

EL2450 Hybrid and Embedded Control

Spring 2019

Dimos Dimarogonas

<http://people.kth.se/~dimos/>

**Decision and Control Systems
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology**

EL2450 Hybrid and Embedded Control

- **Disposition**

7.5 credits, 28h lectures, 28h exercises, 3 homework

- **Instructors**

Dimos Dimarogonas, lecturer, dimos@kth.se

David Umsonst, teaching assistant, umsonst@kth.se

Dirk van Dooren, teaching assistant, dirkvd@kth.se

Christos Verginis, teaching assistant, cverginis@kth.se

Course goal

Participants should gain insight into computer implementation of control algorithms, and realize the need to integrate control and computer engineering in the design of networked embedded control systems

Milestones

After the course, you should be able to

- Analyze, design and implement sampled-data control systems
- Characterize possibilities and limitations of real-time operating systems through mathematical models
- Appreciate flexibilities and compensate for uncertainties in networked control systems
- Apply hybrid systems modeling and analysis techniques to embedded systems

EL2450 Hybrid and Embedded Control

Lecture 1: Introduction

- Practical Information
- What is Hybrid and Embedded?
- Motivating example
- Course outline
- Review of sampled signals

Course Information

- All info available at
`https://kth.instructure.com/courses/7591`

Material

- **Reading Material:** Book chapters and papers, reading per lecture found at course homepage
- **Lecture notes:** Available online after each lecture
- **Exercises:** Class room and home exercises
- **Homework:** 2 computer and one laboratory exercises
- **Software:** Matlab etc (see homepage)

Homework need to be submitted on time (no exceptions)

What is an Embedded System?

Computer system with the computer **embedded** in the application/physical process

Contrast with general-purpose and desk-top computers



Example: Design of VDC System

Vehicle dynamics control¹ (VDC) systems assist car driver in oversteering, under-steering and roll-over situations

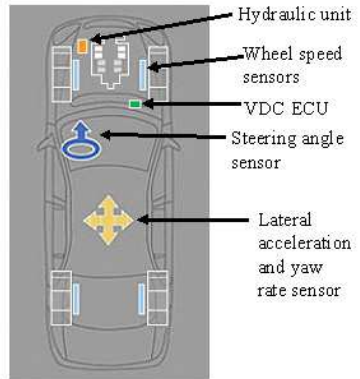
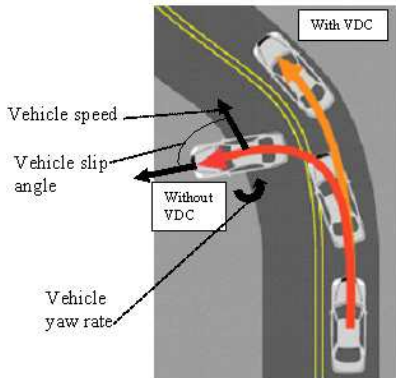
System design flow

1. Product specification: control objectives
2. Architecture definition: control structure, communication
3. Software development: control algorithms, filters
4. Physical implementation

¹Also known as electronic stability program, dynamic stability control, or active yaw control.

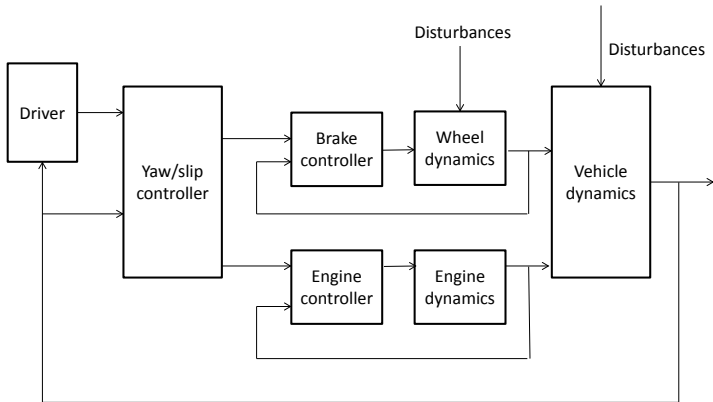
VDC Over-Steering

1. Friction limits reached for rear wheels
2. Car starts to skid, i.e., yaw rate (*girvinkel*) and side slip angle deviate from driver's intended
3. VDC detects emerging skidding and computes compensating torque
4. Applies braking force to outer front wheel



