Timing properties
of
computer controlled systems

Real-time Computer Control Systems, spring 2002

Martin Sanfridson
Mechatronics Lab
Machine Design
KTH
email: mis@md.kth.se

Intro — Mapping the timing properties

Timing problems

- Response time
- Execution time
- Blocking
- Scheduling
- Arbitration
- Transmission time
- Output jitter
- Offset
- Utilization
- Clock skew
- Vacant sampling
- .
- etc.

}

Control delay
Choice of control period
Jitter
Transient error

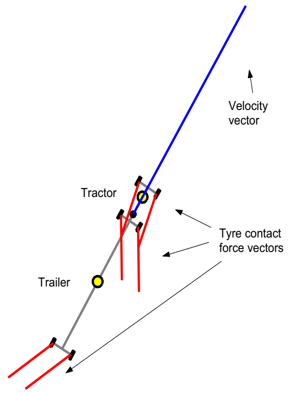
VDC — Vehicle Dynamics Control, an example

- *Assists* the driver in difficult situations by quick responses.

- Dangerous situations: skidding, over-turn, jack-knife.
- Sensors: yaw rate, lateral acceleration, wheel speed, driver's intentions.
- Actuators: *braking on individual wheels* (and possibly steering).

VDC — To state the control problem

- Specification of control problem: What is considered to be good, satisfying and bad?
- Question: How does the timing properties of the computer system affect control performance and stability?



Basics — The complete design process

1. Design of the *mechanical system*.
2. Choice of sensors and actuators and additional hardware.
3. *Modelling* of the system and disturbances.
4. *Linearization* and system identification.
5. Analysis of dynamic properties including simulation.
6. Control requirement specifications.
7. Control design.
8. Implementation of the algorithms on the computer system.

Basics — Continuous time model

- A *linear time invariant system*, a differential equation (example)

$$\ddot{y}(t) + a_1\dot{y}(t) + a_2 y(t) = \dot{u}(t) + b_1 u(t)$$

- can be written in a state space form

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

- or as a transfer function

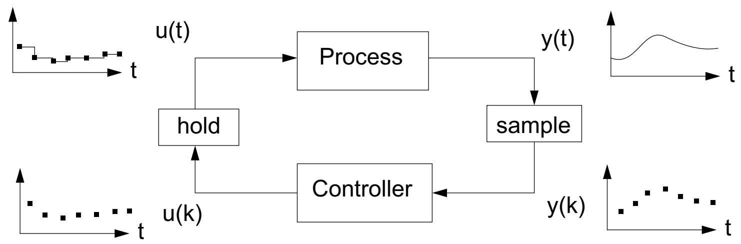
$$Y(s) = G(s)U(s)$$

with

$$G(s) = C(sI - A)^{-1}B + D = \frac{b_1 + s}{s^2 + a_1 s + a_2}$$

Basics — Sampling of a continuous process

- Advantages: digital signal processing, digital communication.



- A/D conversion. ZOH circuit, zero-order-hold.

Basics — Properties of sampling

- Intersample behaviour.
- Signals with frequency lower than the Nyquist frequency $\omega_s/2$ can be reconstructed after sampling.
- Analog anti-aliasing filter.
- The effect of s&h can for high sampling rates be approximated as

$$\frac{1}{h}\frac{1 - e^{-sh}}{s} \approx \frac{1 - 1 + sh - (sh^2)/2 + \dots}{sh} = 1 - \frac{sh}{2} + \dots$$

which is the Laplace transform of a delay of h/2.

- Quantization error.

Basics — Discrete time model

- A *linear time invariant system*, a difference equation (example)

$$y(kh) + a_1 y(kh-h) + \ldots = b_1 u(kh-h) + \ldots$$

- can be written in a state space form

$$x(kh+h) = \Phi x(kh) + \Gamma u(kh)$$
$$y(kh) = Cx(kh) + Du(kh)$$

- or as a transfer function

$$Y(z) = G(z)U(z)$$

with

$$G(z) = C(zI - \Phi)^{-1}\Gamma + D = \frac{b_1}{z + a_1}$$

Basics — Translation continuous to discrete

- Alternatives:

1. *Discrete* analysis and synthesis, or
2. *Continuous* analysis and synthesis, followed by discretization.

- Some of the physical insight is lost when leaving the differential equations in favour of the difference equations.
- Discrete vs continuous: deadbeat control, time continuous counterpart.

Basics — Analysis methods

- Analysis of both the open loop system and the controlled system.
- Controllability, observability.
- Stability and performance.
- Domains:
  - Time domain: e.g. step and ramp responses, steady state errors, or
  - Frequency domain: Bode plots for amplitude and phase, Nyquist and Nichols diagrams.
- Damping and natural frequency are important characteristics.

