Timing problems and opportunities for embedded control systems
- modeling and co-design

Prof. Martin Törngren, KTH,
Embedded control systems/
Mechatronics/Machine Design
Royal Institute of Technology - KTH

Review and collaboration

- Work since the 90s

  Example results

  - Scheduling theory better tailored to control systems
  - Relaxed timing requirements for control systems
  - Tools

- Cooperative work
  - Academia: Lund, Chalmers, MDH & **Artist** network
  - Industry: Arcticus, dSPACE, Scania, Volvo AB

  People: Karl-Erik Årzen, Björn Wittenmark, Johan Nilsson, Magnus Gäfvert, Christer Norström, Kristian Sandström, Jan Torin, Henrik Lönn, Mats Andersson, Jad Elkhoury, Martin Sanfridson, Ola Redel, DeJiu Chen and Jan Wikander

Questions addressed in the talk:
- sampled control theory – feedback control

- Which kind of timing problems can be analyzed and handled?
- How can timing requirements be formulated and derived?
- What are the approaches for codesign of controllers and their embedded systems implementation?
- How can timing properties be expressed for analysis and/or synthesis?

Outline

- **Background and motivation**
  - Industrial control systems perspectives
  - Gaps and misconceptions
- Sampled control theory
  - Assumptions, properties and timing problems
- Codesign
  - Approaches
  - Abstractions
- Conclusions

Background

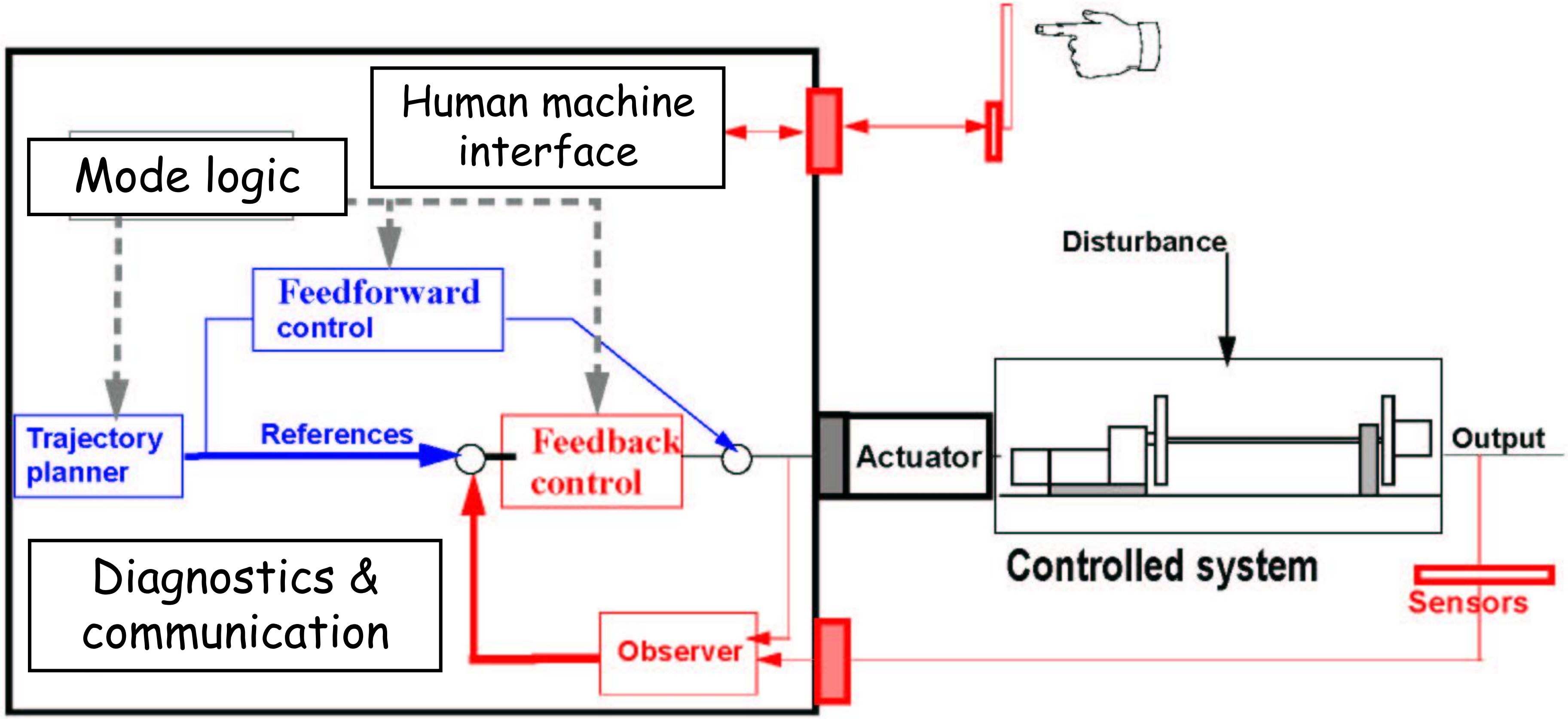
- Embedded control systems are evolving
  - More applications, more functions and modes
  - More networking and resource sharing
  - ➔ More timing issues in distributed systems
  - ➔ More safety concerns
  - ➔ Systems integration problems: feature interaction
- Despite work since the 70s, is still a gap from control theory to embedded systems
  - Pieces exist
  - Multidisciplinary challenge

Innovative Centre for Embedded Systems

Motion control demo

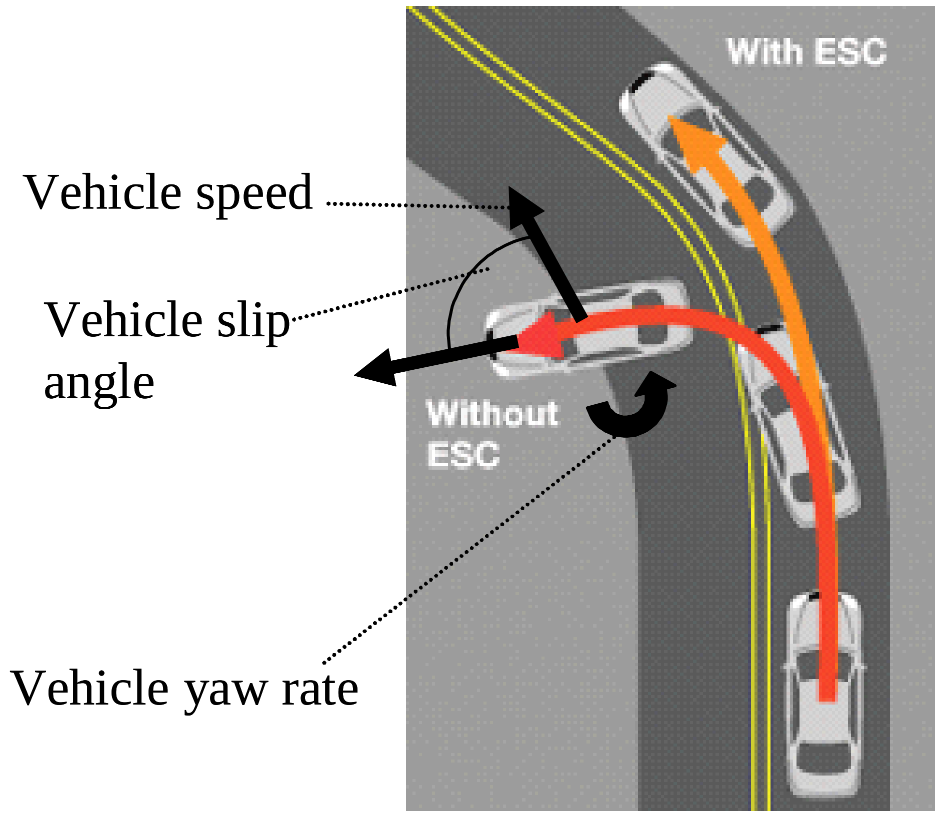
- Is this a hard or soft real-time system
- Why type of control is exercised?
  - Feedforward?
    - And mode changes?
  - Feedback?
  - Hierarchical control?
    - Inner and outer loops!
- What are the timing constraints?

