Communications and Control
An introduction to Cyber Physical Networking
12th ITG SCC, Rostock, Germany
Feb. 11, 2019

Wolfgang Kellerer
based on joint work with
Sandra Hirche (ITR)
Onur Ayan (LKN)
Markus Klügel (LKN)
Vahid Mamduhi (ITR)
Touraj Soleymani (ITR)
Mikhail Vilgelm (LKN)
Samuele Zoppi (LKN)

DFG SPP 1914 project „Optimal Co-Design of Wireless Resource Management and Multi-loop Networked Control“

©2019 Technical University of Munich

This work is supported by the DFG Priority Programme 1914 Cyber-Physical Networking grant number KE1863/5-1
Motivation: 5G Vision

- **eMBB**: enhanced Mobile BroadBand
- **mMTC**: massive Machine-Type Communication
- **URLLC**: Ultra Reliable Low Latency Communication

Source: 3gpp.org
Not all aspects are needed for all services.

Motivation: 5G Vision

- Enhanced mobile broadband
- Massive machine type communications
- Ultra-reliable and low latency communications

Source:
It is mostly machines that communicate over networks.
Motivation: Control

- Networked Machines → Networked Cyber Physical Systems (NET CPS)

*Control matters*

- Multiple control loops
- High Reliability
- Scalability
- Low latency

Industry 4.0, Tesla Factory

Robot cooperation
[http://iridia.ulb.ac.be/~mathews](http://iridia.ulb.ac.be/~mathews)

Trucks platooning, Scania
[www.scania.com](http://www.scania.com)
Motivation: Cyber Physical Networking

- Key challenge in design and analysis of cyber-physical systems: **Control over shared communication networks**
  - quality of control may be degraded due to the congestion while accessing the scarce communication resources

- **Cyber Physical Networking**: joint consideration of control and networking concepts to improve the system performance
  - possibly involving
    - all network layers (cross-layer design, …)
    - all communicating nodes between devices (edge computing, …)
    - multiple control loops with different control strategies
Focus of this tutorial

- Support of control over shared communication networks

- Focus on
  - **Communication**: Medium Access Control (MAC)
  - **Control**: multi-loop networked control system (NCS), all control loops share a communication network
Outline

- System model: Networked Control System
  - Including a short primer on control

- Selected use cases and results
  - Decentralized wireless MAC & Control: Adaptive Random Access
  - Scheduled wireless access & Control: Age of Information vs. Value of Information

- NCS experience for everybody:
  Intro to NCS benchmark platform

- … with a break in between
DFG SPP 1914 Cyber Physical Networking

- **DFG Priority Programme Cyber-Physical Networking (SPP 1914)**
  https://www.spp1914.de/

- **Understanding the fundamental trade-offs** btw. communication and control systems
  - **Fundamental limits** for communication latency, reliability, efficiency, and control performance including the role of feedback/side information
  - **Joint analysis methods and joint optimisation metrics** defining the interfaces
  - **Mathematical models** and analysis of interacting communication and control dynamics

- **Design methods for horizontal/vertical coordination and control**, surpassing the limitations of today's abstraction
  - **Co-design and adaptive feedback mechanisms** for control and protocols over unreliable communication channels such as wireless
  - **Distributed control** and communication in large-scale systems
  - **Latency-aware horizontal/vertical coordination**: interfaces, integration of network, operating system and applications
DFG SPP Cyber Physical Networking

Project areas

- Cooperative control and networking for wireless networks (e.g., topology control, consensus-based control, multi-agent systems, event-based c.)
- Co-design of control and networking/communications (e.g., information exchange between control and networking, model predictive CPN)
- Higher layer network aspects (e.g., latency, resilience-aware networking, co-designed architecture for in-network control)
- Performance measurements and modeling

in interdisciplinary teams of control/automation and communication/network experts

https://www.spp1914.de
Networked Control Systems
Networked Control Systems

- Machine-to-Machine: Sensing & Actuation
- Control systems, coupled via communication networks

→ Networked Control Systems

The following system model is based on the view of the DFG SPP 1914 Cyber-Physical Networks project „Optimal Co-Design of Wireless Resource Management and Multi-loop Networked Control“ (Hirche, Kellerer)
Cross-Layer Design Framework

