Highlights from Year 2 – A Quick Glance

Smörgåsbord Tapasbord
Outline

• Highlights related to CoDesign Tools – Martin Törngren, KTH
• Embedded Control Design – Alfons Crespo – UPVLC
• Conclusions from the ”Workshop on Interaction Between Control and Embedded Electronics in Automotive Industry” – Karl-Erik Årzén
• New European Projects
• Technical Highlights:
  – Recent advances in event-based control – Karl-Erik Årzén
  – 6 Project Examples
Co-design tools

- Individual tool development
- Point integration of tools
- Extended tool survey, including tool characterization


Towards systematic model/tool integration and management

- Survey
- Cross-cluster cooperation
- Initiated ARTIST2 synchronization
  - Which tools are available? Needs and potential for cooperation and integration?
Tool developments, individual and integration

**Jitterbug**: [http://www.control.lth.se/~lincoln/jitterbug/](http://www.control.lth.se/~lincoln/jitterbug/)
- GUI in development

**Truetime**: [http://www.control.lth.se/truetime/](http://www.control.lth.se/truetime/)
- Version 1.4 released (1.5 on its way)
- > 2,000 downloads of 1.3


**AIDA/Saint**

Use-case demonstrations of point tool integration
- TrueTime – co-simulation --- results as inputs to Jitterbug – analytic measures of control performance
- Torsche – response time analysis --- results as inputs to Truetime co-simulation
- Simulink, UML tools, PDM database for system configuration and architectural design
Extended tool survey – Co-design of control systems and their real-time implementation

Industrial model based development, perspective

Tools from closely related domains
  – Aires, Metropolis, Giotto, HIP-Hops, Sildex, Syndex, TT-Tech tool suite, …
  – Analytical verification, Distributed systems, Safety & reliability

Detailed overview and characterization of selected tools
  – AIDA, Jitterbug, ORCCAD, Ptolemy II, RTSIM, Syndex, Targetlink, Torsche, TrueTime

Discussion: Trends and challenges
  – Complementary tool functionality
  – Opportunities for joint work, need for ARTIST2 synchronization
Characterizing co-design tools

Digital controller

Domain specific design parameters and Qualities

Plant

Abstraction, constraints & views

Mapping

Analysis & synthesis

Domain specific design parameters and qualities

platform

Hardware

Software

Digital controller

Computation

Communication

Timing

Accuracy

Memory,

Power, ...

Abstraction,

constraints & views

Platform

SW

HW
Control-platform co-design

Control

Hybrid Systems
- e.g. Dymola, Hyvisual

Multi-Model Design Environments
- e.g. Ptolemy II

Discrete-event systems
- e.g. Uppaal, Sildex

Hardware-software co-design
- e.g. Metropolis

Co-Design Tools

Safety & Reliability
- e.g. FTA-tools

Real-Time Scheduling & Computing
- e.g. Aires, MAST

Networking
- e.g. NS2, TT-tech tools
Model and tool integration survey

Coarse grouping of approaches:
- Modeling languages; e.g. SysML, UML2, AADL, EAST-ADL
- Model & tool integration; e.g. ToolNet, GeneralStore, Fujaba, MIC
- Information management; e.g. PDM tools, SystemWeaver, ModelCenter

Domain/discipline origins, different focus, tailorability

Engineering and information management dimensions

Several integration (process) and product decomposition (architecture) patterns

APIs, Formats, Model transformations, Information models/languages

Product analysis and synthesis

Adaptability, Configuration management, consistency, views, work-flow management
Outline

• Highlights related to CoDesign Tools – Martin Törngren, KTH
• **Embedded Control Design** – Alfons Crespo – UPVLC
• Conclusions from the "Workshop on Interaction Between Control and Embedded Electronics in Automotive Industry" – Karl-Erik Årzén
• New European Projects
• Technical Highlights:
  – Recent advances in event-based control – Karl-Erik Årzén
  – 6 Project Examples
Common framework definition

- A **common framework** of the **control parameters** that can be influenced by an embedded control system implementation

- **Real time operating systems criteria** that can be adjusted to increase the **robustness** of the control system
Parameters analysis:

- **Structural.** Linked to the control structure.
  - single/multiple loop, adaptive, coordinated, fault tolerant

- **Requirements / Performance**
  - Temporal aspects, frequency, cost indexes, control effort

- **Operational**
  - Timing Parameters, Latencies, Data availability, Event based control.