Embedded control systems



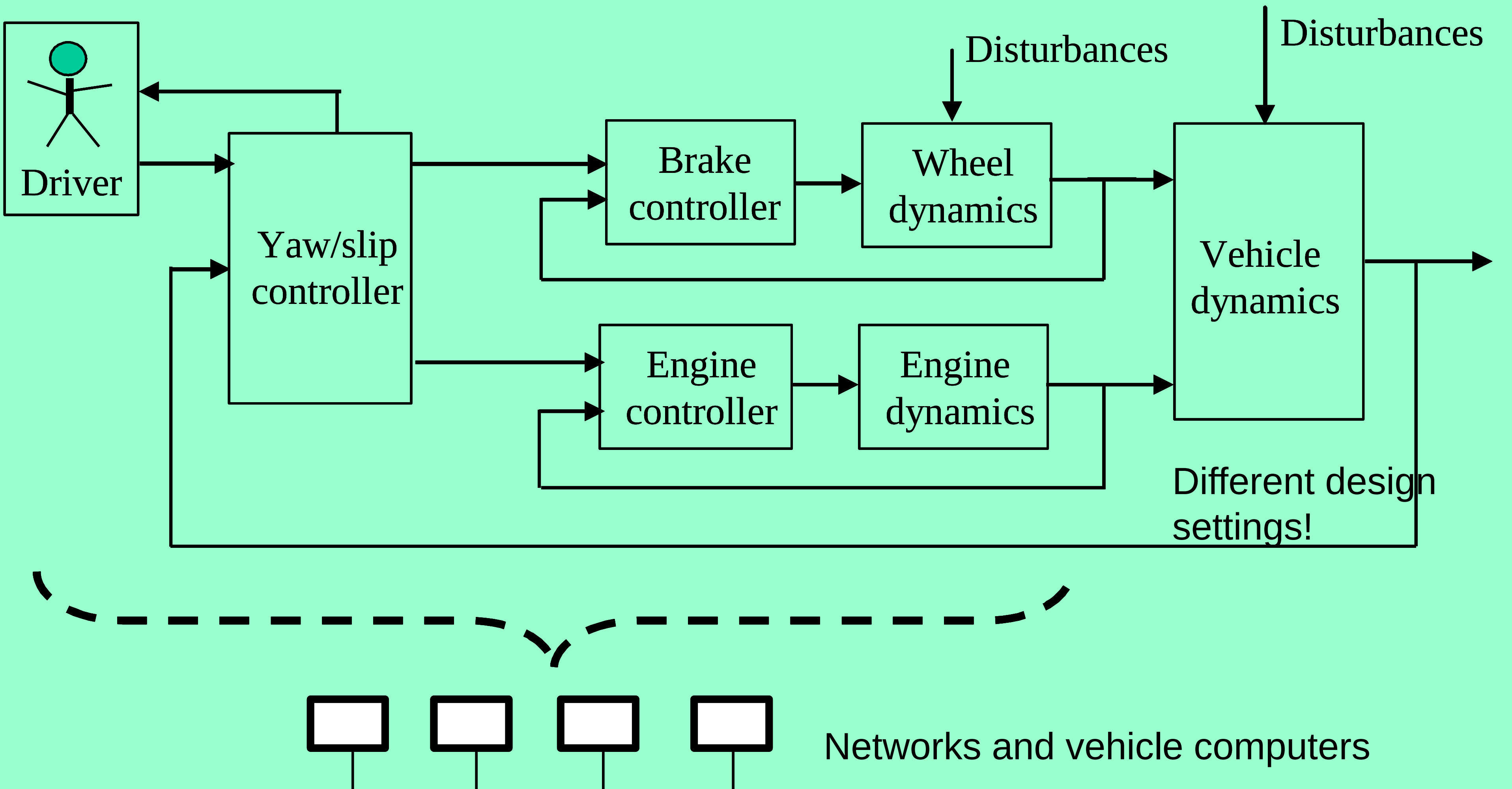
Note: Controller code is usually a small part of the total amount of code

Example application: stability control

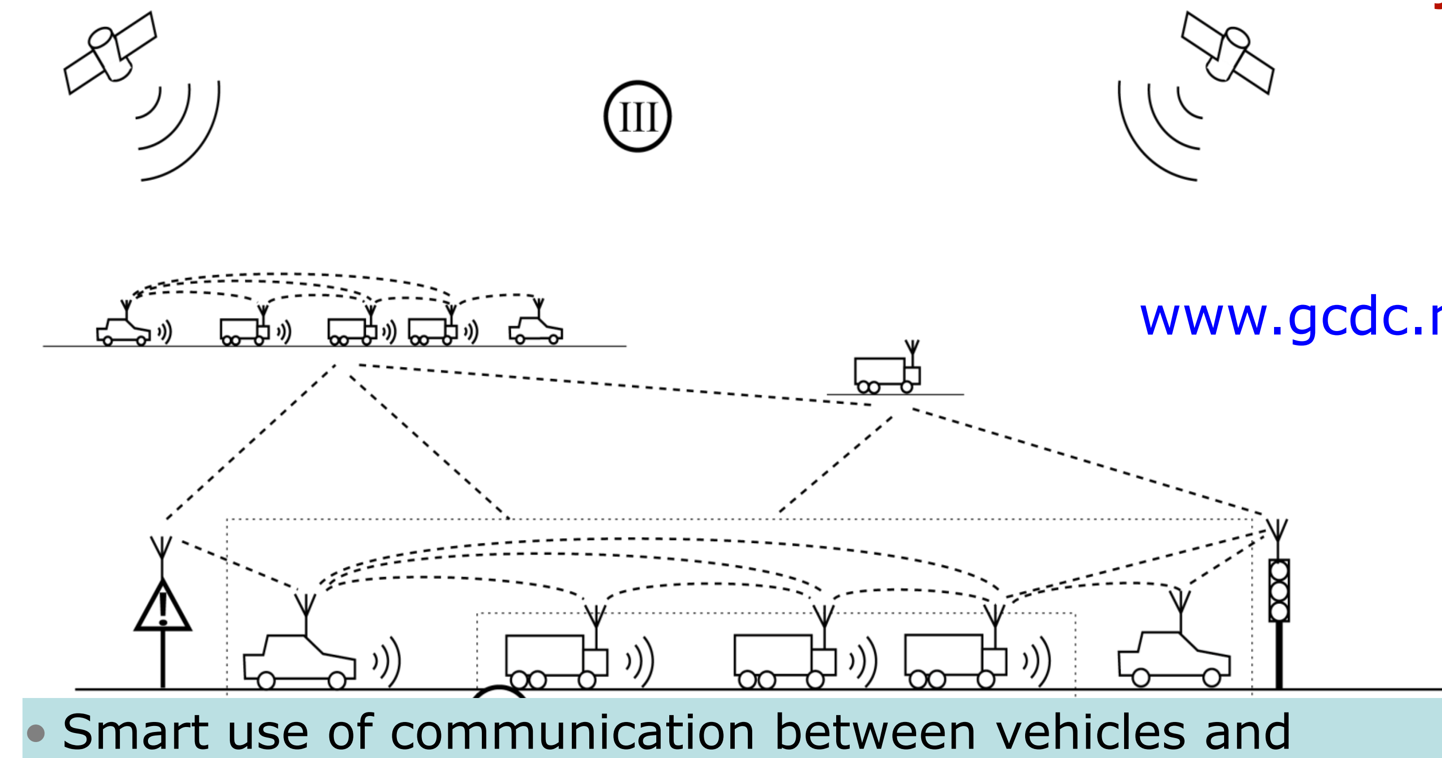


Source: ESC education - www.esceducation.org

Example application: vehicle stability control



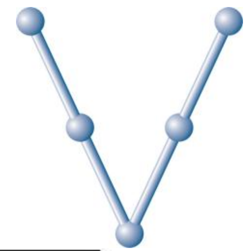
Cooperative driving competition to stimulate work towards traffic efficiency



www.gcdc.net

- Smart use of communication between vehicles and infrastructure
- Cooperative Driving promises to results in:
  - Efficient and smooth driving (50% less traffic congestion[1])
  - Safety (8% less traffic accidents[1])
  - Clean driving (5% less CO_2 emission and fuel consumption[1])