- Optimal Network & Control – Global Optimization Problem

- Control and network protocols: distributed solutions to global OP
Scenario & Problem Formulation

- $N$ stochastic Linear Time Invariant (LTI) systems
  \[ x_{k+1}^i = A^i x_k^i + B^i u_k^i + w_k^i \]

- Colocated Controller - (Actuator) - Plant

- Plant state is sensed remotely, e.g., camera

- Shared network: blocking / collisions / packet errors

\[
\theta_k^i = \begin{cases} 
1, & \text{if OK} \\
0, & \text{otherwise}
\end{cases}
\]
Excursion: Quick Introduction to Control (1)

- Control: use of algorithms & feedback in engineering systems; usually for dynamic system

- Dynamic system: a system whose behavior changes over time, often in response to external stimulation

Based on a tutorial given by Sebastian Trimpe, MPI für Intelligente Systeme, 2018.
Quick Introduction to Control (2)

- **Feedback:** two (or more) dynamic systems connected such that they influence each other

![Diagram of a closed loop system with feedback](image)

- **Control System:** design a dynamic system “the controller“ (= system 2) to influence the process (= system 1) in a desired way
Quick Introduction to Control (3)

- Control System
  - design a dynamic system “the controller“ (= system 2) to influence the process (= system 1) in a desired way
  - modern control systems: controller is an algorithm running on a computer

![Control System Diagram]

- system 1: process or dynamic system
- system 2: controller
- computer with control algorithm/control law: transforms $y \rightarrow u$
Typical representations of Dynamic Systems

(a) continuous time

\[ x(t): \text{system state} \]

Differential equation:

\[ \dot{x}(t) = \frac{dx}{dt} = f(x(t), u(t), d(t)) \]
\[ y(t) = x(t) \]

(b) discrete time

Differential equation

\[ x_{k+1} = f(x_k, u_k, d_k) \]
\[ y_k = x_k \]

Here: discrete-time linear time-invariant (LTI) systems

\[ x_{k+1} = Ax_k + Bu_k + w_k \]
Quick Introduction to Control (5)

Discrete-time Linear Time-Invariant (LTI) stochastic Networked Control Systems (NCS)

\[ x_{k+1} = Ax_k + Bu_k + w_k \]
\[ x[k + 1] = Ax[k] + Bu[k] + w[k] \]

- \( k \in \{0,1,2, \ldots \} \) discrete time-step
- \( x \in \mathbb{R}^n \): system state, \( A \in \mathbb{R}^{n \times n} \): state matrix
- \( u \in \mathbb{R}^m \): control input, \( B \in \mathbb{R}^{n \times m} \): input matrix
- \( w \in \mathbb{R}^n \): random noise vector

Special: 1-dim.
- \( x[0] = 0 \)
- \( A = 1 \in \mathbb{R}^{1 \times 1} \)
Quick Introduction to Control (6)

- Networked Control System (NCS)

![Diagram showing the components of a networked control system](attachment://networked控制系统.png)

1. Process or dynamic system 
2. Controller
3. Sensors
4. Actuator
5. Communication network

**System 1:** Process or dynamic system

**System 2:** Controller

**Operator input**

**Computer with control algorithm/control law:** transforms $y \rightarrow u$
Scenario & Problem Formulation

Generalized optimization problem:
• with control and scheduling/link access policies as optimization problem variables

\[
\max_{f,y} QoC(x, u, \delta) \quad \text{s.t. } \sum_{s \in S} \delta_l(s) \nu_s \leq C_l \quad \text{and } x_{k+1} = Ax_k + Bu_k + w_k
\]
Scenario & Problem Formulation

- **Dead-beat control law**
  (linear discrete-time control: feedback $\rightarrow$ stable state)
  \[ u^i_k = -L_i E[x^i_k | Z^i_k], \]
  with \( Z^i_k = \{ z^i_o, \ldots, z^i_k \} \) and \( L_i \) - arbitrary stabilizing feedback gain