Basics — Synthesis methods

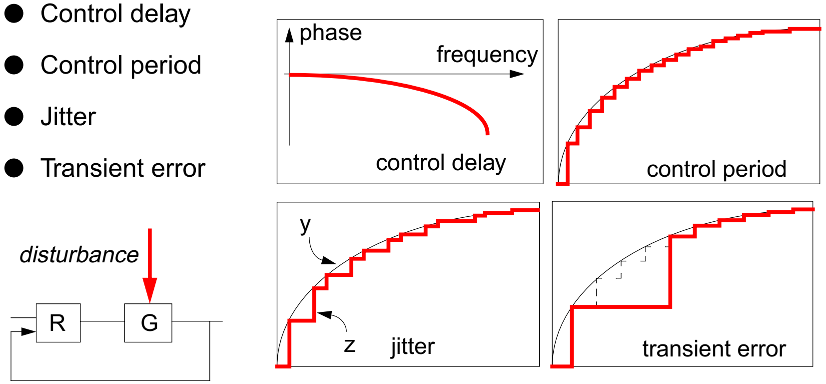
- Pole placement
  - Pole placement by state feedback
  - Polynomial design
  - IMC - internal model controller
- Optimality
  - LQG - linear quadratic gaussian
  - Robust control
  - MPC - model predictive controller

Basics — Additional synthesis

- Observer: The complete state vector is estimated based on measurement of some states. The observer is a dynamic system and the observer design problem relates to selecting the dynamics of state reconstruction.
- Additional filters might be needed.
- Be aware of the limitations for nonlinear systems.

Timing Properties — From a control viewpoint

- Control delay
- Control period
- Jitter
- Transient error



- Combinations possible. MIMO systems.

Timing Properties — Demo

- Two processes, a first order (controlled by a PID-controller)

$$G_1(s) = \frac{1}{s+1}$$

- and a double integrator (controlled by a deadbeat controller)

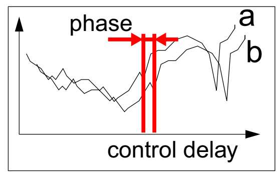
$$G_2(s) = \frac{1}{s^2}$$

- Performance index a relative difference:

$$I = \sqrt{\frac{\int e_1^2 - \int e_2^2}{\int e_1^2 + \int e_2^2}} signum\left(\int e_1^2 - \int e_2^2\right)$$

Timing Properties — Control Delay I

- Definition
  Any constant delay in the control loop, from sensor to actuator.
- Caused by



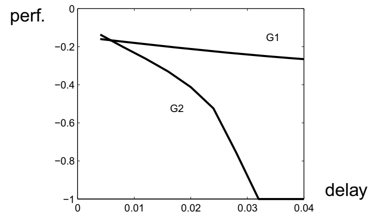
  - ✗ Delays in a distributed computer system are caused by: S&H-circuits, execution of control algorithms, communications, execution/communication overhead and bad synchronization.
  - ✗ A delay in a process itself is often caused by mass transportation in combination with the physical location of sensors.

Timing Properties — Control Delay II

- Effect on performance and stability

$$G(s) = e^{-\tau s}$$

The bandwidth of the closed loop system is upper bounded by $1/\tau$.

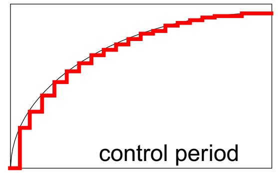


- Control analysis and synthesis
  - Constant delay is well-known from control theory. Delays can not be removed by any kind of controller. The total delay should always be as small as possible.
  - Analysis and synthesis can be performed in the time discrete domain.

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

Timing Properties — Choice of period I

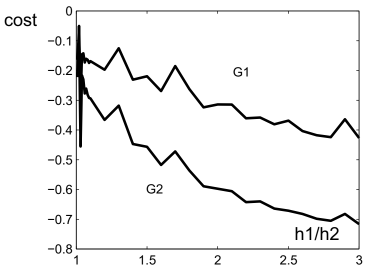
- Definition
  The sampling or control period determines the rate of actions.



- Choice of period
  - The period must be chosen based on the dynamics of the closed loop system.
  - There is a potential conflict between the available computational resources and the needs of the controller.

Timing Properties — Choice of period II

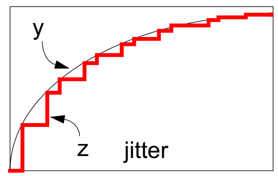
- Effect on performance and stability. The general rule is: the faster sampling the better performance.



- Control analysis and synthesis
  - The choice is often based on a rule-of-thumb. Intersample behaviour must be taken into account.
  - The magnitude of the control signal depends critically on the sampling period for some type of controllers.
  - Trade-off when more than one controller need a limited resource.
  - A multi-rate system can be analysed with a lifting technique.