Industrial engineering practice



	Model-in-the loop	Software-in-the-Loop	Processor-in-the-Loop	Hardware-in-the-Loop
logic, algorithms	√			
accuracy effects		√	√	√
programming errors		√	√	√
compiler errors			√	√
real-time scheduling problems				√
speed/memory problems				√
I/O software				√

Inspired from Hanselman, 1998

Incorporation of system level timing analysis and synthesis – an exception

- Vaguely specified timing constraints
- Exceptions (examples)
  - Pragmatic use of RMA – a la RM handbook
    - Example: ESA since long uses RMA in analysis
  - Cyclic executives and synchronous languages
  - Volvo example
    - Volcano – Volvo CAN (Ken Tindell)
    - Rubus component mode and RTOS
- Increasing demand and new work
  - TADL (Timmo, ATESST2), SymptaVision
  - ADLs, Marte

Gaps

Computer performance
(utilization, throughput, …)

Computer design parameters
(structuring, scheduling, messaging, priorities, deadlines, …)

Possible intermediate abstractions?

Complex relationships

Control performance
(variance, rise time, … cost functions, ….)

Control design parameters
(control principle, structure, dynamics shaping, …

- Impact of design decisions
- Need for appropriate interactions & abstractions
  - Example: Scheduling execution models: {*C*, *D*, *T*}
  - Simple but does not capture true requirements
  - Overconstraining.

Misconceptions

- Constant delay is better than a shorter but varying delay

Average delay is most important
Response time delay can not be avoided

- There is one suitable sampling period given by control design

A range of feasible sampling periods
Many aspects influence choice

- Control systems involve hard real-time constraints
  - A deadline for one cycle is a HRT constraint

Hard vs. Soft – two interpretations!
Have to define what deadline refers to!
Most control systems are soft in that missing one cycle can be tolerated

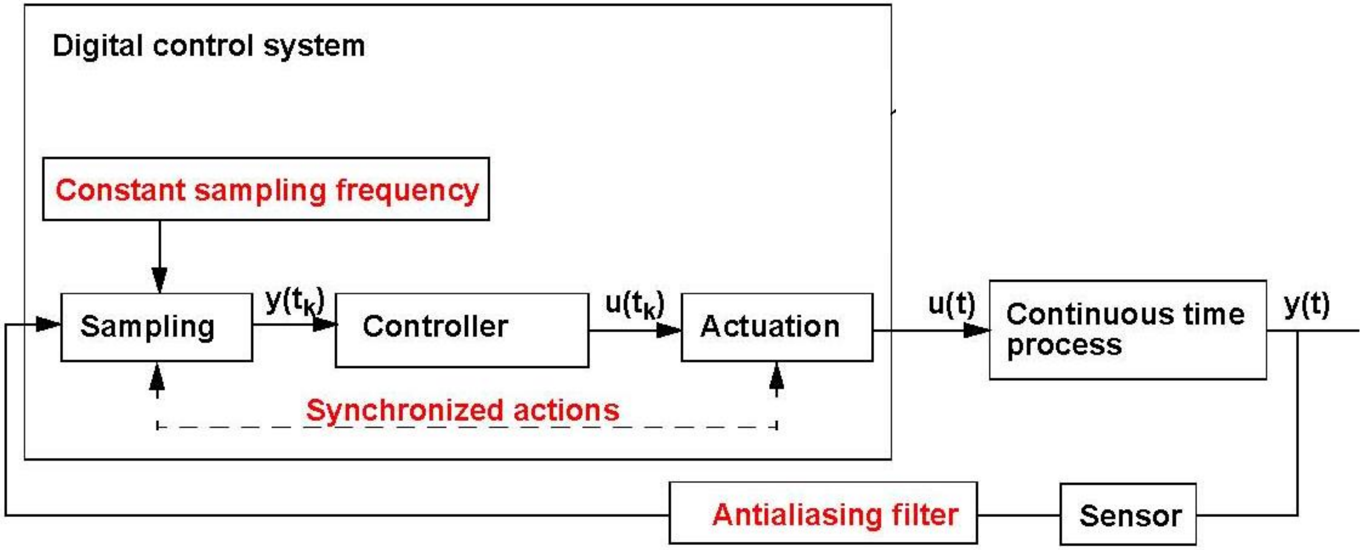
Outline

- Background and motivation
  - Industrial control systems perspectives
  - Gaps and misconceptions
- **Feedback control systems**
  - Assumptions, properties and timing problems
- Codesign
  - Methods
  - Abstractions
- Conclusions

Control systems and sampled control theory – recap

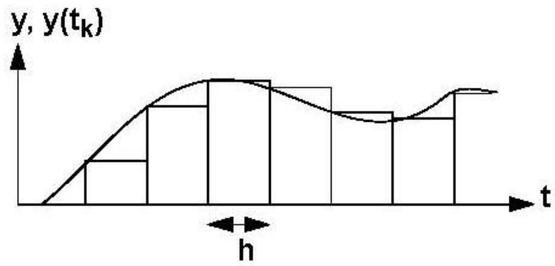
- Feedback provides robustness to disturbances
  - Sensitivity to sensor noise
  - Trade-off between performance and robustness
  - Delays are bad
- Control design in continuous or discrete-time
- Basic assumptions in sampled theory
  - Periodic sampling - external world synchronized with sampling instants
  - Zero order hold
  - Zero delay

Sampled data control systems perspective



Sampling interval $h = t_k - t_{k-1}$

Sampling frequency $f = 1/h$



Typical timing constraints

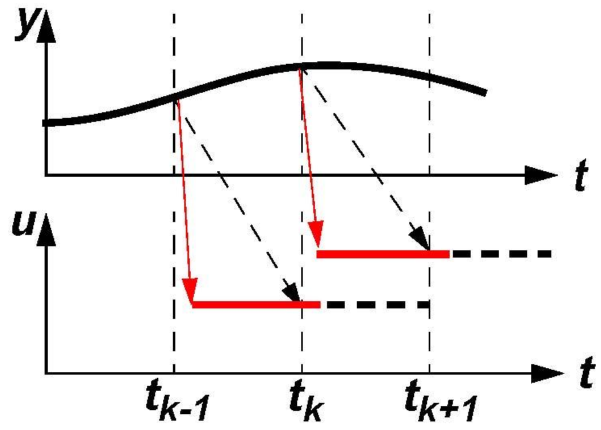
- Bounded or constant sampling-actuation delay
  - Delay jitter
- Sampling period
  - Period jitter
- Sampling and/or actuation synchronization
- Response time from external event to loop

Characterization

- Timescale (fast vs. slow); Sensitivity (tight vs. non tight), Critical vs. non critical, …

