Lecture

Control and Perception in Networked and Autonomous Vehicles

Patrick Scheffe, M. Sc. | Dr.-Ing. Bassam Alrifaee
Winter Semester 2021/2022

Part 6
Software Architectures and Testing Concepts
Course contents

► Dynamic vehicle models
► Control and optimization
► Network and distribution
► Machine perception
► Software architectures and testing concepts
► Case study
CPM Lab architecture

Setup/Infrastructure

Sense
Plan
Act

Mission planning
Scenario server
Simulation of passive traffic participants

Scenario, static map, mission plan
Object data
Map
Preview map
Dynamic map
Static map

Perception
Sensor data
Sensors

Coordination
Coupling graph
Error, noise

Environment model
Environment model
Environment model

Decision-Making
States, plan
Control inputs

Routing
Behavior
Trajectory
Control

Verification
States, plan

Environment model
Scenario, static map, mission plan
Object data
Map
Error, noise

States, plan

Sensors

Control inputs

Actuators

Environment model

Control and Perception in Networked and Autonomous Vehicles

Informatik 11
Embedded Software
RWTH Aachen University
Further literature (1)


Further literature (2)


CPM Lab motivation

**Simulation:**
abstracts from real-world behavior

**Lab:**
rapid functional prototyping, cost and time efficient

**Real-world:**
expensive and time-consuming
CPM Lab motivation

► Dynamics

► Network
  ▪ Delays
  ▪ Packet drop

► Hardware / software
  ▪ Sensors
  ▪ Actuators

► Isolate effects – control uncertainties

► Goal
  ▪ Determinism and reproducibility
  ▪ In-the-loop testing for networked systems
In-the-loop testing (XIL)

Software

Plant

Controller

Model

Generated Code

Hardware

Desktop PC

HIL simulator platform (Real-time capable)

Target HW (e.g., Microcontroller)
In-the-loop testing (XIL)

Model in the loop (MIL)

Software in the loop (SIL)

Processor in the loop (PIL)

Hardware in the loop (HIL)
CPM Lab components

High-level controller (HLC)

Indoor positioning system (IPS)

Mid-level controller (MLC)
**CPM Lab room**

- **Main PC**
- **Vehicle shelf**
- **Lab camera**
- **Driving area**
  - $x_{\text{max}} = 4.5 \, \text{m}$
  - $y_{\text{max}} = 4 \, \text{m}$
- **Windows**
- **Door**

---

*Control and Perception in Networked and Autonomous Vehicles*

CPM Lab architecture: communication

Setup/Infrastructure
- Sense
- Plan
- Act

Sense
- Mission planning
- Scenario server
  - Simulation of passive traffic participants
- Map
  - Preview map
  - Dynamic map
  - Static map
- Scenario, static map, mission plan
- Object data
- Environment model
- Error, noise
  - Environment model
- Coordination
  - Coupling graph
- Decision-Making
  - States, plan
- Verification
  - States, plan
- Control inputs
  - Control
  - Trajectory
  - Behavior
  - Routing
- Actuators
- Environment model
- Sensor data
- Perception
- Sensors
Data Distribution Service
Data Distribution Service – CPM Lab

Domain 1: local

- HLC 1
- ...
- HLC n

Domain 21: network

- Vehicle 1
- ...
- Vehicle n
- IPS

Middle-ware

Domain 1: local

- HLC 1
- ...
- HLC n

Domain 21: network

- Vehicle 1
- ...
- Vehicle n
- IPS

Middle-ware
Data Distribution Service – message format

► Message datatype on topic defined with Interactive Data Language (IDL)

► Examples
- Pose ID, Header, $x(t)$, $y(t)$, $\psi(t)$
- State ID, Header, $x(t)$, $y(t)$, $\psi(t)$, $s(t)$, $v(t)$, $a(t)$, ...
- DirectControl ID, Header, motor_throttle, steering_servo

► Header
- create_stamp
- valid_after_stamp
CPM Lab architecture: indoor positioning system

Setup/Infrastructure

Sense
Plan
Act

Control and Perception in Networked and Autonomous Vehicles
Indoor positioning system

- Camera mounted on the ceiling
  - Top-down view on the field
  - Streams images with 50 Hz

