Minor Project – Distributed Systems

Date: 2021-03-23, Rev: 3.0

Background:

The idea is to apply the principles learnt in the module of distributed real-time systems as introduced in the theory lessons.

We intend to emulate a real time vehicle model which has the capability to steer both the axels. For this purpose, the communication protocol used is Time-Triggered CAN. By connecting several microprocessors together via Linking Interface, where every individual microprocessor behaves as a single node is responsible for execution of a task in the ensemble for the emulation of the vehicle.

The proposed vehicle should do the following tasks – propulsion and steering. A reference model will generate messages for propulsion (v_F & v_R) and steering ($\alpha 1$ & $\alpha 2$) based on a pre-defined profile. The output messages that will be generated by the motion controller are torques for all the four wheel-hub motors (T_{FL}, T_{FR}, T_{RL} & T_{RR}). The output messages from the vehicle emulator are the rotation angles of all the four wheel-hub motors (α_{FL}, α_{FR}, α_{RL} & α_{RR}) and the steering angles of the axels ($\alpha 1$ & $\alpha 2$).

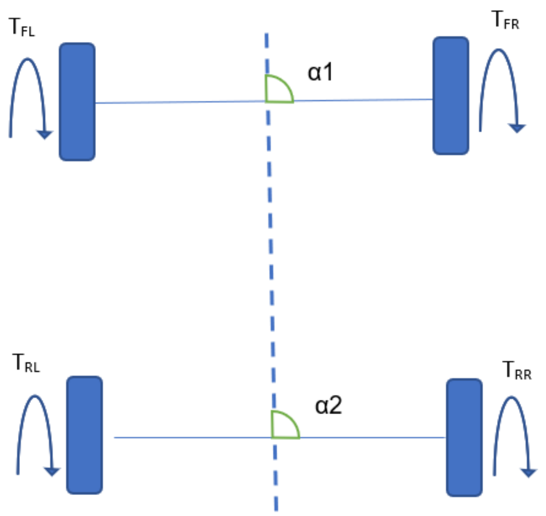

Fig 1: The vehicle model parameters

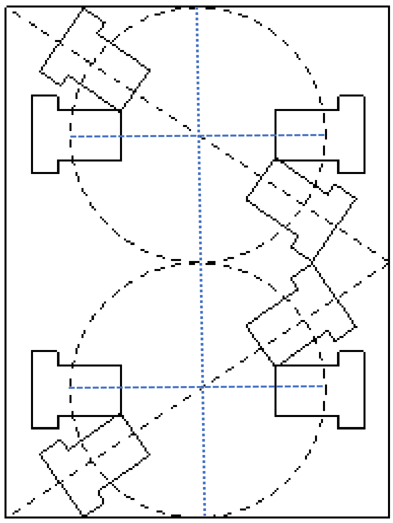

Fig 2: Vehicle turning right (Top View)

The messages generated by the reference model will be fed into the vehicle controller, which would have to combine the messages of propulsion and steering and then control them. The output of the controller should generate messages which in-turn are fed to the actuators/controlled object.

In case of communication failure or wrong messages being sent/received, the nodes in the ensemble should be able to detect it and take corrective action with the implementation of membership function and TMR (Triple Modular Redundancy) voting logic.

Goal of the Project:

To develop a redundant control system consisting of FTU's (Fault-Tolerant Unit), using several FCU (Fault-Containment Units) for the vehicle propulsion and steering tasks and implement a Fail-safe / Fail-operational mechanism in case if any failure is detected.

In this semester we will restrict the scope to vehicle propulsion with fixed steer angles in the vehicle heading direction ($\alpha 1 = 0$ & $\alpha 2 = 0$). This restriction should be explained in more detail and agreed with the supervisors of the project.

High-Level Overview of the Control Structure:

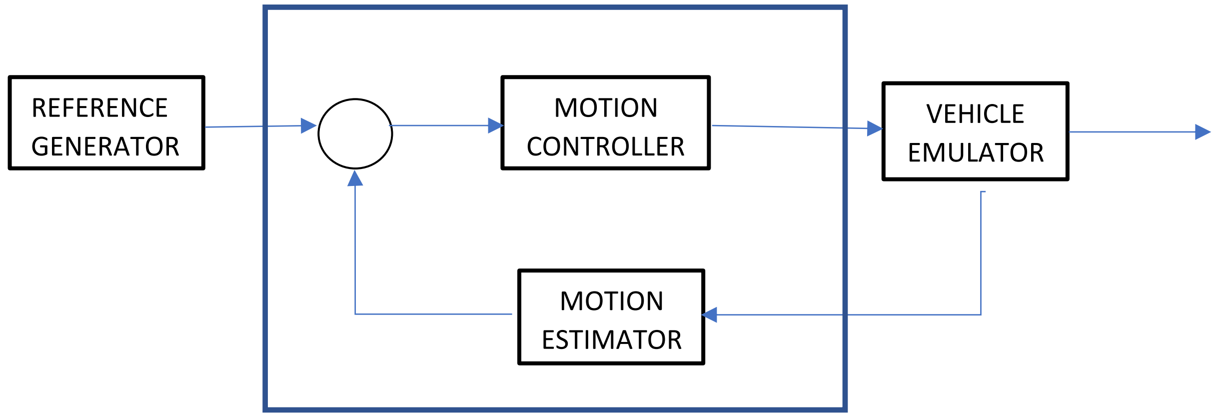

Fig 1: General Controller description

A general overview of the controller for the intended vehicle is described above in Fig 1. The scope of supply by the student groups is indicated by the rectangular block. The circle is technically part of the motion controller block, it is drawn here to have similarity with the generic feedback structure.

The reference generator is responsible for generating the reference values for the motion controller. It gives setpoint values for speed and steering values. In this project, the reference generator produces a time series of setpoints based on a predefined motion profile.

The motion controller processes the setpoint values from the reference generator and computes specific speed, torque and steering values for the front and the aft (rear) wheels on both the sides. Simplified algorithms have been developed in Simulink and provide the value-domain calculations.

The motion estimator processes the sensory outputs of the vehicle emulator. It serves as the feedback signal that is used to calculate the error and adjust the output computed by the motion controller block.

The output of the motion controller is then sent to the vehicle emulator, which in this case acts as a plant equipped with sensors and actuators and in this case emulates the behaviour of an actual vehicle.

<u>Description of the Provided platform for Simulation:</u>

The above controller will be provided as a priori for this project. The controller is implemented on a single channel which is intended to perform its tasks as designed. The controller works under some assumptions:

- The architecture is fully redundant.
- There is a fixed correlation of the value domain with the time domain.
- All the components are working and there is no fault in the system.
- The operational scenario is fixed and should not be affected by any changes in the physical conditions of the hardware. For example: sudden power loss in a component, babbling node, occurrence of faults etc.

Objective:

- Make a viable project plan to achieve the objectives.
- Decide the number of dedicated microprocessor boards responsible for execution of the tasks as shown in fig 1.
- Implement the concepts learned in the theory such as voting logic, membership functions, TMR (Triple Modular Redundancy) etc.
- Implement the TTCAN for the defined system architecture.
- Make the system architecture fault redundant.
- Develop test scenarios as a system verification plan for the developed fault redundant system and test them.

Next steps in preparing the minor project:

- Design a single channel architecture and draft the algorithms that describe the equations of motion and the controllers
- Design a simplified communication matrix
- Realize a skeleton application running on two nodes
- Define an operational scenario for a manoeuvre that included speed changes and steering changes
- Run and log the operational scenario and present the variables against time (requires a CAN logging node)
- Consider additional scenarios that inject faults