VDC Control Architecture

VDC is a cascade control structure with three controllers:

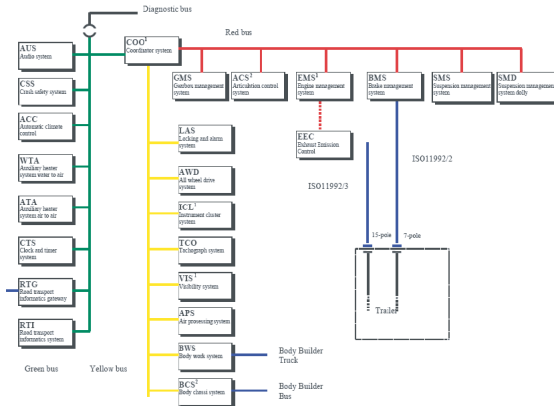


VDC Communication Architecture

- VDC is a networked embedded control system
- Utilizes the Controller Area Network (CAN) bus
- CAN is a communication bus that connects sensor, actuator and controller nodes
- A modern car has several such electronic control units

Scania Truck CAN

A Scania truck has three CAN buses



Characteristics of Embedded Systems

- Computational systems (but not a computer)
- Integrated with physical world via sensors and actuators
- Reactive (at the speed of the environment)
- Heterogeneous (mixed hw/sw architectures)
- Networked (share data and resources)

...

Relevance of Embedded Systems

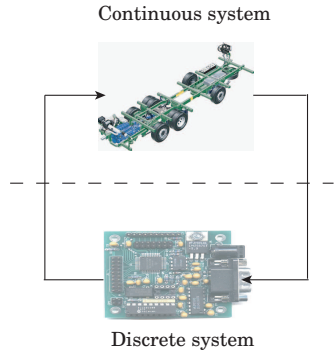
- Dominates computer market completely ($\approx 98\%$)
- Increasing interest in multi-disciplinary research fields

Examples:

- Transportation: modern car has several computers and networks
- Robotics: service robots, mobile manipulators
- Manufacturing and process industry
- Power generation and distribution

What is a Hybrid System?

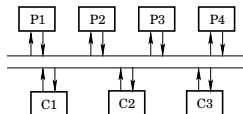
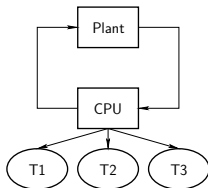
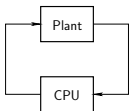
- A hybrid system is a mathematical model of an embedded system
- Dynamical system with interacting time-driven and event-driven dynamics



The Main Thread of the Course

Integrated design of computer-controlled systems

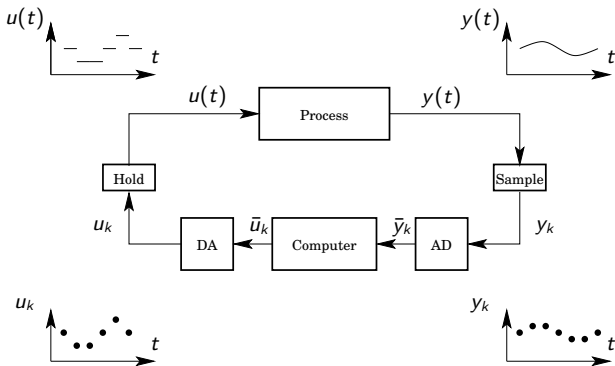
1. Single-task
(time-triggered)
2. Multi-task
(event-triggered)
3. Hybrid
(networked)