- Vehicles are equipped with LEDs
  - 3 LEDs to encode pose
    - Non-equilateral triangle
  - 1 LED to encode ID
    - Frequency
Indoor positioning system

- Low exposure time leads to high contrast of LEDs to background

- IPS pipeline:
  1. Extract LED points of camera image
  2. Transform image coordinates to world coordinates
  3. Find vehicles
  4. Map found vehicles to past vehicles
  5. Extract position, orientation, and ID
Data Distribution Service – CPM Lab

Domain 1: local

- HLC 1
- HLC n

Middle-ware

Domain 21: network

- Vehicle 1
- Vehicle n
- IPS

Data Distribution Service

– CPM Lab
CPM Lab architecture: high level controller and vehicle

**Setup/Infrastructure**
- Sense
- Plan
- Act

**Sense**
- Perception
  - Mission planning
  - Scenario, static map, mission plan
  - Sensor data

**Plan**
- Coordination
  - Error, noise
  - Environment model
  - Coupling graph

**Act**
- Decision-Making
  - States, plan
  - Control inputs

**Control**
- Actuators

**Perception**
- Sensor data

**Coordination**
- Environment model

**Decision-Making**
- States, plan
- Coupling graph
Mechanical platform XRAY M18 PRO LiPo

**Sensors**
- IMU (DeboSens BNO055)
- Odometer (3 Hall-effect sensors + magnet)

**Actuators**
- Servo motor (Hitec D89MW) for steering
- Brushless DC Motor (Pololu VNH5019) for propulsion

**Computation**
- Mid-level controller (Raspberry Pi Zero W) for WLAN, clock synchronization, sensor fusion, trajectory following control and path tracking
- Low-level controller (ATmega2560) for reading sensor data and actuation
Raspberry Pi Zero W
ATmega2560
Motor Driver
Inertial Measurement Unit
Hall Effect Sensors and Magnet
Data Distribution Service – CPM Lab

Domain 1: local

Domain 21: network

Vehicle 1
Vehicle n
IPS

HLC 1
HLC n

State
Pose

Middle-ware

State
Pose
High-level controller

► Direct control
  ▪ Motor as input
  ▪ Steering servo as input

► Path tracking
  ▪ Path and speed as inputs
  ▪ Stanley controller [1]

► Trajectory following
  ▪ Trajectory as input
  ▪ Model predictive controller on Raspberry Pi Zero W using a simplified bicycle model [2]

► HLC in any programming language, only interface to DDS necessary

Data Distribution Service – CPM Lab

Domain 1: local

Domain 21: network

Vehicle 1

Vehicle n

IPS

HLC 1

Command

State

Pose

Command

State

Pose

Middle-ware

HLC n

Vehicle n

Control and Perception in Networked and Autonomous Vehicles
Data Distribution Service – CPM Lab

Domain 1: local

HLC 1

Command

State

Pose

Domain 21: network

Middle-ware

Command

State

Pose

Vehicle 1

Vehicle n

IPS
CPM Lab architecture: Scenario definition
Scenario definition: CommonRoad [1]

- **Lanelets**
  - Left and right bounds
  - Connections

- **Static obstacles**
  - Dimensions

- **Dynamic obstacles**
  - Dimensions
  - Trajectory

- **Planning problem**
  - Initial state
  - Goal state

---

Experimental concept

Operating Mode

--- Triggered ---

Development Computer
- Trajectory planning
- Middleware

Vehicle

--- Real-time ---

Raspberry Pi
- Trajectory following
- Time synchronization

ATmega
- Actuation
- Measurement reading

Hierarchy

--- HLC ---

MLC

LLC
Experimental concept

HLC

Middleware

MLC

LLC

\[ t \text{ [s]} \]
Process model – eXtreme Programming

- Evolutionary development in small increments
- Most important features first
- Test-first, automated testing
- Have working code
- Pair programming

- No explicit design, documentation, review → suitable for small projects
Git commands

1. **git clone**
   To copy a git repository from remote source

2. **git status**
   To check the status of files you’ve changed in your working directory

3. **git add**
   Adds changes to stage/index in your working directory

4. **git commit**
   Commits your changes and sets it to new commit object for your remote

5. **git push**
   Push your changes to remote

6. **git pull**
   Pull changes from the remote

You can find a more elaborate introduction to git in the moodle room.