- **Model-based estimation** (if \( \theta^i_k = 0 \) i.e. communication failed):
  \[ E[x^i_k | Z^i_k] = (A_i - B_i L_i) E[x^i_{k-1} | Z^i_{k-1}] \]

- **Network Induced Error (~estimation error)** [MTH15]
  \[ e^i_{k+1} = (1 - \theta^i_k) A_i e^i_k + w^i_k \]
Scenario & Problem Formulation

- Network Induced Error (~estimation error) [MTH15]
  \[ e_{k+1}^i = (1 - \theta_k^i)A_i e_k^i + w_k^i \]

\rightarrow Separation of Control and Communication problems

Two application examples:
(1) Decentralized wireless MAC & Control
(2) Scheduled wireless access & Control (up-/downlink scheduling)
Outlook: Global Optimization Problem

Generalization of the above problem

- Multi-loop
- Single-hop → Multi-hop
  - Base station (2 hop, central)
  - Multiple hops (wireless and wired)
- MAC → Multi-layer
  - Routing (topology, node buffering)
  - Transport (TCP congestion control)
- Network functions
  - Edge computing (location/migration of controller)

=> computationally very hard to solve – decomposition needed
Global Optimization Problem

- **Cost function**

  \[ J_i = \lim_{K \to \infty} \frac{1}{K} \mathbb{E} \left[ \sum_{k=0}^{K-1} x_k^i \mathbf{Q}_x x_k^i + u_k^i \mathbf{Q}_u u_k^i \right] \]

  \[ \min_{\psi, \pi, \varphi, \xi} \sum_i w_i J_i \]

  \[ \text{s.t. } \psi \in \Psi, \varphi \in \Phi, \pi \in \Pi, \xi \in \Xi \]

  - \( \psi \): congestion control law from the admissible set \( \Psi \)
  - \( \pi \): scheduling law from the admissible set \( \Pi \)
  - \( \varphi \): sampling law from the admissible set \( \Phi \)
  - \( \xi \): control law from the admissible set \( \Xi \)

- Nodes are linked according to the topology \( Q \)
- Action set \( A \)
- Transmission determined by a choice of \((Q;A)\)
Outline

- System model: Networked Control System
  - Including a short primer on control

- Selected use cases and results
  - Decentralized wireless MAC & Control: Adaptive Random Access
  - Scheduled wireless access & Control: Age of Information vs. Value of Information

- NCS experience for everybody:
  Intro to NCS benchmark platform
Decentralized wireless MAC & Control: Adaptive Random Access
Adaptive Random Access: Scenario

- Adaptive decentralized MAC for Event-Triggered NCS

- LTI control loop
  \[ x^{i}_{k+1} = A_ix^i_k + B_iu^i_k + w^i_k, \]

- State dynamics \( \rightarrow \) estimation error dynamics
  \[ e^{i}_{k+1} = (1 - \theta^i_k)A_ie^i_k + w^i_k. \]

- Local scheduler: event-based with threshold \( \Lambda_i \)

- Decentralized medium access with \( M_k \) channels
  - \timeslot = control period
  - uniform choice of the channels
  - collision occurs if the same channel is chosen
  - channel feedback: collision \((1,0), M_k\)

Adaptive Random Access: Scenario

- Adaptive decentralized MAC for Event-Triggered NCS

- LTI control loop
  \[ x_{k+1}^i = A_i x_k^i + B_i u_k^i + w_k^i, \]

- State dynamics $\rightarrow$ estimation error dynamics
  \[ e_{k+1}^i = (1 - \theta_k^i) A_i e_k^i + w_k^i. \]

- Local scheduler: event-based with threshold $\Lambda_i$

- Decentralized medium access with $M_k$ channels
  - timeslot $\Rightarrow$ control period
  - uniform choice of the channels
  - collision occurs if the same channel is chosen
  - channel feedback: collision (1,0), $M_k$

Adaptive Random Access: Threshold-based Triggering

- Event-triggered NCS and Multichannel Slotted ALOHA
  - Communication delay $\approx$ connection establishment delay
- Threshold-based event triggering:

\[ P[\delta^i_k = 1|e^i_k] = \begin{cases} 
0, & \text{if } ||e^i_k|| \leq \Lambda_i \\
1, & \text{otherwise}
\end{cases} \]

with $\delta^i_k$ (local) scheduling variable.