Timing Properties — Jitter I

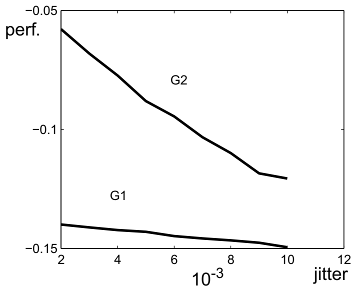
- Definition
  Time variations in a specified time interval.
  Two kinds: ***period jitter*** *and* ***delay jitter****.*
  $t_k = t_k + \tau_k$ or $t_k = kh + \tau_k$



- Caused by
  The process itself can cause variations in deadtime. Examples of jitter caused by a computer system:
  - Hardware: cache memory.
  - Software: branching in the code gives variable execution time.
  - Scheduling of tasks: e.g. blocking and interference.

Timing Properties — Jitter II

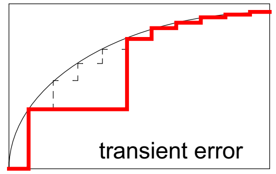
- Effect on performance and stability
  In general, jitter degrades the control performance. Vacant sampling is a possible outcome.



- Control analysis and synthesis
  - ✗ Not much in the control theory; jitter is often neglected in the analysis.
  - ✗ Jitter is often modelled as stochastic (as opposed to “known” jitter). Some results on controller that takes stochastic jitter into account exist.

Timing Properties — Transient error I

- Definition
  A timing error caused by a fault with a transient duration. One or more samples are lost.

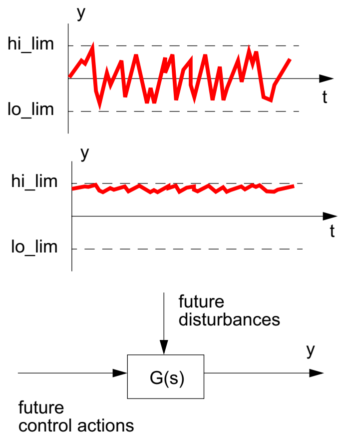


- Caused by

  An example is that a message is lost or corrupted or delivered too late.

Timing Properties — Transient error II

- Effect on performance and stability
  The control action is temporarily suspended, which may lead to a catastrophe.



- Control analysis and synthesis
  - ✗ Not much done here from a control point of view.
  - ✗ The marginal depends e.g. on the current situation, thus is not constant.
  - ✗ Reconfiguration of the control algorithm in case of partial permanent failure (graceful degradation).

Timing Properties — Alternative views

- Traditional boundary between control design and computer implementation:
  - ✗ Negligible delay jitter,
  - ✗ Negligible period jitter,
  - ✗ Constant sampling interval, and
  - ✗ No transient errors.

Timing Properties — Lesson learned

- Traditional view:
  When implementing the controller, try to minimize delay, jitter and transient errors.
- Opportunity view:
  Use the knowledge of the timing properties to improve the overall behaviour of the system, e.g. flexibility (future proof).

Timing Properties — compensation

- Compensation for static delay.
- Compensation for delay jitter
- Compensation for transient error.
- Compensation for varying period.

CCS — Computer Control Systems

- Fast enough for the dynamics of the controlled system and for changes in the environment:

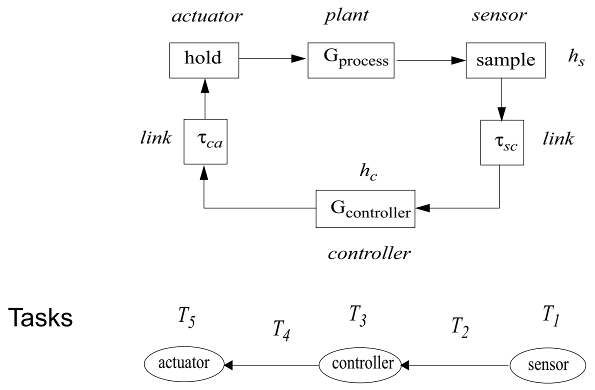
  A control system is a real-time system!

- Basic real-time properties, find
  - ✗ Execution time and other delays.
  - ✗ Sampling period.
- Requirements not only on control performance but also on robustness and dependability.

CCS — Design task

- Choice of architecture, e.g. distributed systems.
- Control structure.
- Translating the control system (equations) to code.
- Triggering and scheduling. Mapping code segments to threads.
- Additional functions e.g. for error handling and diagnostics.