- **Computational**
  - Fixed-time computation, Variable-time computation, Optional parts: reasoning, optimization
Control Design

- Performances: Control effort
- Sampling pattern:
  - Sampling period, Multirate operation, Nominal delays, Delays interval, Sensed data availability, Control data availability
- Control strategy:
  - Alternative controllers, Refining (optional) tasks, Controllers transfer policy, Internal information updating
- Timing Parameters:
  - Priority, Relative deadline, Execution time distribution, Response time, Input / Output jitter, Control server period and budget, …
- Real-time operating system support
  - Scheduling policies, control kernel support,
Control Implementation

- To provide a specific middleware for control applications and a scheduling scheme to execute complex control applications.
Control kernel

Functionalities

- **Ensuring control action (CA) delivering** at time even if the calculated action for this period has not been updated. In this case, a **backup control action** has to be delivered.
- **Data acquisition** of major signals. A **past data** is considered **better** than a delayed data.
- **Transfer to new control structure**: Control system provides several regulators that can be used in different situations.
- **Communication facilities**
Control scheme

- Optional activities
- Controller
- Control Kernel

Controller

Optional

define send actions

API

DAT
- Period
- Value, Time
- Threshold
- Max_latency

OAT
- Period
- Offset
- Safe Value
- Backup
- Current

events
RTOS support: Implementation

- A new RTOS **eRTLinux (Partikle core)** has been developed by UPVLC
  - POSIX Compliant
  - Control Kernel support
  - Control Server support
  - C, C++, Ada, RTJava
- It can be **deployed** for three targets:
  - Linux process
  - Stand-alone (bare machine)
  - XtratuM domain (system partition with temporal and spatial isolation)

The control middleware is under development.
Control Kernel features

- **Control kernel extracts** the basic control services and defines a middleware for control purposes.

- The scheduling scheme that support this architecture permits the execution of the control kernel and control loops.

- Different scheduling policies can be applied to different control levels.

- The control kernel obtains an improvement with respect to the CPU use when the system is in stable conditions.

- The control kernel can be implemented in hardware or in a RTOS.

- The CPU saving allows the execution of optional parts to improve the control answer or to recalculate predictive models.
Outline

• Highlights related to CoDesign Tools – Martin Törngren, KTH
• Embedded Control Design – Alfons Crespo – UPVLC
• Conclusions from the ”Workshop on Interaction Between Control and Embedded Electronics in Automotive Industry” – Karl-Erik Årzén
• New European Projects
• Technical Highlights:
  – Recent advances in event-based control – Karl-Erik Årzén
  – 6 Project Examples
Conclusions from the Workshop on Interaction Between Control and Embedded Electronics in Automotive Industry

• Co-located and integrated with the "Beyond Autosar" workshop
• Organized jointly with the RT Components cluster (Albert Benveniste and Werner Damm)
• Three keynote presentations:
  – Stefan Kowalevski, RWTH Aachen (Bosch)
  – Karl-Erik Årzén
  – Carlos Canudas de Wit, LAG Grenoble
• Panel discussion
Conclusions

- There is a still a misunderstanding between control & software engineers in the automotive industry
Different Worlds

Software Community
- UML2
- OMG
- SysML
- Marte
- EAST ADL
- MDA

Control Community
- Simulink components
- Modelica
- Real-Time Workshop
- Stateflow
- Model-based design

?
Conclusions

- The move from federated to integrated architectures will have large consequences for automotive control
- The needs of control should have strong influence on the integrated architectures
  - Control is absolutely essential in modern cars
    - Powertrain, emissions, vehicle dynamics, safety systems, …
  - Control gives performance, safety, and low emissions
Consequences for Control

• A sensor will be used by several systems
  – part of the vehicle platform, or
  – part of one system but made available to other systems, possibly using middleware techniques

⇒ **Sensor components will be special**

• The same actuator will be used by several systems
  – Brakes will be used by intelligent cruise control, lane following system, collision avoidance system, ESP, anti-rollover, sidewind compensation, ..