Sampled control theory – dealing with constant delays

- Solve the system equation



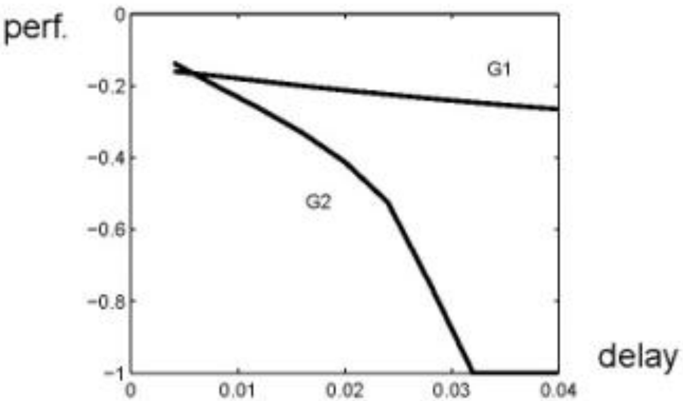
$$\frac{dx(t)}{dt} = Ax(t) + Bu(t-\tau), \qquad \tau < h$$

$$\begin{aligned} x(kh+h) - \Phi x(kh) &= \int_{kh}^{kh+h} e^{A(kh+h-s)} Bu(s-\tau)ds \\ &= \int_{kh}^{kh+\tau} e^{A(kh+h-s)} B\,ds\,u(kh-h) + \int_{kh+\tau}^{kh+h} e^{A(kh+h-s)} B\,ds\,u(kh) \\ &= \Gamma_1 u(kh-h) + \Gamma_0 u(kh) \end{aligned}$$

$$\begin{pmatrix} x(kh+h) \\ u(kh) \end{pmatrix} = \begin{pmatrix} \Phi & \Gamma_1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x(kh) \\ u(kh-h) \end{pmatrix} + \begin{pmatrix} \Gamma_0 \\ I \end{pmatrix} u(kh)$$

Constant delay compensation

- Delays
  - Limit control performance
  - Deteriorate phase margin; can lead to instability
  - Disturbance interpretation!
- Delay compensation is possible
  - Requires plant model and known delay
  - Removes dynamic effect!
  - Still degrades performance!



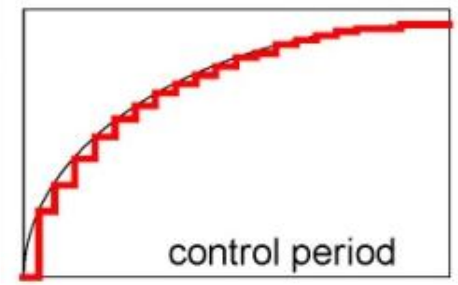
Controllers are bound by contracts to the ”plant”

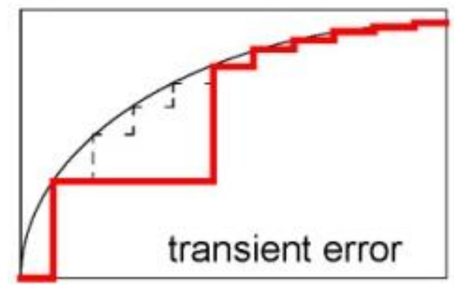
In the sense that

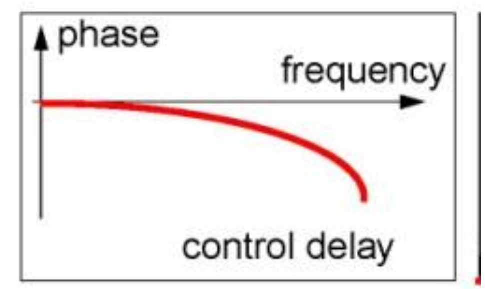
- Controller parameters will refer to closed loop system dynamics – determined by the plant dynamics
- Timing constraints refer to open and closed loop system, and usage scenarios
  - Controller parameters will be a function of the chosen sampling period
  - Will be a function of expected delay if compensation is included in controller
    - Shorter actual delay can be as bad as long delay

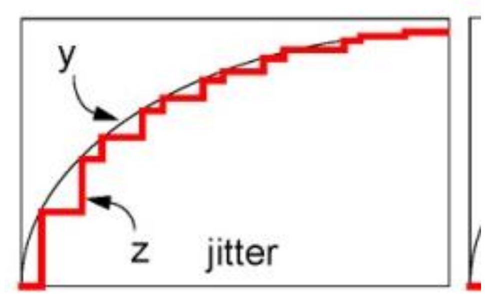
Timing problems

- Period
  - Dynamics of closed loop!
  - Response time!
  - Rules of thumb
- Transient errors / outage
  - Somewhat similar to delays/jitter
- Delays
  - Average is most important
- Jitter
  - Variations in delay and/or period
- Modes/operational dependence

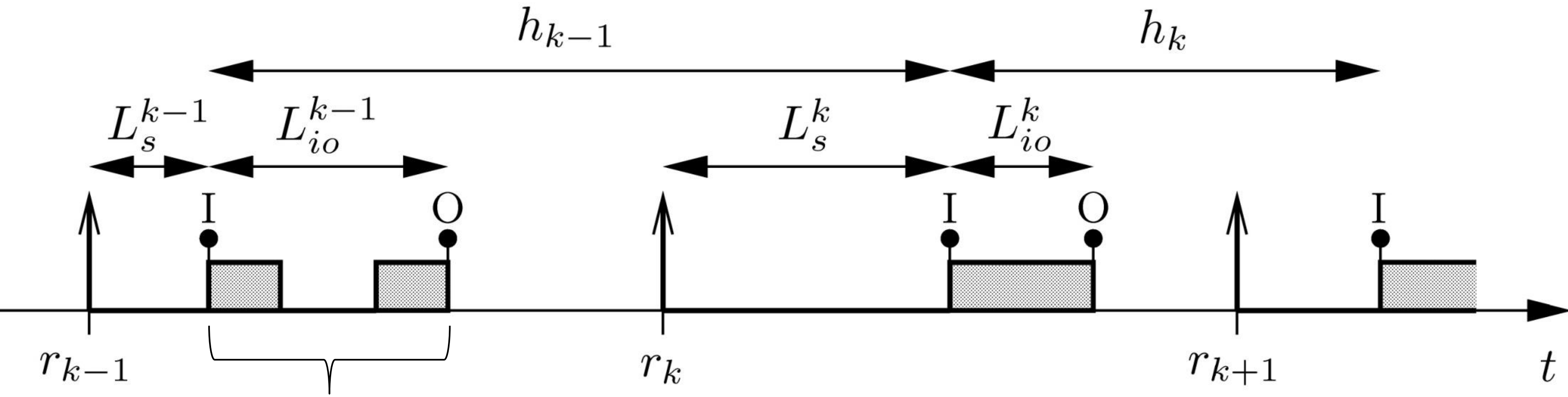








Time variations in embedded control systems



Output
- minimize!

Update
- sufficient during period

Note: Figure illustrates
- Time varying sampling period!
- Time varying delay!

Innovative Centre for Embedded Systems

Time variations – analysis

- Time varying delay and periods

  $$x(t_{k+1})=\Phi(h_k)x(t_k)+\Sigma\,\Gamma(h_k)u(t_{k-(nd-g)})$$

  - Assuming models of plants and variations
- Jitter often modelled as stochastic
  - As opposed to “known” jitter
  - Markov chains for describing switching – e.g. overruns
- Transient behavior
  - Simulation
  - Robustness norms – Max gains
- Stationary
  - Cost functions

    $$J=\int_0^\infty (x(t)^T Q_c x(t) + u(t)^T R u(t))dt$$
- Jitterbug toolbox in Matlab (Lund)

Rules of thumb and heuristics (examples)

- Jitter of 10% in period and delay
  - Ok for performance assuming otherwise correct design
- Extreme jitter appears to be worst
  - The maximum values occur when the jitter changes between the two extreme values only.
  - Relevant for both delay and sampling jitter
- Transient errors:
  - Need to deal with overruns and errors anyway
  - Design for worst case + exceptions OR something more flexible?

