MScCCSE/ISEP – IDUCE – 2022/2023, 2nd Semester
Platoon Monitoring

Context

Autonomous Vehicles

Current vehicles capable of autonomous driving are composed of a wide variety of subsystems, that can be broadly placed in the categories of:

- sensors (components that monitor the state of the vehicle and build a representation of the world around the vehicle),
- actuators (components responsible for the dynamics of the vehicle),
- control (decision-making processes),
- communication (with other vehicles or with the infrastructure).

Platooning

Furthermore, two or more vehicles may be bundled together to create a platoon, in which the vehicle in front (leader) is followed by the vehicle(s) behind (followers), trying to safely replicate the trajectory of the leader. Vehicular platoons enable fuel savings by reducing the inter-vehicle distance, which allows to benefit of slipstream; in turn, the response time required for safe braking in the event of an emergency is reduced. As such, platooning demands reliable and responsive systems, and can be greatly augmented by a platoon-level management component (typically hosted at the leader vehicle) that receives status data from all the member vehicles.

Predictive Maintenance

The complexity of the overall system and the natural wear of physical components of the vehicles makes it natural that malfunctions will occur. Freight companies are interested in performing predictive maintenance in order to minimize downtime of their vehicles. Predictive maintenance aims at predicting and preventing such malfunctions before they happen, through a thorough monitoring (and modelling, e.g., through digital twins) of the vehicle's subsystems.

Objective

The goal of this work is to produce a system design and specification, with particular focus on the topics addressed in this semester, for the following application:

- The leader vehicle of the platoon should keep an updated view of the status of all vehicles. Upon identification of a potential malfunction in one of the vehicles, the leader should steer the platoon to the shoulder of the road or highway.

<u>Vehicle's subsystems:</u>

- Actuators: steering, braking, powertrain.

- Sensors: cameras, Lidar, ultrasonic sensors, GPS, wheel speed sensor, steering sensor, tyre pressure, engine temperature, brake status.
- Control (processes residing at ECU):
- Vehicle control (VC): short-term (0.1s to 5s) decision-making (accelerate, brake, turn) based on world representation (obtained from sensors, external information) and intended route
- Navigation (NAV): create a route from point A to B
- Platoon management (PlatMgmt): keep updated view of position, direction, speed, and status of the other platoon members
- Predictive maintenance (PredMaint): monitor status of the vehicle
- Communication (COMM): communication between platoon members

Matrix of feeds:

- Cameras → VC
- Lidar → VC
- Ultrasonic sensors → VC
- GPS → VC, NAV
- Wheel speed sensor → VC, NAV, COMM, PredMaint
- Steering sensor → VC, COMM
- Tyre pressure (4 tyres) → PredMaint
- Engine temperature (10 points) → PredMaint
- Brake status (4 tyres) → PredMaint
- Vehicle control → Steering, Braking, Powertrain
- Navigation → VC
- Platoon management → VC
- Predictive maintenance →@Leader: PlatMgmt; @ followers: COMM
- Communications → PlatMgmt → VC

Challenges

Communications

The decision-making modules (particularly the VC) need to receive data from many of the sensors in order to decide the immediate actions (0.1s to 5s) and issue commands to the steering, braking and powertrain. As the leader also needs to be aware of the state of the followers, this makes it necessary to share information not only between the vehicle's subsystems, but also between vehicles. Design the distributed information system to support this application.

Real-Time OS

The processes running at the (various) ECU(s) need to be supported by adequate runtime OSs/environments and also scheduled taking into consideration their execution frequency and the various operational frequencies (sensor sampling rates, ideal actuator command update frequency,

reception rates of information from other vehicles/infrastructure). What types of real-time OS could be considered? What data should be exchanged between tasks and how should these tasks be scheduled?

Verification

Finally, we wish to verify the correct operation of the data sharing and platooning performance. What tests can be done? What safety specifications can be written in a formally checkable way?

Note:

For all the challenges introduced, what is expected from the groups are guidelines / specifications / requirements / pseudo-code / algorithms and not concrete implementations/instances.

