))
\]

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

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

Sztipanovits: 14
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

Passive down-sampler and Passive up-sampler
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.

\[ \text{Passive down- and up-sampler} \]

\[ \text{PUS, PDS} \]

- 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

Bilinear transform:

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

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

Passive down-sampler and Passive up-sampler
Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)

\[ u_{pk}(i) = \frac{1}{\sqrt{2b}} (b\theta_{pk}(i) + \tau_{dc}(i)) \]

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

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: power and wave vars.

- Bilinear transform (b)
- Power and Wave variables
- Passive down- and up-sampler
- Delayed and Passive up-sampler

(PUS, PDS)

Passive down-sampler and Passive up-sampler

Sztipanovits: 19
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
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
Controller Subsystem

- **Components**
  - DigitalController
  - BilinearTransformC
  - Reference Input
  - ZeroOrderHold

- **Interconnections**
  - Controller_Bilinear
  - ZOH_Controller
  - Input_ZOH
• **Components**
  
  – **PowerInputPowerOutput** (Plant connection to PowerJunction)
  
  – **PowerOutputPowerInput** (Controller connection to PowerJunction)

\[
\sum_{k=m+1}^{n} (u_k^T u_k - v_k^T v_k) \geq \sum_{j=1}^{m} (u_j^T u_j - v_j^T v_j)
\]
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
Sample OCL Implementation

- OCL Implementation
  - Connection between BilinearTransformC and DigitalController

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

Equation:

```java
self.connectionParts("Controller_Bilinear").size() = 1
```
• 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.

$$\begin{bmatrix} A^T P A - P - Q & A^T P B - S \\ (A^T P B - S)^T & -R + B^T P B \end{bmatrix} \preceq 0$$

$$Q = c^T Q c, \quad S = c^T S + c^T Q D$$

$$R = D^T Q D + (D^T S + S^T D) + R$$

$$\exists \alpha > 0, \quad Q = -\alpha I, \quad R = 0, \quad S = -\frac{1}{2} I$$

- CVX semi-definite programming tool (SDP) used in a Matlab script to solve LMI.
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
- structural constraints
- component passivity analysis

SL/\text{tt} model generator
- 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
Outline

• Effects of heterogeneity
• Passivity-based design
• Toward a high-confidence model-based design tool chain
Model-based Design Flow

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 introduce time variant delays that may require re-verification of key properties on all levels.
Controller dynamics is developed using passivity-based design. Current work focuses on performance optimization and safety.

Assumption: Time varying delays do not impact stability.

Software models need to be verified for essential properties e.g. deadlock freeness. -> BIP

Assumption: Effects of platform on properties can be analyzed

System-level architecture defines implementation platform configuration. Scheduling, network uncertainties, etc. are introduce time variant delays that may require re-verification of key properties on all levels.
Experimental Platform

- Gumstix/Robostix
- Linux + AVR micro
- TT Virtual Machine on Linux/UDP + FreeRTOS
- No fault tolerance (yet)
Passive systems are robust to network delays and quantization errors. We can design controllers to “passify” many systems that are not quite passive.
This example is a subset of the full quadrotor model we’re using for testing.
Design Flow: Moving toward implementation

Plant Dynamics Models  Controller Models

Controller design

Simulink models imported into GME

Sztipanovits: 42
Design Flow: Platform

Two processors connected by a time-triggered bus

System timing parameters are captured here:

- fundamental tick time of processing units
- data transfer setup times
- bus rates
Design Flow: Software

Software architecture (types and dependencies)

Software Architecture Models ↔ Software Component Code
Software design

Deployment details:
- tasks
- messages
- processor assignments
HCDDES Tool Chain

PaNeCS GME

PaNeCS 2tt

EsMoL2tt

Simulink TrueTime

mdl2mga

EsMoL GME

EsMoL2 Sched

BIP PN-Mod

BIP Anal.

BIP – based deadlock analysis

Passivity – based design

sl C/G

s/f C/G

robo C/G

gstix C/G

XPC Plant

robo Frodo

gstix Frodo

Sztipanovits: 45
• Composition in heterogeneous systems requires decoupling among design concerns
• 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 still may be the right choice.
Some References


Passive Up-sampling and Down-sampling

- Because of bandwidth constraints, the local digital controllers for each robot run at a faster rate than the network controller.
- Ensure that no energy is generated, and thus passivity is preserved.
- Passive down-sampling

\[ u_{pDsk_k}(j) = \sqrt{\sum_{i=M(j-1)}^{Mj-1} u_{pk_k}(i)^2 \sqrt{\sum_{i=M(j-1)}^{Mj-1} u_{pk_k}(i)}} \]

- Passive up-sampling

\[ u_{pk_k}(i) = \sqrt{\frac{1}{M}} u_{pDsk_k}(j-1), \quad i = Mj, \ldots, M(j+1) - 1 \]

where \( i = \left\lfloor \frac{t}{T_s} \right\rfloor \) and \( j = \left\lfloor \frac{t}{MT_s} \right\rfloor \)
• Compose a network in which multiple passive plants can be interconnected to multiple passive controllers

• Interconnect wave variables from multiple controllers and plants such that the total power input is always greater than or equal to the total power output

\[ \sum_{k=m+1}^{n} (u_k^T u_k - v_k^T v_k) \geq \sum_{j=1}^{m} (u_j^T u_j - v_j^T v_j) \]