- Successful reception: $\theta^i_k = \delta^i_k \gamma^i_k$ with

\[ P[\gamma^i_k = 1|\delta^i_k = 1] = \left(\frac{M_k - 1}{M_k}\right)^{g_k} \]

Adaptive Random Access: Initial Evaluation

- Given $N$ subsystems with $A_i$, $W_i$, and $M_k$ channels
  - Network performance depends on control loop & $\Lambda_i$
  - Control loop performance depends on network & $\Lambda_i$
    - Metric: variance of an estimation error

Performance Evaluation: Threshold

- Network and control performance are *coupled via the threshold*
- If the threshold is set too low, performance degrades drastic due to *collision*
- If the threshold is set too high, performance degrades slowly due to *underutilized network*
- Always exists a threshold (global), for which control and network performance are optimal

→ to optimally use the network, adaptive scheduling policy is required

Adapting to varying number of channels – *network state*

Adaptive Random Access: Adaptation gain

- Adaptive choice of the threshold based on available channels

\[ \Lambda' = f(M), \]

- Relative gain from adaptation depends on the variability of the number of channels

Scheduled wireless access and control: 
Age of Information 
vs. Value of Information

„Age-of-Information vs. Value-of-Information Scheduling for Cellular Networked Control Systems“
Scheduled wireless access: Scenario

- $N$ stochastic LTI control loops share the same network
- Centralized scheduler in Base Station (BS) determines UL and DL transmissions

Plant $N$ is observed by Sensor $N$
Scheduled wireless access: Scenario

- $N$ stochastic LTI control loops share the same network
- Each sub-system consists of sensor $S_i$, controller $C_i$ and plant $P_i$
- Observed plant state $x_i[k_i]$ at time-step $k_i$ is transmitted towards $C_i$
  - First on uplink (UL) from $S_i$ to base station (BS)
  - Then on downlink (DL) from BS to $C_i$
- Only the latest generated measurement is stored in the packet queue
- Centralized scheduler determines UL and DL transmissions

How to distribute (schedule) the UL and DL resources among the sub-systems (control loops)?
Challenge: two-hop communication system

- Central scheduler has to consider the importance of a sensor value to decide for scheduling considering both hops

- Possible “importance“ metrics:
  - Delay → **Age of Information (AoI)**
  - Meaning of content of sensor value → **Value of Information (VoI)**

- We compare both in this example: *Age-of-Information vs. Value-of-Information Scheduling for Cellular Networked Control Systems*
Age of Information (AoI)

- a recently proposed performance metric that measures information freshness at the destination node
- proposed in 2011 by S. Kaul and R. Yates for vehicular networks [1,2]
  - [1]: “Average end-to-end (application-to-application) delay observed in any vehicle’s state”
  - [3]: “Time since last update was received”

- Age of Information $\Delta(t)$:
  $$\Delta(t) = t - u(t)$$

- $t$: current time
- $u(t)$: time-stamp of the most recent update

Value of Information (VoI)

- deals with the **content** of a new update independently of its timeliness
- VoI stems from information theory (Shannon)
- The amount of reduction in the uncertainty of a stochastic process at the recipient

![Diagram of Value of Information](image)

**Value-of-Information**

deals with the **content** of a new update independently of its timeliness

**Age-of-Information**

deals with the **freshness** of a new update independently of its content
Back to our scenario

- $N$ stochastic LTI control loops share the same network
- Each sub-system consists of sensor $S_i$, controller $C_i$ and plant $P_i$
- Observed plant state $x_i[k_i]$ at time-step $k_i$ is transmitted towards $C_i$
  - First on uplink (UL) from $S_i$ to base station (BS)
  - Then on downlink (DL) from BS to $C_i$
- Only the latest generated measurement is stored in the packet queue
- Centralized scheduler determines UL and DL transmissions

How to distribute (schedule) the UL and DL resources among the sub-systems (control loops)?
Recap: Stochastic LTI Networked Control Systems

• as also before:

Discrete linear time-invariant (LTI) stochastic NCSs are modeled as:

\[ x[k + 1] = A \cdot x[k] + B \cdot u[k] + w[k] \]

\( k \in \{0, 1, 2, \ldots \} \) discrete time-step
\( x \in \mathbb{R}^n: \) System state, \( A \in \mathbb{R}^{n \times n}: \) State matrix
\( u \in \mathbb{R}^m: \) Control input, \( B \in \mathbb{R}^{n \times m}: \) Input matrix
\( w \in \mathbb{R}^n: \) Random noise vector
Network Model

- Faster networking than control $\Rightarrow T_i^s \geq t \quad \forall i$
- UL/DL schedules $\pi_{UL/DL}(t) \in \{0, 1\}^N$
  - $\pi_i^{UL/DL}(t) = 1 \iff$ sub-system $i$ transmits at $t$
- $\mathcal{R}^{UL}$, $\mathcal{R}^{DL}$ set of UL and DL resources
  - $|\mathcal{R}^{UL}| = R^{UL} < \infty$
  - $|\mathcal{R}^{DL}| = R^{DL} < \infty$
  - $\mathcal{R}^{UL} \cap \mathcal{R}^{DL} = \emptyset \iff$ Frequency-Division Duplex (FDD)
- Reception at the end of the slot

$R^{UL}$: Number of UL resources (per slot)
$R^{DL}$: Number of DL resources (per slot)
$T_i^s$: Sampling period of the $i$-th sub-system
Control Model (1)

- Stochastic LTI control systems:
  \[ x_i[k_i + 1] = A_i \cdot x_i[k_i] + B_i \cdot u_i[k_i] + w_i[k_i] \]
  with \( x_i[0] = w_i[0] \) and \( w_i \sim \mathcal{N}(0, W_i) \).

- Periodic sampling with sampling period \( T_i^s \) slots with initial sampling \( T_i^o \sim U(0, T_i^s) \)

- Stairwise system evolution:
  \[ k_i(t) = \left\lfloor \frac{t - T_i^o}{T_i^s} \right\rfloor \]

- Sampling events at slots \( \{k \cdot T_i^s + T_i^o\}, k \in \mathbb{N} \) \( \Rightarrow \) TX-Buffer update at sensor \( S_i \)
Control Model (2)

- Packet reception indicator variable $\delta_i[k_i] \in \{0, 1\}$:

$$z_i[k_i] = \begin{cases} x_i[k_i], & \text{if } \delta_i[k_i] = 1 \\ \emptyset, & \text{if } \delta_i[k_i] = 0. \end{cases}$$

- Information set $\mathcal{I}_i[k_i]$ available at $C_i$:

$$\mathcal{I}_i[k_i] = \{k_i, z_i[0], \ldots, z_i[k_i], u_i[0], \ldots, u_i[k_i - 1]\}$$

- State estimation at $C_i$:

$$\hat{x}_i[k_i] = \mathbb{E} [x_i[k_i] \mid \mathcal{I}_i[k_i]]$$

- Control input:

$$u_i[k_i] = -L_i \hat{x}_i[k_i]$$

state feedback gain matrix $L_i$
Age of Information and Value of Information

- **Age-of-Information:**
  \[ \Delta_i(k_i) = k_i - s_i[k_i] \]
  with \( s_i[k_i] = \sup\{s \in \mathbb{N} : s \leq k_i, z_i[s] \neq \emptyset\} \leftrightarrow s_i[k_i]: \text{Generation time of the most recent received information} \]

- **Estimation error:**
  \[ e_i[k_i] = x_i[k_i] - \hat{x}_i[k_i] \]

- **Value-of-Information:**
  \[ E \left( \| e_i[k] \|^2 \right) = \begin{cases} 0, & \text{if } \Delta_i[k] = 0 \\ g(\Delta_i[k]), & \text{if } \Delta_i[k] > 0 \end{cases} , \]
  with:
  \[ g(\Delta_i[k]) \triangleq \sum_{r=0}^{\Delta_i[k]-1} \text{tr} \left( (A_i^T)^r A_i^T W_i \right) \]
  \( \text{tr}(.): \text{Trace operator} \)

