Toward temporal constraints in self-driving vehicles

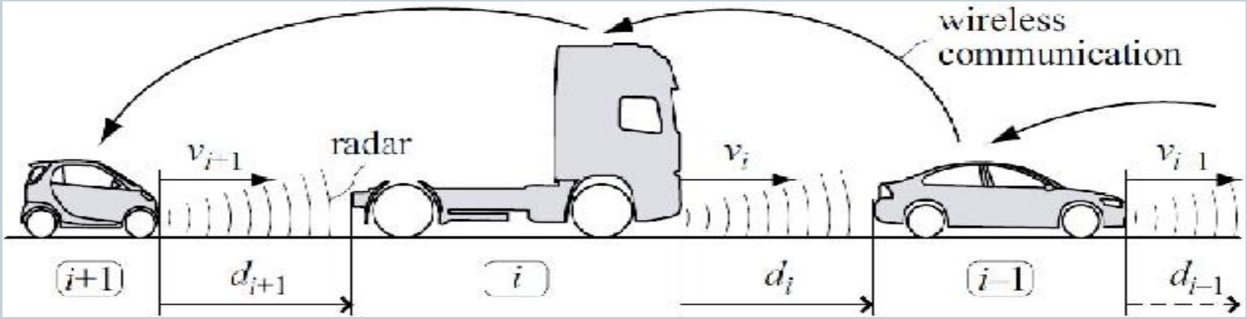
Evariste NTARYAMIRA, Cristian MAXIM, Carlos Flores PINO, Liliana CUCU

Introduction and motivation(1)

- In recent years, automated vehicles have been gaining interest for their capabilities to solve many transport issues.
- The automated vehicles are composed by numerous interconnected **embedded systems** that must cooperate between them.
- The knowledge of the system is fundamental for the correct design of the vehicle's architecture.

Introduction and motivation(2)

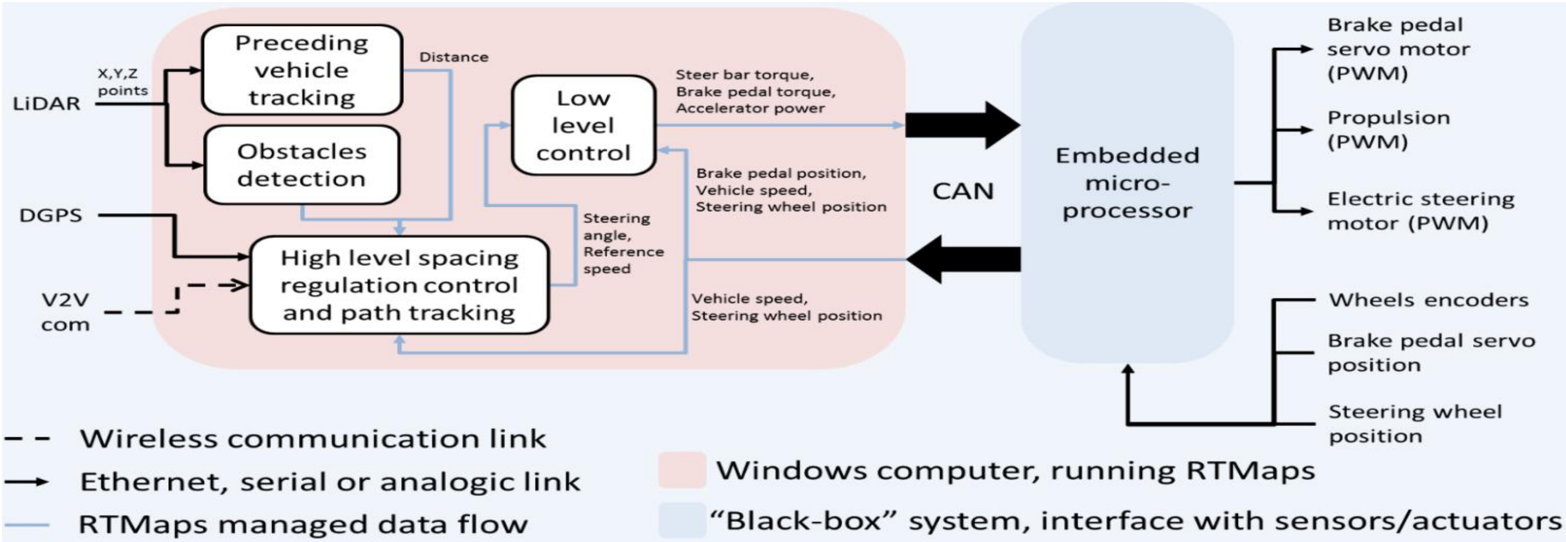
- **Platooning :** The term **"platoon"** is used to describe several vehicles operated under automatic control as a unit when they are traveling at the same speed with relatively small inter-vehicle spacing.