⇒ **Actuator components will be special**

```
System 1
System 2
System 3
Selector Logic
Actuator Controller
Actuator
```

Consequences for Control

- Cascaded control structures will dominate
  - hierarchical, layered
- The different controller components will be part of different systems residing on the same or on separate ECUs
Conclusions

- There are no clear-cut answers on the ET/TT debate from a control perspective
  - It is an advantage if sampling is time-triggered
    - Ex. multi-input controllers with sensors at different nodes
  - It is an advantage if actuation is event-triggered
    - A short but varying input-output latency is almost always better from a control performance point of view than a long but constant latency, also if the latter is compensated for
    - Specially important in multi-layer cascade structures
Conclusions

• The move from a federated to an integrated architecture opens up new possibilities for automotive control
  – The constraints of the federated system architectures have had a strong impact on how the control systems currently are structured
  – With an integrated architecture new ways of structuring the control systems are possible
• Take a completely new look upon how the overall control system for a car ought to be structured, including powertrain control, chassis control, safety systems, etc.
Outline

• Highlights related to CoDesign Tools – Martin Törngren, KTH
• Embedded Control Design – Alfons Crespo – UPVLC
• Conclusions from the ”Workshop on Interaction Between Control and Embedded Electronics in Automotive Industry” – Karl-Erik Årzén

• New European Projects
• Technical Highlights:
  – Recent advances in event-based control – Karl-Erik Årzén
  – 6 Project Examples
New European Projects

- **Service-Oriented CRoss-layer infrAstructure for Distributed smart Embedded deviceS (SOCRADES)** – EU/IST IP
  - Schneider Elec. (coord), **ABB**, Siemens, **KTH**, ….

- **Framework for Real-time Embedded Systems based on ContRacts (FRESCOR)** – EU/IST STREP
  - Universidad de Cantabria, University of York, Scuola Superiore Sant´Anna, TU Kaiserslautern, **UPVLC**, **CTU**, ENEA, ….

- **Smart Embedded Network of Sensing Entities (SENSE)** – EU/IST STREP
  - ARC Seibersdorf, **UPVLC**, PARAGON, University of Galati, University of Patras, TUW, University of Krakow,….

- **Advancing Traffic Efficiency & Safety through SW Technology (ATESST)** – EU/IST STREP
  - **Volvo**, VCC, Mecel, Daimler, Etas, SiemensVDO, Carmeq, Volcano/MentorGraphics, Mecel/Delphi, CEA, **KTH**, TUB

- **Dynamically Self-Configuring Automotive Systems (DYSCAS)** – EU/IST STREP
  - **Volvo**, Bosch, Daimler, Movimento, Systemite, **KTH**, C-Lab (Paderborn), Univ. of Greenwich

- **European Leadership in System Modeling and Simulation through advanced Modelica Libraries (EUROSYSLIB)** – ITEA 2 proposal
  - Dassault Systèmes (coord), INRIA, EDF, DLR, Fraunhofer, Siemens, **Lund**, ….
New Project: FRESCOR

- Framework for Real-time Embedded Systems based on ContRacts
  - EU/IST STREP started 01.06.2006 (three years)
  - 10 partners including Universidad de Cantabria, University of York, Scuola Superiore Sant’Anna, TU Kaiserslautern, **UPVLC, CTU, ENEA, ....**
  - 4,26 M€

- Define, implement and provide support to a contract model that specifies which are the application requirements with respect to the flexible use of the processing resources in the system.

- Control parameters and execution support for control applications are included in the contract requirements
  - Strong connection to the framework activity within this cluster
New Project: SENSE

• Smart Embedded Network of Sensing Entities
  – EU/IST STREP started 01.09.2006 (three years)
  – 9 partners including ARC Seibersdorf, UPVLC, PARAGON, University of Galati, University of Patras, TUW, University of Krakow….
  – 1,7 M€

• To create a system in which distributed embedded devices cooperate to form and maintain a self-consistent global world view from local sensor information and which is robust to the addition and removal of devices from the network.

• Development of the embedded system platform incorporating both acoustic and visual sensors and data processing.

• Prototype: International Airport Krakow and Balice prototype. Control of objects (luggage, ….), personal movement, …
New Project: ATESSST

- Advancing Traffic Efficiency & Safety through SW Technology
  - EU/IST STREP (2006-2008) – 4 M€
  - **Volvo, VCC**, Mecel, Daimler, Etas, SiemensVDO, Carmeq, Volcano/MentorGraphics, Mecel/Delphi, CEA, **KTH, TUB**

- Background:
  - Architecture Description Language for Automotive Embedded Systems, EAST ADL 2.0

- Alignment: SysML, AADL, AUTOSAR and MARTE
- Dissemination by UML2 Profile definition
- Methodology, prototype tool and demonstrator
New Project: DYSCAS

- Dynamically Self-Configuring Automotive Systems
  - EU/IST STREP (2006-2008), 3,5 M€
  - Volvo, Bosch, Daimler, Movimento, Systemite, KTH, C-Lab (Paderborn), Univ. of Greenwich

- A middleware for automotive embedded systems with support for configurational flexibility, scalability and automotive constraints.