Course Outline

- **Introduction:** course outline, motivating examples, review of sampled signals [L1]
- **Time-triggered control:** models, analysis, implementation [L2-L5]
- **Event-triggered control:** real-time operating systems, models of computations and software, scheduling [L6-L9]
- **Hybrid control:** modeling time-triggered and event-triggered systems, control and verification of hybrid systems [L10-L13]
- **Summary** [L14]

Computer-Controlled System



Signals in the loop have different characteristics

Signals

Continuous-valued and continuous-time signals:

- $u(t) \in \mathbb{R}, t \in [0, \infty)$
- $y(t) \in \mathbb{R}, t \in [0, \infty)$

Continuous-valued and discrete-time signals:

- $u_k \in \mathbb{R}, k = 0, 1, \dots$
- $y_k \in \mathbb{R}, k = 0, 1, \dots$

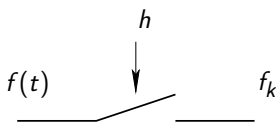
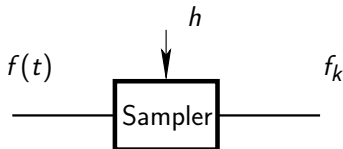
Discrete-valued and discrete-time signals:

- $\bar{u}_k \in \{\bar{u}^1, \dots, \bar{u}^N\} \subset \mathbb{Q}, k = 0, 1, \dots$
- $\bar{y}_k \in \{\bar{y}^1, \dots, \bar{y}^N\} \subset \mathbb{Q}, k = 0, 1, \dots$

Sampling

Uniform sampling with sampling period $h > 0$ generates

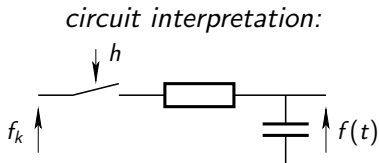
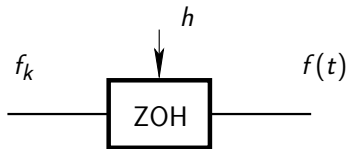
$$f_k = f(kh), \quad k = 0, 1, \dots$$



Zero-Order Hold

ZOH *holds* the input constant over intervals h :

$$f(t) = f_k, \quad t \in [kh, kh + h)$$



“Zero”: interpolation by polynomial of order zero.
Exist also first-order and higher-order holds

Fourier Transforms

$F(\omega)$ is Fourier transform of continuous-time signal $f(t)$:

$$F(\omega) = \int_{-\infty}^{\infty} e^{-i\omega t} f(t) dt, \quad f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega t} F(\omega) d\omega$$

$F_s(\omega)$ is Fourier transform of discrete-time signal $f_k = f(kh)$:

$$F_s(\omega) = \sum_{k=-\infty}^{\infty} f_k e^{-i\omega k}, \quad f_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i\omega k} F_s(\omega) d\omega$$

Shannon's Sampling Theorem

A continuous-time signal $f(t)$ with Fourier transform

$$F(\omega) = 0, \quad \omega \notin (-\omega_0, \omega_0)$$

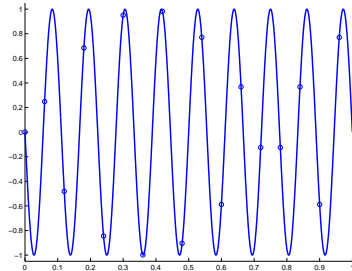
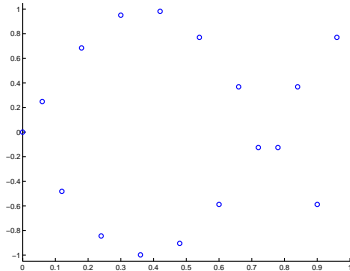
is uniquely reconstructed by its samples $f_k = f(kh)$, if sampling frequency $\omega_s = 2\pi/h$ is larger than $2\omega_0$. Then, $f(t)$ can be reconstructed from f_k as

$$f(t) = \sum_{k=-\infty}^{\infty} f_k \operatorname{sinc} \frac{\omega_s(t - kh)}{2}$$