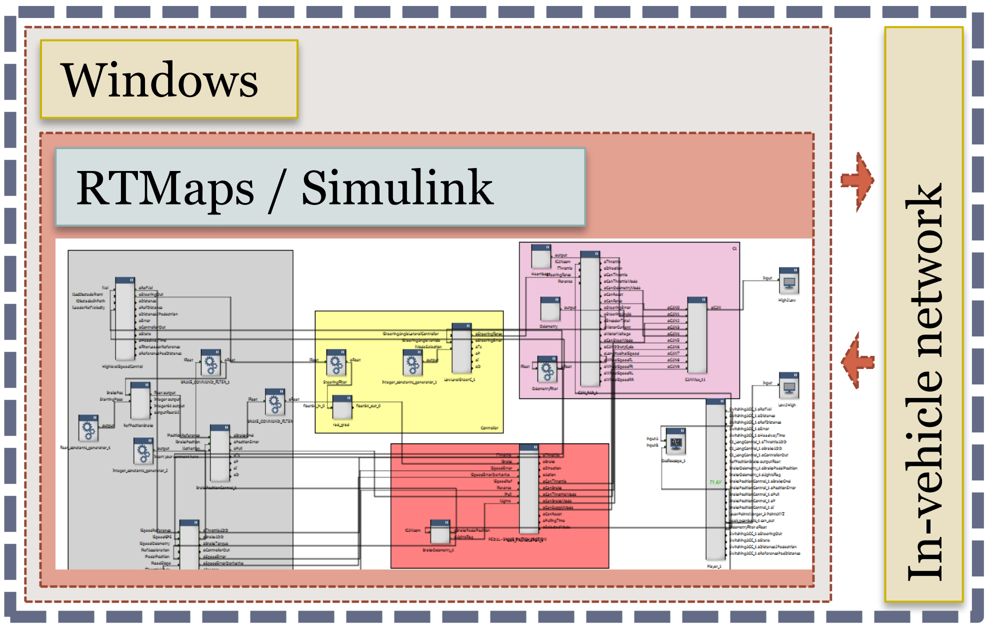


- **Organization:**
  - ➢ Each vehicle has on-board sensors, actuators automation and vehicular communication(v2v)
  - ➢ The main goal is to regulate the spacing gaps in order to increase safety and to improve traffic flow.
  - ➢ These goals are achieved thanks to a regular driving data exchange between vehicles within the platoon

For the correct and safe deployment of the platoon system, the temporal knowledge of the system is a priority