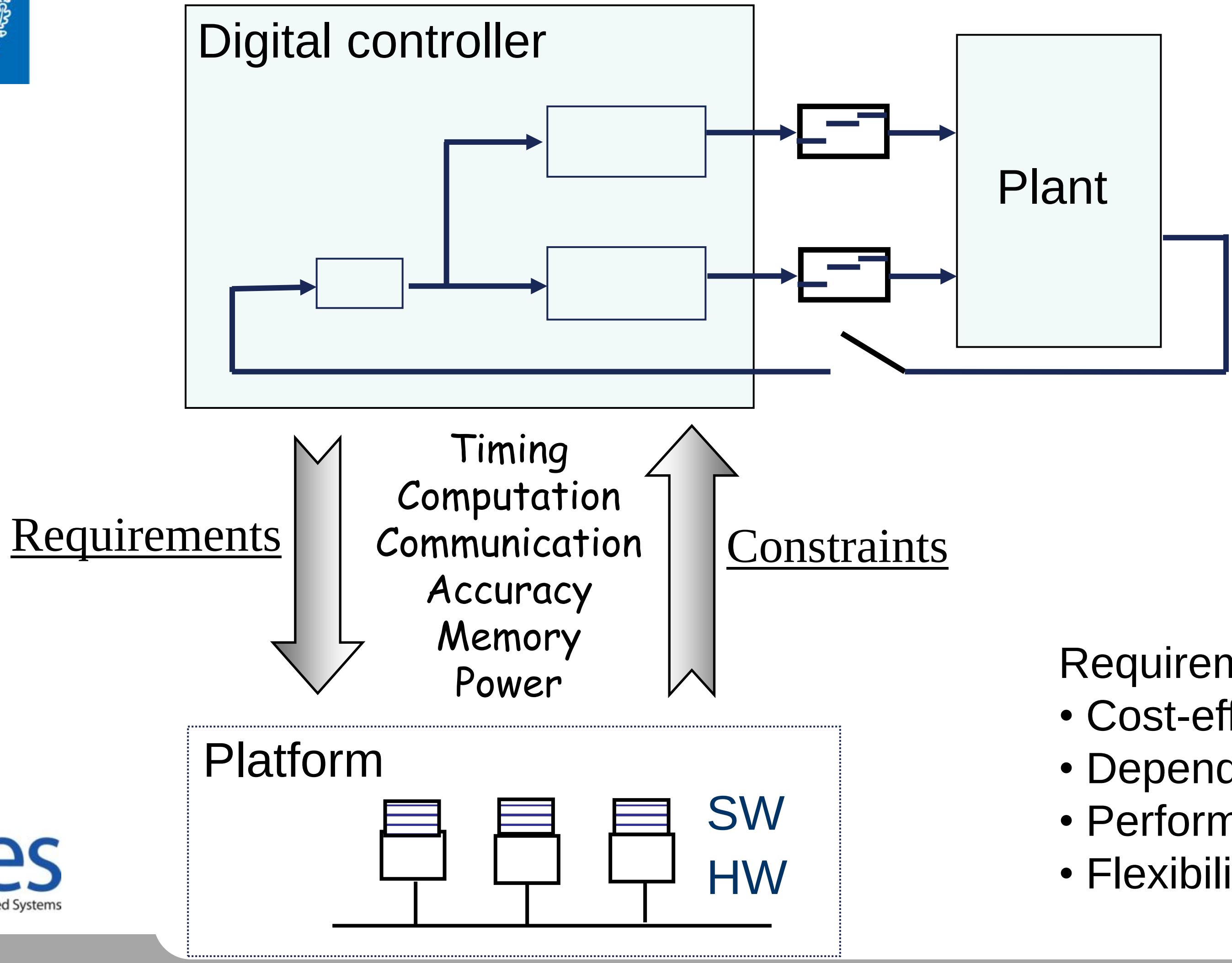
Complications (and interesting problems!)

- Many settings and aspects!
  - Controller design and plant characteristics
  - Compensation or not for constant delay?
  - Type of variation, in period or delay
  - Combined variations
  - Where delays enter – makes a difference!
- MIMO systems and synchronization
- Multirate systems
- Iterative (anytime) algorithms

Outline

- Background and motivation
  - Industrial applications
  - Gaps and misconceptions
- Feedback control systems
  - Assumptions, properties and timing problems
- **Codesign**
  - Methods
  - Abstractions
- Conclusions

What we want to achieve – codesign



Requirements:
- Cost-efficiency
- Dependability
- Performance
- Flexibility

Example co-design problem formulation: Control and Scheduling

"Given a set of systems to be controlled and a computer system, design a set of controllers and schedule them as real-time tasks such that the overall control performance is optimized"

- Objective function can be based on control performance, but it could also be a measure such as the end-to-end delay
- Many opportunities for optimization
- However, challenging dependencies,
  - E.g. accounting for delay and period in scheduling

Controllers and real-time constraints

- Desired dynamics (speed) will determine the sampling period

$$\sum \frac{Ci}{Ti} \leq 1; \; \mathbf{C_i < T_i << t_{rise}} \qquad (f_{bw} << f_i < 1/C_i)$$

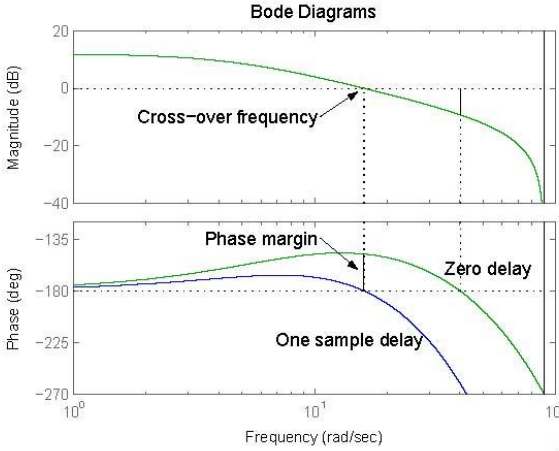
C = execution time,
T, f : sampling period & frequency
t_{rise} , f_{bw}: Closed loop system: rise time & bandwidth

- Closed loop system characteristics determines delay bound, τ_{max}

Discrete time: Check root locus, closed loop
Cont time: $\boldsymbol{\tau_{max} << \varphi_m / \omega_c}$

$$\mathbf{L_{io}(k) \leq \tau_{max} < R(k)}$$

L_{io} : Input output latency
R: “Output” task response time
φ_m , ω_c: phase margin, cross-over freq.