- **Use cases**: Optimal resource utilization, External hardware integration (plug-and-play), Fault-tolerance, Remote diagnostics and maintenance

Vision
New Project: SOCRADES

- Service-Oriented CRoss-layer infrAstructure for Distributed smart Embedded deviceS
  - EU/IST FP6 IP
  - 3-years project started 01.09.2006
  - 15 Partners incl. Schneider Elec. (coord), ABB, Siemens, KTH, .....
  - Total Budget: 13.7 M€

- Focus areas:
  - Industrial Automation
  - Networked Embedded Systems (wired and wireless)
  - Web Services technology
New Project Proposal: EUROSYSLIB

- European Leadership in System Modeling and Simulation through advanced *Modelica* Libraries
  - ITEA2 proposal under evaluation
  - 21 partners incl. Dassault Systèmes (coord), INRIA, EDF, DLR, Fraunhofer, Siemens, **Lund Univ.**
  - 23,1 M€ (9.6 M€ Funded) over three years

- **Modelica**
  - Multi-domain modeling and simulation
  - Equation-oriented acausal object-oriented models

- **Project:**
  - Library development within the Mechanical, Electrical and Electronics, Thermofluid, Control, Safety and Automotive domains
  - Establish *Modelica* as *The European System Modeling Language*
    - Compare with SysML
  - Lund University part
    - Real-time network simulation library based on the network simulation facilities of TrueTime
Outline

• Highlights related to CoDesign Tools – Martin Törngren, KTH
• Embedded Control Design – Alfons Crespo – UPVLC
• Conclusions from the ”Workshop on Interaction Between Control and Embedded Electronics in Automotive Industry” – Karl-Erik Årzén
• New European Projects
• **Technical Highlights:**
  – Recent advances in event-based control – Karl-Erik Årzén
  – 6 Project Examples
Recent Advances in Event-Based Control

• Control almost always assumes and requires *periodic* sampling
  – Well-developed theory

• Cornerstone for large part of real-time systems

• However, what if we relax this assumption?
  – Sample and control only when an *event* has occurred, e.g., a threshold crossing
  – Most likely closer to how nature performs feedback
  – Several practical observations have reported that event-based control can perform as good or better than time-based control
  – But, very very little theory
Previous Work

[Åström and Bernhardsson, 1999]

- Aperiodic event-based control of first-order stochastic systems
- Reduction of variance by a factor 3 for an integrator systems, assuming the same average sampling interval

- However, sampling infinitely often is not realistic
Sporadic Event-Based Control

- Recent work by Johannesson, Henningsson and Cervin, Lund
- First-order stochastic system
  - $dx = a \ x \ dt + u \ dt + dw$
  - $w$ – Wiener process with unit incremental variance
- Apply impulse control when $|x| > r$
  - State is reset to zero
  - Dirac-pulse control signal
- **Sporadic Event-Based Control**
  - Minimum inter-event time $\rightarrow$ finite utilization factor
- Two versions:
  - Discrete-time measurements (DT)
  - Continuous-time measurements (CT)
Example: Sporadic Control (CT)

- $T = 1$
Cost Function

\[ J = \lim_{t \to \infty} \frac{1}{t} \int_0^t x^2(\tau) \, d\tau + \rho \lim_{t \to \infty} \frac{1}{t} N_u(0, t) \]

\[ J_x \quad \text{State cost} \]
- Stationary process variance

\[ J_u \quad \text{Control cost} \]
- Average number of events / time unit
Solution Principle

1. Find stationary distribution of $x$ as function of $r$
   - No analytical solutions found
   - Approximated by piecewise polynomials (CT case)
   - Approximated by discretization (DT case)
2. Calculate $J_x(r)$ and $J_u(r)$
3. Choose $r$ to minimize $J = J_x + \rho J_u$
Optimal Choice of Threshold under Sporadic Control (CT)

- Local minimum for some $r > 0$
- For small errors, it is better to wait than to control!
Optimal Cost for the Integrator

![Graph showing the optimal cost for the integrator with different controllers.](image)
Conclusions

• Sporadic control can reduce the process variance and/or the control frequency compared to periodic control
• More realistic than aperiodic event-based control
• Many interesting further problems for higher-order systems
  – When to generate events?
  – What control actions to apply?
  – Event-based observers (MSc thesis by Toivo Henningsson)
• Applications:
  – Applications where sampling and computation are costly
    • Networked embedded control systems with limited bandwidth
    • Inventory control
  – Applications that are event-based, but are approximated by continuous-time models
    • "Liquid" models of queueing systems in server systems
Outline