Platoon control engineering: Overview





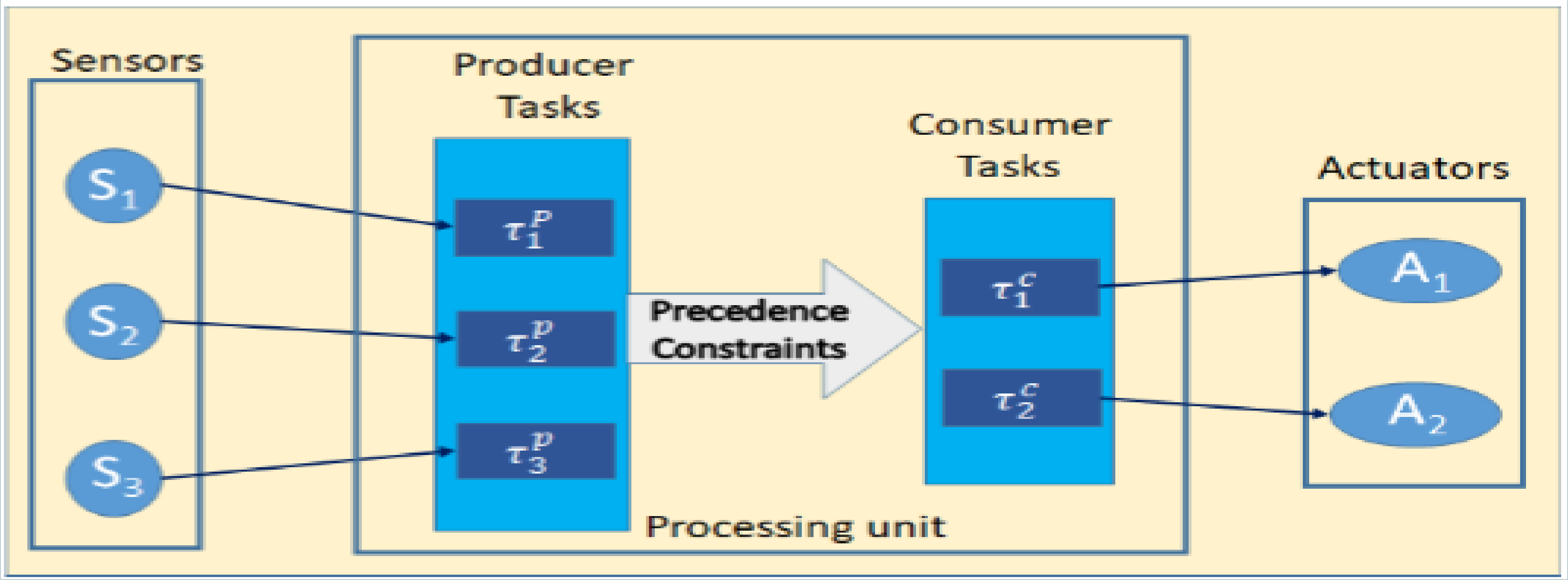
- Several components
- Heterogeneous communications tools
- High dependency on the OS

Observations & Weaknesses

- **The control model is not designed with a formal methodology in mind**
  - Verified and tested for functional requirements fulfillment by simulations(Feedback control loop, Infinity norm, PID regulator parameters adjustments, etc..)
- **Unknown component processing time**
  - Varying processing time for a component due to
    - Dependability of the state of other components it is connected to
    - Task switching to processor, disk and I/O access
    - Variation due to the dynamics of the environments
    - High dependency on the host OS(components priority assignment constraints)
- **Data relevancy constraints**
  - Since the interconnected components sample at different rates, there is possibility of data loss or same data being used several times, even when out-of-date
- **Issues related to the scheduling specification-** usage of multicore (not only) systems stays unclear

System model description

- System S of a set of n tasks. Every task τ_i is characterized by a tuple (C_i, T_i, D_i) where C_i, T_i and D_i stand respectively for worst-case execution, inter-arrival time and deadline
- Tasks are splited into two classes: producers (τ^p), and consumers (τ^c) tasks
- Producers as well as consumers gather the data at different rates due to the heterogeneity of devices and communication gateways



Scheduling challenges and open problems

For the considered model, here are scheduling-oriented challenges to overcome:

- How to provide a worst-case bound on the task response time of the system?
- How are the tasks with precedence constraints scheduled on a multicore processor such that data relevancy constraints is ensured?
- How can we estimate/compute the execution time of each task by considering:
  - **Communication delays induced by different communication channels(WIFI, CAN, LiDAR and/or RADAR) ?**
  - **Possible variation of task timing parameters (execution time for instance) due to dynamics of the environment**

Questions?