Deriving timing constraints in control design

- Specify a range of sampling periods
- Investigate tolerances to period jitter
  - Tolerances for jitter
- Investigate sensitivity to delays
  - Tolerances for constant delay
  - Max delay bound
  - Outage:
    - Max delay w.r.t. performance across cycles
    - Max delay w.r.t. stability
    - Safety may also be an issue

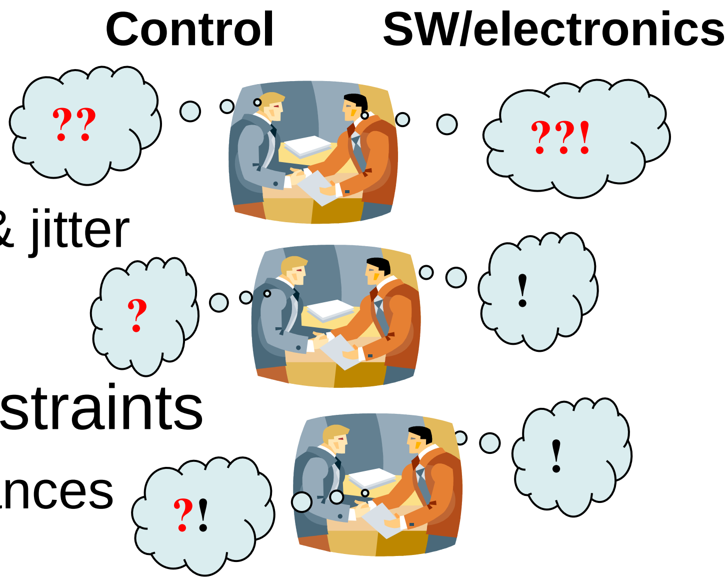
Example of relaxed timing constraints

- *Sampling instants:*
  - $t_{sample}(k+1) = t_{sample}(k) + T \pm tol_T$ Note: drift allowed
- *Actuation instants*
  - $t_{actuate}(k) = t_{sample}(k) + \tau \pm tol_{Tao}$ Note: Assuming constant delay

Alternative formulation for sampling instants:

- $t_{sample}(k+1) = kT + T \pm tol_T$ Note: without drift

Codesign illustration (timing only)



- ”Synchronous” model
  - Fixed period, zero delay & jitter
- Constant delay
- Inherent robustness constraints
  - Set of periods, jitter tolerances
  - Outage time constants
- Extended robustness design
  - Expected timing behavior as input to control design
  - Redesign of controller for robustness
  - Even on-line compensation possible
- Dynamic configuration (QoS)

Approaches for design – ”separation of concerns”

- ”HRT approach”
  - Synchronous approach or cyclic executive
  - Make sure sampling period is met and delay as small as possible
  - Predictable
  - Works well for single loops but more complicated for multi-rate systems and multiple functionalities
  - Pessimistic resource utilization (WCET bounds)
  - Potentially costly

Approaches for co-design

- Exploit robustness of controllers to get more flexibility in embedded systems design
- Design controller explicitly to be robust to delays
- Run-time compensation/estimation
  - Measurement based active compensation
  - Gain-scheduling and feed-forward from disturbances
  - Applicable to delays and lost data
  - Run-time knowledge of delays is important
- Exploit run-time adaption for on-line optimization of resource usage

Timing abstractions for control and codesign

- Concern / viewpoint?
  - What is the model supposed to support?

    Analysis and/or Synthesis?

    For example
    - Control analysis,
    - Code generation,
    - Execution/infrastructure generation,

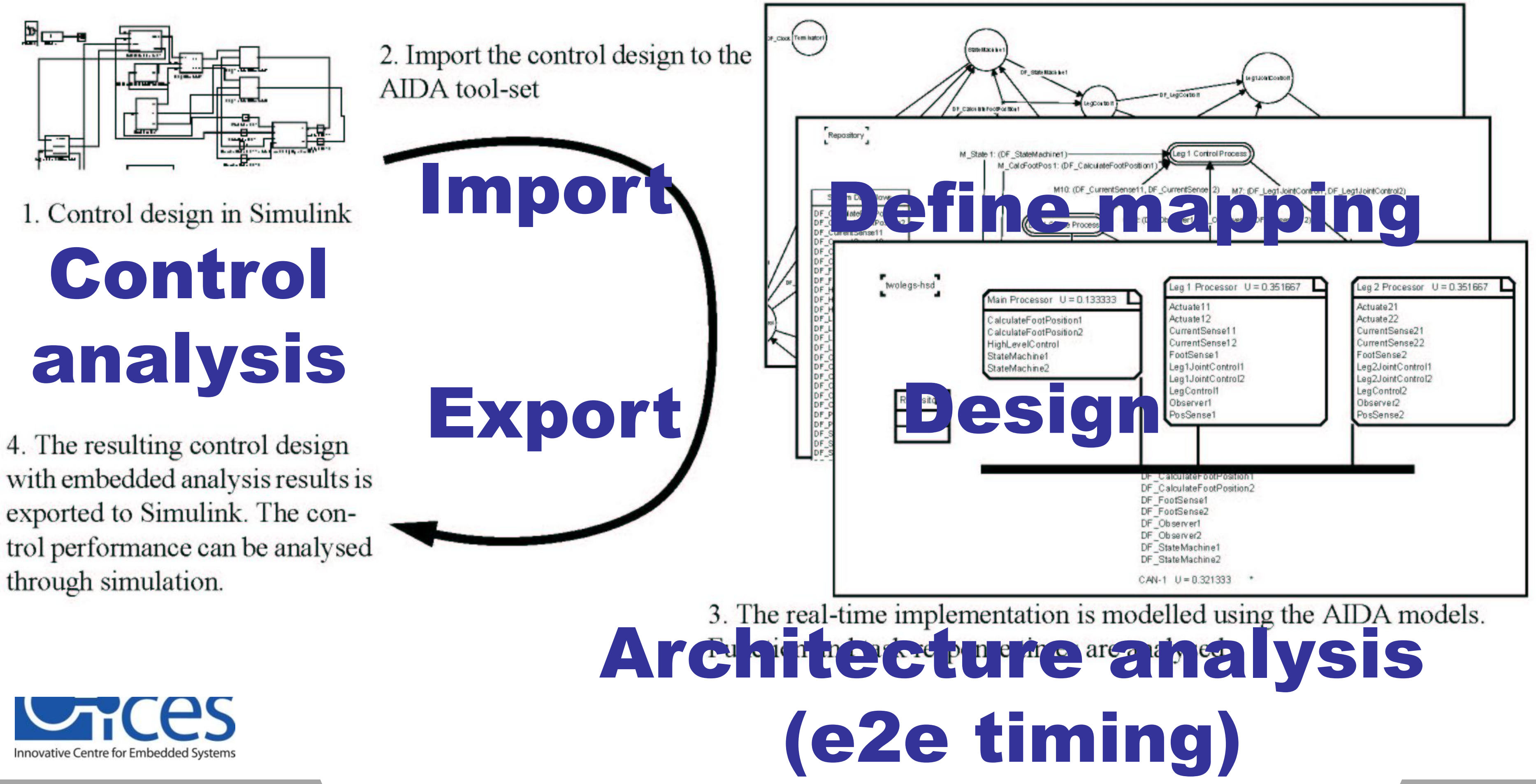
    Design parameters/Constraints (e.g. performance, power, etc.)

- Related questions
  - Which level of abstraction ?
  - Which formalism(s)?
  - Which tools?