- **Expected value of squared estimation error**

**AoI = time difference to sensor value generation time**

**Vol = expected value of squared estimation error**
System Dependability of Vol

- Vol depends on plant dynamics (system matrix A)
- $A < 1$: sub systems tend to stability / $A > 1$: plant dynamics require control

\[ E[\| e_i[k] \|^2] = \begin{cases} 
0 & \text{if } \Delta_i[k] = 0 \\
g(\Delta_i[k]) & \text{if } \Delta_i[k] > 0 
\end{cases} \]

\[ g(\Delta_i[k]) \triangleq \sum_{r=0}^{\Delta_i[k]-1} \text{tr} \left( (A_i^T)^r A_i W_i \right) \]
Value-of-Information on UL / DL

Assumption 1. The scheduler at the BS observes the content of any packet it receives on the UL.

Assumption 2. The scheduler is aware of system parameters $A_i, W_i, B_i, L_i, T^{s}_i, T^{o}_i, \forall i$

- Reception variable $\delta_i[k_i] = \{0, 1\}$
- Age-of-Information $\Delta_i[k_i]$ available at BS:
  - $\Delta_i[k_i] \leq \Delta_i[k_i]$
- Information set $I^\beta_i[k_i]$ available at BS:
  - $I^\beta_i[k_i] \supseteq I_i[k_i] \forall i, k_i$
- Analogously:
  $$e_i^\beta[k_i] = x_i[k_i] - \hat{x}_i^\beta[k_i]$$
  $$\hat{x}_i^\beta[k_i] = f(\Delta_i^\beta[k_i], I^\beta_i[k_i])$$
  $$\mathbb{E}\left[\|e_i^\beta[k_i]\|^2\right] = g(\Delta_i^\beta[k_i])$$
Value-of-Information on UL / DL

- Value of UL packets:

\[ v_i^{UL}(t) = \mathbb{E} \left[ \| e_i^B[k_i] - e_i^S[k_i] \|^2 \right] \]
\[ = \mathbb{E} \left[ \| e_i^B[k_i] \|^2 \right] \]

with \( k_i = k_i(t) \) and sensing error \( e_i^S[k_i] = 0 \).

- Value of DL packets:

\[ v_i^{DL}(t) = \mathbb{E} \left[ \| e_i[k_i] - e_i^B[k_i] \|^2 \right] \]
\[ = \| \hat{x}_i^B[k_i] - \hat{x}_i[k_i] \|^2 \]

- UL Scheduling:

\[
\max_{\pi_i^{UL}(t)} \sum_{i=1}^{N} \pi_i^{UL}(t) \cdot v_i^{UL}(t)
\]
subject to \( \sum_{i=1}^{N} \pi_i^{UL}(t) \leq R^{UL} \),

- DL Scheduling:

\[
\max_{\pi_i^{DL}(t)} \sum_{i=1}^{N} \pi_i^{DL}(t) \cdot v_i^{DL}(t)
\]
subject to \( \sum_{i=1}^{N} \pi_i^{DL}(t) \leq R^{DL} \).
Simulation Results

(a) Average Age-of-Information per sub-system over increasing $N$.  
\[ \bar{\Delta} = \frac{1}{N} \frac{1}{T_{sim}} \sum_{i=1}^{N} \sum_{t=0}^{T_{sim} - 1} \Delta_i(t) \]

(b) Average Integrated Absolute Error per sub-system over increasing $N$.  
\[ \Sigma_e = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T_{sim} - 1} || e_i[k_i(t)] || \]

- stable sub-systems (control loops) are less scheduled by Vol-scheduler (→ delay) with scarce resources (increasing N)
- Vol: less improvement expected from sensor values for stable loops
• Uplink (UL) capacity increased => bottleneck shifts from UL to downlink
• VoI-scheduler can better deal with scarce resources (N=120)
• VoI buffers information that is not urgent (low VoI) (stable loops)
System model: Networked Control System
- Including a short primer on control

Selected use cases and results
- Decentralized wireless MAC & Control: Adaptive Random Access
- Scheduled wireless access & Control: Age of Information vs. Value of Information

NCS experience for everybody:
Intro to NCS benchmark platform
NCS benchmark platform
https://github.com/tum-lkn/NCSbench
Introduction & Motivation

Network Domain

Control Domain

- We combined **Network** and **Control** domains
  - towards our **benchmarking** platform
    - $\rightarrow$ **NCSbench**

- in a **practical** approach
  - $\rightarrow$ **Two-Wheeled Inverted Pendulum**
NCSbench

- ... a Benchmarking Platform that is ...