Note: Shannon reconstruction is not casual, ZOH practical alternative

Example: Sinusoid

Reconstruction possible if sampled at least twice per period:



Nyquist Frequency

$\omega_N = \omega_s/2$ is called the Nyquist frequency

Shannon's Sampling Theorem:

If $f(t)$ has no component above ω_N , then $f(t)$ is uniquely determined by its samples f_k

What happens when $\omega_s < 2\omega_0$? (equiv. $\omega_N < \omega_0$)

High frequencies of $f(t)$ are interpreted as low frequencies when not sampled sufficiently often

This is called aliasing or frequency folding

Aliasing

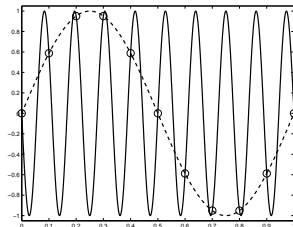
Components above ω_N in $f(t)$ cannot be distinguished in f_k

$$F_s(\omega) = \frac{1}{h} \sum_{k=-\infty}^{\infty} F(\omega + k\omega_s)$$

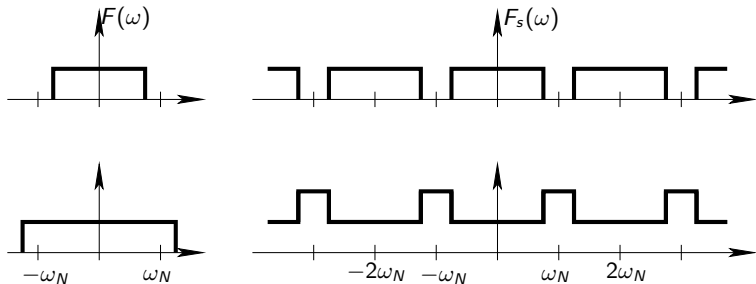
Example:

Frequencies $k\omega_s \pm \omega_1 > \omega_N$ of $f(t)$ are mapped into lower frequencies $0 \leq \omega_1 < \omega_N$ in the sampled signal.

$0 \leq \omega_1 < \omega_N$ is called *alias* of $k\omega_s \pm \omega_1$



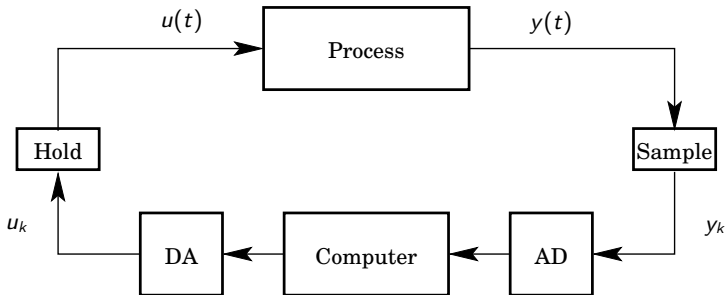
Aliasing: Frequency domain visualization



The original signal can be reconstructed from $F_s(\omega)$ in the first case with a low pass filter with appropriate cut-off frequency, since $F(\omega)$ does not contain frequencies over ω_N . This is in contrast to the second figure.

Discrete-Time Systems

We need tools to analyze discrete-time system $u_k \rightarrow y_k$:



Discrete-Time Models

Many results similar to continuous-time systems

- Input-output models: $y_{k+1} + ay_k = bu_k$
- State-space model: $x_{k+1} = ax_k + bu_k, \quad y_k = x_k$
- z-transform: $Y(z) = \sum_{k=0}^{\infty} y_k z^{-k}$
- (Pulse-)Transfer function $H(z)$ with $Y(z) = H(z)U(z)$

Please, revisit the Signals & Systems Course (or similar)

Next Lecture

Models of sampled systems

- Sampling of continuous-time systems
- State-space and input-output models
- Poles and zeros