KTH experiments

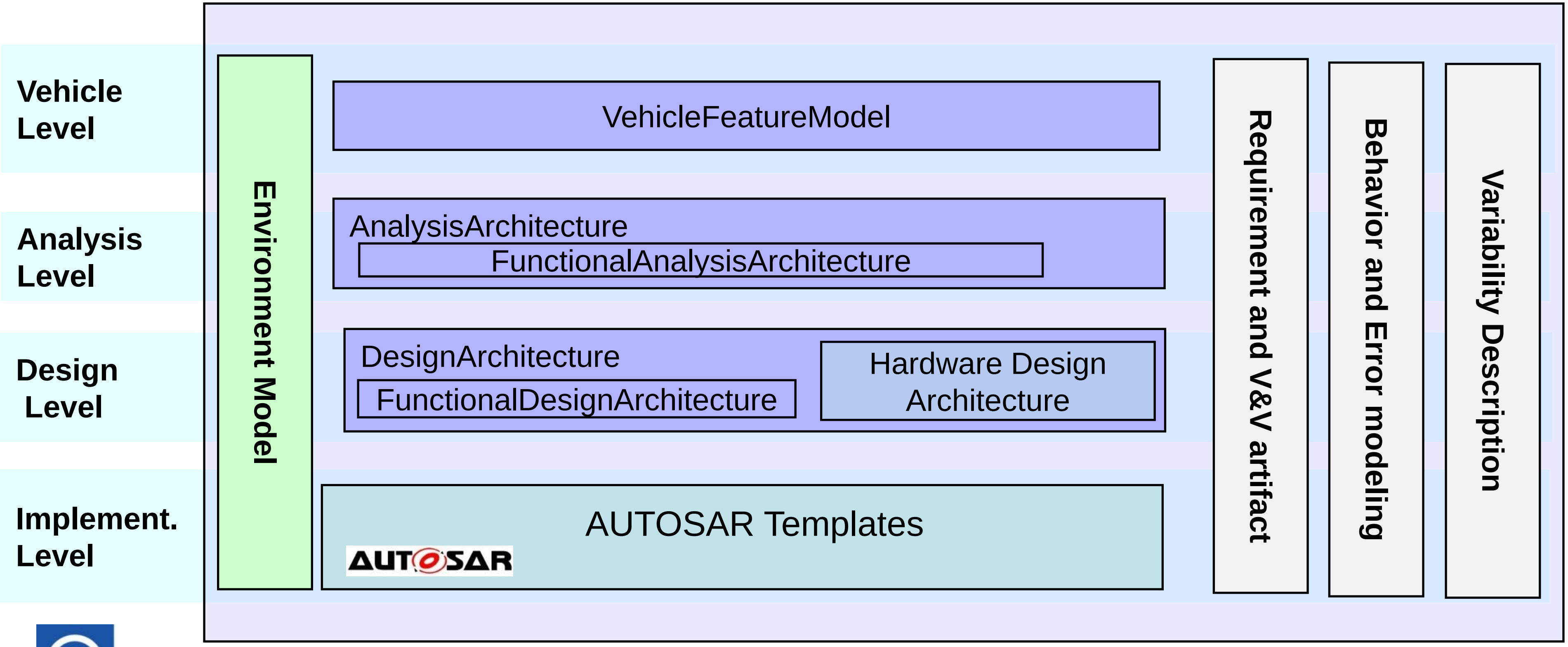
- Codesign environments: Control and embedded system in a combined model
  - XILO, Truetime (algorithms, resources, scheduling)
- Control design supported by "control abstractions” of embedded systems
  - AIDA (algorithms, delays, jitter, …)
- Multiview modeling
  - Multiple concerns: functions, timing, architecture, embedded system behavior, reliability/safety, ...
- East-adl
- Tool integration (tight vs. loose integration)

Example co-design tool-set: AIDA



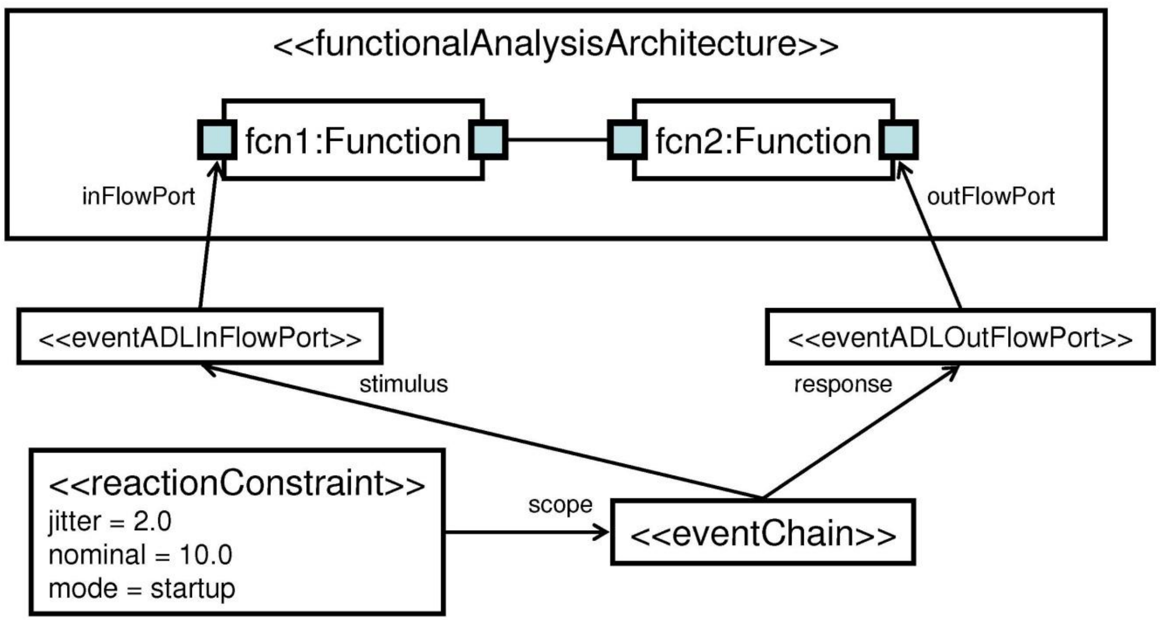
Innovative Centre for Embedded Systems

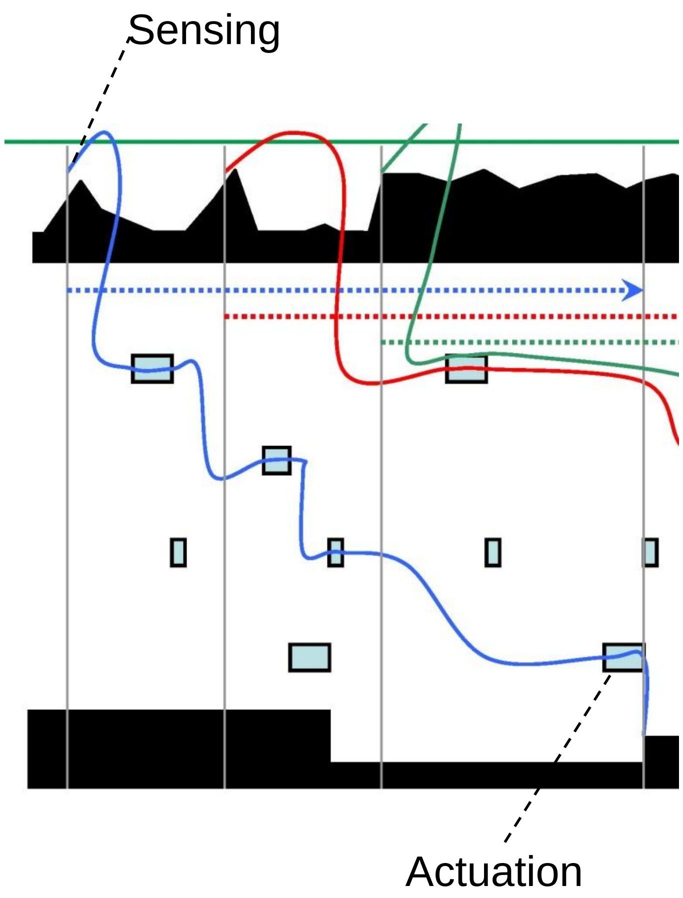
EAST-ADL – An Automotive ADL extending Autosar



EAST-ADL modeling concepts

Timing Augmented Description Language (TADL)





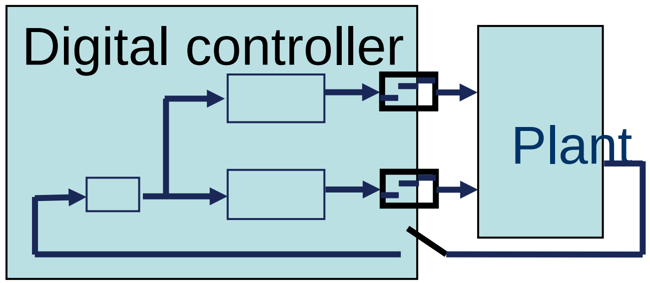
Courtesy of Volvo and the Timmo project

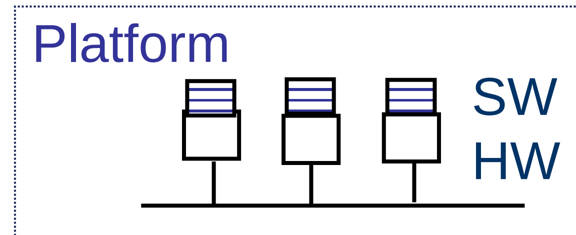
Timing abstractions for control and codesign - discussion

- Need to derive timing properties explicitly:
  - Period, delays, jitter, possible errors
- How to express the derived timing specifications from the control system?