- Easy to recreate & affordable
  → Lego Mindstorm EV3

- Easy to reproduce
  → Public GitHub Repository & Wiki
  → Step-by-step instructions for usage
  → Documentation for extension

https://git.io/fpaU4
Current Status & Outcome

[1] *Benchmarking Networked Control Systems, CPSBench, 2018*


[3] *Design Of a Networked Controller For a Two-Wheeled Inverted Pendulum Robot, (under submission)*
NCSbench Platform – Implementation

- Flexible model of the CPS
  1. Computing System
  2. Communication Network
  3. Control Logic
     → allows the **performance analysis** of the individual components!

- In our implementation
  1. Lego Mindstorm & any PC
  2. Ethernet & Wi-Fi networks
  3. Delay & packet loss tolerant
NCSbench Platform – Performance

- Measures of the **delays** of the NCS
  - Network delays \(d_N\)
  - Controller \(d_{P,C}\)
  - Sensor \(d_{P,S}\)
  - Actuator \(d_{P,A}\)
  - Computing system

- Measures of the **control performance**
  - Sensor → pitch angle, robot position (fig)
  - Actuation → motor voltage (fig)
NCS Benchmark

- **Measurement scenario:**
  - Robot balancing for one minute
  - Data collection via scripts on Controller
  - Logging on Robot too expensive (only one CPU core, slow disk), data sent to Controller
  - Network: wired (Ethernet) & wireless (IEEE 802.11g, 2.4 GHz)

- **KPIs:**
  - Network:
    - Transmission Latency (in ms)
    - Jitter
  - Control:
    - Pitch angle of robot (gyro)
    - Rotation angle of motors
    - Motor voltage
    - Lost predictions

### Network KPIs

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Median + 95%</th>
<th>Q3</th>
<th>99.9%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wired</td>
<td>4.38 +- 0.041</td>
<td>5.03</td>
<td>6.66</td>
</tr>
<tr>
<td>Wireless</td>
<td>8.09 +- 0.053</td>
<td>8.54</td>
<td>10.88</td>
</tr>
</tbody>
</table>

### Control KPIs

<table>
<thead>
<tr>
<th>Scenario</th>
<th>ΣPitch</th>
<th>ΣRot.</th>
<th>ΣVolt</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wired</td>
<td>763</td>
<td>152090</td>
<td>2067</td>
<td>0</td>
</tr>
<tr>
<td>Wireless</td>
<td>938</td>
<td>217080</td>
<td>2637</td>
<td>10</td>
</tr>
</tbody>
</table>

*cumulative deviations from reference value*
NCSbench: Summary

- **Results:**
  - Several publications directly based on the TWIP and the NCSbench
  - Collaboration between different project partners
  - Reproducible NCS benchmark combining Network & Control KPIs

- **Open Source NCSbench framework (https://git.io/fpaU4)**
  - TWIP software
  - Measurement scripts
  - Plotting scripts

- **Future Work**
  - Benchmarking platform is currently limited by Robot’s controller
  - Solution: Better hardware (Raspberry Pi-based)
  - Testing with different networks (WLAN 802.11ac, Bluetooth)
  - Better sensors
  - Extend the TWIP to a non-linearized controller
Conclusion

- M2M Applications → Networked Control Systems
- NCS Model → Network Induced Error for Decoupling from Control
- Global Optimization model needs further decomposition

- Threshold-based policy for multi-channel ALOHA

- Network induced error → up-/downlink scheduling problem in a cellular network scenario

- NCSbench to experiment with your favorite
  - Control law
  - Communication network strategy
References


[NCS19] https://github.com/tum-lkn/NCSbench
Questions?