Deliverables

First Delivery (25%) – initial design (deadline 29/05/2022 @ 14H00):

- (FVOCA, 40,0%) System Architecture diagram, including interfaces between components and sensors/actuators,
- (SYOSY, 22,5%) Selection of communication technology to be used, specification of data models,
- (RTOSP, 22,5%) Selection of operating system technologies and features,
- (PMDEV, 15,0%) Project management (planning* and reporting of Period 1; planning of Period 2).

*) the planning for Period 1 must be submitted in Moodle on the 22nd of May, until 23h59m.

Second Delivery (25%) – revised/refined design (deadline 06/06/2022 @ 14H00):

- (FVOCA, 30%) NuSMV (simplified) model and specifications of the target system,
- (SYOSY, 30%) Temporal analysis of data transfers; analysis of communications reliability and mitigation solutions,
- (RTOSP, 30%) Identification of tasks, synchronization mechanisms, scheduling properties,
- (PMDEV, 10%) Project management (reporting of Period 2; planning of Period 3).

Final Delivery (30%) – final design (deadline 16/06/2023 @ 14H00):

- (FVOCA, 25%) NuSMV (simplified) model and specifications of the target system (improved),
- (SYOSY, 35%) Pseudocode/configurations/guidelines for setup and operation of the envisioned communications system,
- (RTOSP, 35%) Pseudocode/configurations/guidelines for the setup and operation of the envisioned real-time operating system support,
- (PMDEV, 5%) Project management report (reporting of Period 3)

Final Presentation (20%) (16/06/2022, in person)

- General presentation of the project development. The presentation shall take between 25-30 minutes.

General Notes:

- Each deliverable must be submitted in IDUCE's Moodle workspace. The format should be a PDF file (if there is external material to support the report, links to those material must be included in the pdf submitted),
- For the first two weeks, a short presentation of 10-15 minutes must also be produced and will be presented online by all member of the group,
- Feedback to the student will be provided in the laboratory classes. Also, for the first two Periods, a short report by the teachers will be provided.

Support Documentation

Links for learning about block diagram:

- "Block Diagram: Definition, Application, and Templates", https://gitmind.com/blockdiagram.html
- "Block Diagram" - ", https://www.smartdraw.com/block-diagram/

UML diagrams:

- Component diagram:
  - https://creately.com/blog/diagrams/uml-diagram-typesexamples/#ComponentDiagram
  - https://www.lucidchart.com/pages/uml-component-diagram
  - https://www.youtube.com/watch?v=VIqJ4rdKUEY
  - https://www.youtube.com/watch?v=uJez6Nd8IRI

- Deployment diagram:
  - https://creately.com/blog/diagrams/uml-diagram-typesexamples/#DeploymentDiagram
  - https://www.lucidchart.com/pages/uml-deployment-diagram
  - https://www.youtube.com/watch?v=3bfJ5AORAjQ
  - https://www.youtube.com/watch?v=sPORiupW4mw

- Package diagram:
  - https://creately.com/blog/diagrams/uml-diagram-typesexamples/#PackageDiagram
  - https://www.lucidchart.com/pages/uml-package-diagram
  - https://www.youtube.com/watch?v=MxWiMSzlHZ0

- Use Case diagram:
  - https://creately.com/blog/diagrams/uml-diagram-typesexamples/#UseCaseDiagram
  - https://www.youtube.com/watch?v=zid-MVo7M-E
  - https://www.youtube.com/watch?v=rKq82MoPbtk
  - https://www.lucidchart.com/pages/uml-use-case-diagram

NuSMV models and specifications:

- RAMDE's lecture slides
- NuSMV's user manual: http://nusmv.fbk.eu/NuSMV/userman/index-v2.html
- NuSMV's tutorial: http://nusmv.fbk.eu/NuSMV/tutorial/index.html
- NuSMV video tutorials:
  - https://www.youtube.com/watch?v=GIrOek9sGyQ&list=PLK50zIm6tHRiKFJvKu1a7q_z2tcXnBUHp&index=8&ab_channel=ModelChecking
  - https://www.youtube.com/watch?v=1vugJ20XucQ&list=PLK50zIm6tHRiKFJvKu1a7q_z2tcXnBUHp&index=10&ab_channel=ModelChecking