  Options
  - Explicit blocks or annotations in the model, such as delays?
    - Used e.g. in AIDA for analysis, but not specificationss
  - Separate descriptions (separation of concerns)
    - Timing constraints
    - Dataflow, precedence
    - Algorithms

Timing problems and opportunities





Increasing industrial demand!

Decoupling desirable but difficult to achieve

- At a minimum, to avoid headaches, need a better cross understanding

Combined treatment important for

- Cost constrained embedded systems
- Mission and safety critical systems

Conclusions – Timing problems and opportunities

- Embedded control systems
  - Sampled data systems; Co-design
  - Case for codesign methods and tools!
- In-depth topics for further elaboration
  - Comparison and evaluation of different abstractions
    - Relate to EDA and computer science abstractions!
    - Characterization of timing problems
  - Co-design methods for embedded control
- Tentative follow up talks
  - East-ADL
  - GCDC/Scoop

Related PhD theses supervised or cosupervised by Martin Törngren

Enforcing Temporal Constraints in Embedded Control Systems, Kristian Sandström, 2001 (cosupervised)

Response Time Analysis for Implementation of Distributed Control Systems, Ola Redell, 2003 (main supervisor)

Topics in Modeling, Control, and Implementation in Automotive Systems, Magnus Gäfvert, 2003, (manager of the project in which the thesis was performed)

Quality of Control and Real-time Scheduling - Allowing for time-variations in computer control systems, Martin Sanfridson, 2004, (main supervisor)

Architecting and Modeling Automotive Embedded Systems, Ola Larses, 2005, (main supervisor)

A Model Management and Integration Platform for Mechatronics Product Development, Jad Elkhoury, 2006 (main supervisor)

Modeling and simulation of physical systems in a mechatronic context, Carl-Johan Sjöstedt, 2009, (main supervisor)

References (1)

- Rickard Svenningsson, Henrik Eriksson, Jonny Vinter and Martin Törngren. Model-Implemented Fault Injection for Hardware Fault Simulation. Models Workshop on Model-Driven Engineering, Verification and Validation (at the Models Conf., Oct. 3, 2010).
- Lei Feng, DeJiu Chen, Martin Törngren: Self Configuration of Dependent Tasks for Dynamically Reconfigurable Automotive Embedded Systems, at 47th IEEE CDC, Dec. 9-11, Cancun, Mexico, 2008.
- Martin Törngren, Årzen Karl-Erik, Henriksson Dan, Cervin Anton, Hanzalek Zdenek. Tools Supporting the Co-Design of Control Systems and Their Real-Time Implementation; Current Status and Future Directions. Proc. of the IEEE Int. Symposium on Computer-Aided Control Systems Design, TU Münich, Germany, October 4-6, 2006.
- DeJiu Chen, Torngren Martin, Shi Jianlin, Arzen Karl-Erik, Lonn Henrik, Gerard Sebastien, Stromberg Mikael, Servat David. Model Based Integration in the Development of Embedded Control Systems – a Characterization of Current Research Efforts. Proc. of the IEEE Int. Symposium on Computer-Aided Control Systems Design, 2006

References (2)

- Jad El-khoury, Ola Redell, Martin Törngren. Integrating Views in a Multi-view Modeling Environment. Proc. INCOSE International Symposium 2005. Rochester, NY. July 10-15. 2005.
- Martin Sanfridson, Martin Törngren, Jan Wikander. The effect of randomly time-varying sampling and computational delay. In Proceedings of the IFAC world congress, Prague, 2005.
- Karl Henrik Johansson, Martin Törngren, and Lars Nielsen. Vehicle Applications of Controller Area Network. In the Handbook of Networked and Embedded Control Systems, D. Hristu-Varsakelis and W. S. Levine (Eds.). 2005. ISBN: 0-8176-3239-5. A Birkhäuser book
- Ola Redell, Jad Elkhoury, Martin Törngren. The AIDA tool-set for design and implementation analysis of distributed real-time control systems. Microprocessors and Microsystems, Vol. 28, Issue 4, 2004, Pp. 163-182.
- Ola Redell and Martin Törngren. Calculating Exact Worst-Case Response Times for Static Priority Scheduled Tasks with Offsets and Jitter. Prof. Of 8th Real-Time and Embedded Technology and Applications Symposium – RTAS’02. IEEE Computer Press, Sept. 2002.

References(3)

- Redell O. and Sanfridson M., “Exact best-case response time analysis of fixed priority scheduled tasks“, Proc. of the 14th Euromicro Workshop on Real-Time Systems, Vienna, 2002.
- Jan Wikander, Martin Törngren and Mats Hanson (2001). Mechatronics Engineering - Science and Education, Invited Paper. IEEE Robotics and Automation Magazine, Vol 8, No. 2, 2001
- Jad Elkhoury and Martin Törngren. Towards a Toolset for Architectural Design of Distributed Real-Time Control systems. In Proc. of IEEE Real-Time Systems Symposium – RTSS, London, December 2001.
- Martin Törngren (1998). Fundamentals of implementing Real-time Control applications in Distributed Computer Systems. J. of Real-time Systems, 14, p. 219-250. Kluwer Academic Publishers
- Björn Wittenmark, Johan Nilsson and Martin Törngren (1995). Timing Problems in Real-time Control Systems: Problem Formulation. American Control Conference, June 1995, Seattle, Washington.
- Törngren Martin, Eriksson Christer and Sandström Kristian (1997). Deriving timing requirements and constraints for implementation of multirate control applications. Technical report 1997:1, Dept. of Machine Design, Royal Institute of Tech, Stockholm Sweden.

References(4) – EAST-ADL

- Philippe Cuenot, Patrik Frey, Rolf Johansson, Henrik Lönn, Yiannis Papadopoulos, Mark-Oliver Reiser, Anders Sandberg, David Servat, Ramin Tavakoli Kolagari, Martin Törngren, Matthias Weber. The EAST-ADL Architecture Description Language for Automotive Embedded Software. Invited chapter in the book Model-Based Engineering of Embedded Real-Time Systems. Holger Giese, Bernard Rumpe, Bernard Schätz (eds). Series: Lecture Notes in Computer Science. Vol. 6100. Springer Verlag, 2010. ISBN: 978-3-642-16276-3
- Anders Sandberg, DeJiu Chen, Henrik Lönn, Rolf Johansson, Lei Feng, Martin Törngren, Sandra Torchiaro, Ramin Tavakoli-Kolagari, Andreas Abele. Model-based Safety Engineering of Interdependent Functions in Automotive Vehicles Using EAST-ADL2. Safecomp 2010.

Innovative Centre for Embedded Systems