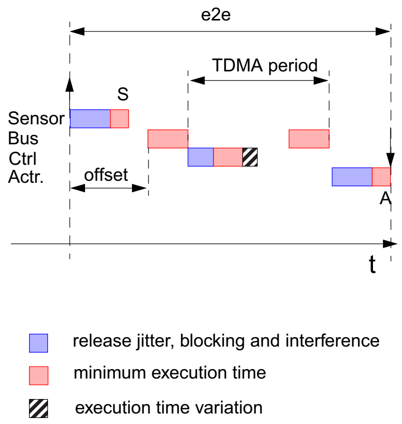
CCS — Model of a distributed system



- Examples of mapping onto the timing properties.

CCS — Scheduling of tasks and messages

- Off-line scheduling — dynamic scheduling
- Task model
  - Period, P
  - Worst case execution time, C
  - Deadline, D
  - Release time, A
  - Priority, p
- Utilization
- Global clock



CCS — Synchronization

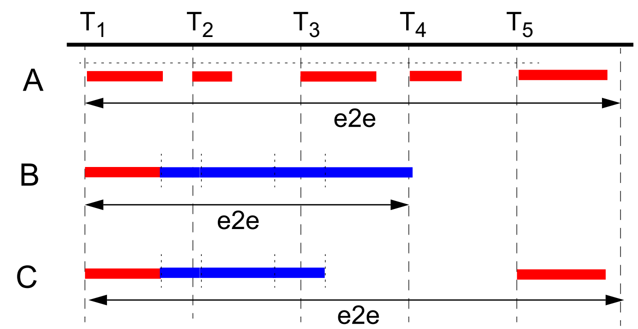
Triggering of the task chain (T_1, T_2, T_3, T_4, T_5)

Sensor	Link	Controller	Link	Actuator
TT	TT	TT	TT	TT
TT	TT	TT	TT	ET
TT	TT	ET	TT	TT
TT	TT	ET	TT	ET
TT	ET	TT	ET	TT
TT	ET	TT	ET	ET
TT	ET	ET	ET	TT
TT	ET	ET	ET	ET

- Messages and tasks: control signals, process events, etc.

CCS — Off-line vs. on-line

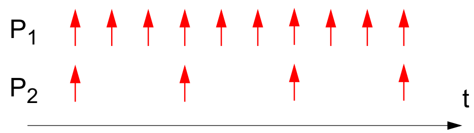
- What is best for control, off-line (static) or on-line (priority driven)?



- Skew between two tasks.

CCS — Multirate sampling

- Subsystems with different dynamics.
- Can usually be analysed by *lifting*.
- It is natural and saves processing time.



CCS — Implementation of the control task I

- A PI-controller:

$$u = K \cdot e + K_i \cdot \int e$$

where e is the reference minus the output, $e = r - y$.

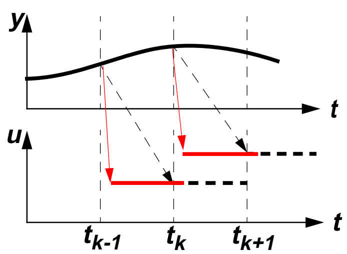
- The discrete PI-controller is (using Tustin's transformation):

$$u_k = u_{k-1} + K \cdot (e_k - e_{k-1}) + \frac{h}{2} \cdot K_i \cdot (e_k + e_{k-1})$$

- The parameters of the control law depend on the sampling period.

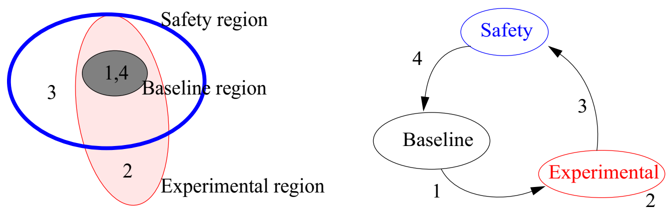
CCS — Implementation of the control task II

```
read sensor y
e := r - y
u := u_old + K(e - e_old) + h/2*Ki(e + e_old)
write output u
e_old := e
u_old := u
```



CCS — To cope with transient errors

- The "simplex architecture" uses several controllers.



- Fault tolerant scheduling, ghost tasks.

Integration — Performance index

- Quadratic performance index in continuous time...

$$J = E \int_{kh}^{(k+1)h} x(t)\bar{Q}_1 x^T(t) + u^T(t)\bar{Q}_2 u(t) dt$$

- ...to be translated to discrete time, augmenting the state vector:

$$z_k = \begin{bmatrix} x_k & x_k^c & u_{k-n} & \dots & u_k \end{bmatrix}^T$$

- First calculate the covariance, $P_k = E(z_k z_k^T)$, then performance.
- Enables comparison between: period, delay and stochastic jitter.

Integration — Opportunities

- Account for static delay and delay jitter.
- Graceful degradation.
- Quality-of-service.
- Event triggered sampling.
- *New heights with flexibility!*