• Highlights related to CoDesign Tools – Martin Törngren, KTH
• Embedded Control Design – Alfons Crespo – UPVLC
• Conclusions from the “Workshop on Interaction Between Control and Embedded Electronics in Automotive Industry” – Karl-Erik Årzén
• New European Projects
• Technical Highlights:
  – Recent advances in event-based control – Karl-Erik Årzén
  – 6 Project Examples
1: Optimal controller period selection under FP scheduling

**Partners:** SSSA (Bini) & Lund (Cervin)

**Context:**
- A set of periodic control tasks executing on a single CPU under fixed priority scheduling

**Background:**
- Traditionally, \( D \leq T \) for control tasks
- Leads to processor under-utilization

**Approach:**
- Explore what control performance is possible to gain by moving outside the FP schedulability bound
- Optimal period assignment using combination of
  - Simple analytical upper bounds on response times \( R \)
  - Control cost functions \( V(T, R) \)

**Conclusion:** Utilization = 1 is optimal under FP scheduling

**Status:** Ongoing work

\[
\text{Minimize } \sum V_i(T_i, R_i) \\
\text{subject to } \sum \frac{C_i}{T_i} \leq 1
\]
2: Control task jitter reduction

**Partners:** Lund (Cervin) & SSSA (Buttazzo)

**Background:**
- Jitter in input/output latency causes poor control performance

**Approach:**
- Comparative assessment and evaluation of three different jitter control schemes – mixed task set (control and non-control)
  - Task splitting (high-priority input/output subtasks)
  - Deadline shortening (raise the priority of the control tasks)
  - Non-preemptive scheduling

**Conclusion:**
- Deadline shortening typically gives the best control performance and is also the easiest to implement
  - Exception: Task splitting works better for system with very poor stability margins

**Status:** Under submission
3: Control task model evaluation

Partners: UPVLC (Lluesma et al) & Lund (Cervin et al)

Background:
- Splitting a control task into subtasks may reduce input/output latency and jitter
- Several alternative approaches

Approach:
- Comparative evaluation of four task models:
  - Standard task model (single task)
  - CalculateOutput + UpdateState (CO_US)
  - Initial, Mandatory and Final (IMF)
    - Sampling (M), Computations (L), Actuation (H)
  - Initial, CO, Final, US (ICOFU)
- Evaluated on a large set of plant models
- Control performance evaluated using the Jitterbug and TrueTime tools from Lund

Conclusions:
- CO_US and ICOFU give best performance

Status: Presented at IEEE CACSD’06 and at the RTCSA’06
4: Feedback Scheduling of Control Tasks

Partners:
- UPC (Castañé, Martí) & Lund (Henriksson, Cervin)

Background:
- Feedback scheduling of CPU time can improve control performance and increase resource utilization

Previous work:
- On-line scaling of task periods such that the total performance is optimized (Eker, Hagander, Årzén – 2000)
  - Actual performance of the individual loops not taken into account
- Feedback scheduling using feedback from plant states (Henriksson, Cervin - 2005)

Current work:
- Extensions to
  - Arbitrary linear controllers incl.e.g., state observers & PID
  - More efficient on-line optimization
  - Experimental verifications

Status: Presented at ECRTS’06
5: Distributed Resource Allocation in Server Clusters for QoS

Partner: KTH (Johansson et al)

Context:
- Large scale web server clusters
- Multiple services are offered, each with its own QoS objectives
- Each server provides one service at a time
- Service selection dynamically allocates cluster resources to services

Approach:
- Resource allocation problem formulated as a discrete utility maximization problem
- Efficient centralized algorithm developed
- Three suboptimal decentralized (local) schemes proposed and evaluated

Status: Will be presented at IEEE CDC’06
6: Code optimization for FPGAs

- **Partner:** CTU (Hanzalek, Sucha)
- High-level *synthesis of control algorithms*
  - algorithm description in Matlab like language
  - off-line scheduling
  - automatic code generation (Handel C or VHDL)
- **Specific HW architecture – FPGAs**
  - high degree of parallelism
  - dedicated units (e.g. floating point)
  - pipelining, shared memory, reconfiguration
- **Optimality** - objective is to find a feasible schedule with minimal $C_{\text{max}}$
  - iterative algorithms (IEEE RTAS)
  - iterative algorithms with imperfectly nested loops (Journal of VLSI Signal processing systems)