# 5th Home Assignment – Simulation and Allocation

## Autonomous Vehicle – Adaptive Cruise Control

Parallel to safety analysis, we have the task of the functional analysis of the Adaptive Cruise Control (ACC) component. As an early prototype, we are planning to use vehicles with safety drivers. Initially, these drivers can turn on the ACC if they decide that the traffic conditions are appropriate. If the ACC is turned on, then the vehicle is responsible for maintaining a controlled speed.

### High-level design

Currently we are working with the following architecture for the ACC component. We will have sensors to measure the speed of the vehicle and the distance of other vehicles on the road ahead (remember, this is a functional model!). The driver can issue commands to the ACC through an unspecified Human Machine Interface (HMI), which will be operated by the ACC Commander function. Finally, the ACC is connected to the Engine Controller to accelerate or decelerate the car. This overview of the structure can be seen in diagram *Adaptive Cruise Control / Adaptive Cruise Control Function Overview*. Figure 1 summarizes these blocks.

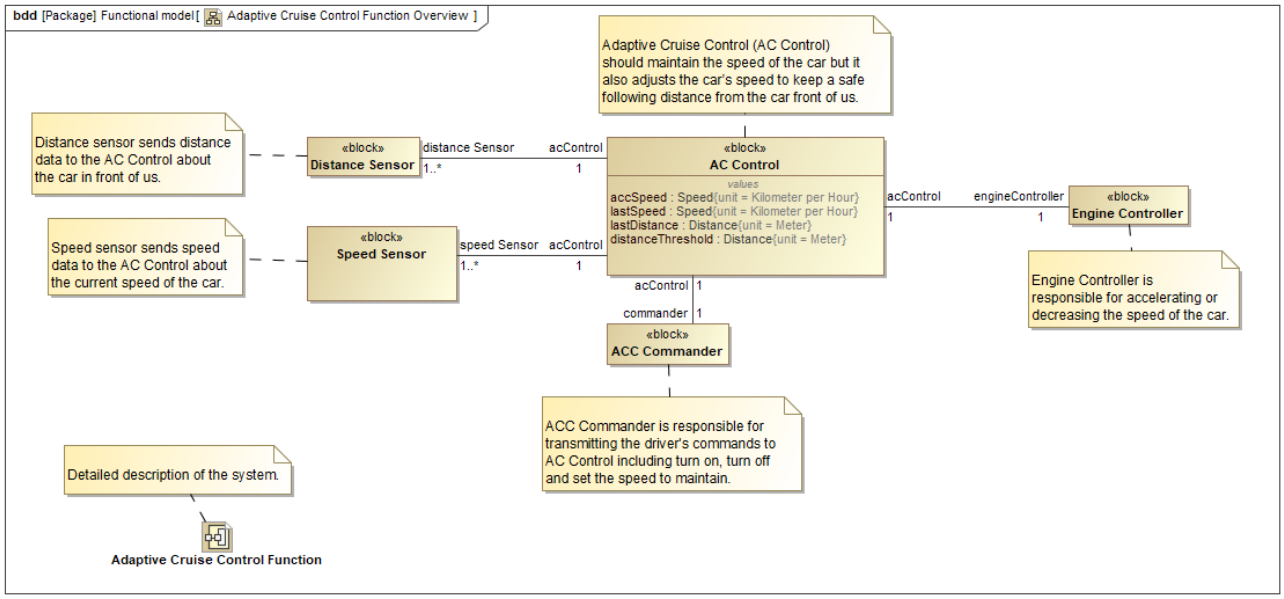


Figure 1: Functional blocks responsible for adaptive cruise control

The senior system engineers have already created a model of the AC controller's internal behaviour. The model is located at *Adaptive Cruise Control / Functional Model* [1].

### Environment model

One goal of this assignment is to perform simulation in order to determine whether the AC Controller is designed correctly. In order to simulate the environment, our engineers have created a simple mock model that can calculate the speed of our vehicle based on the instructions sent to the *Engine controller* and report the speed back to the *AC Control* component simulating its sensors. The environment model is located at *Adaptive Cruise Control / Simulation / Components / Environment*. Unfortunately our engineers do not have any expertise in SysML or simulation and their model contains many elements described in natural language.

### Tasks - Simulation

Your job is to simulate the AC Control function in its mocked environment.

a. Replace the natural language elements of the environment model with SysML elements that can be parsed and simulated automatically. To ensure the environment model can now be simulated, run a simulation on the state machine model alone, invoking events manually. The simulation should cover all states and transitions of the model. Document the input sequence, the visited states and the result.

[1] See *Adaptive Cruise Control / Overview* for all the relevant parts of the model

b. Create new diagrams and extend simulation components if necessary to simulate the AC controller together with the environment model. Use an Activity diagram to represent the driver turning the ACC on. Run simulation with the following initial conditions for 60 seconds:
   - The initial speed as well as the target speed is 50 km/h.
   - The initial speed is 45 km/h and the target speed is 50 km/h.
   - The initial speed is 55 km/h and the target speed is 50 km/h.

## Low-level architecture design

Now that we have experimented with the high-level functional specification of the ACC, it is time to allocate the functions and design a system architecture. For this, the package *Adaptive Cruise Control / Platform model* contains a platform model equivalent to the one described for the safety analysis (extended with distance sensors). According to this, we have various sensors and computing units to be used redundantly, which will require a platform-specific refinement of the high-level functional model to address the extra functional requirements introduced by redundancy (as described in the previous assignment).

Furthermore, the package *Adaptive Cruise Control / System Architecture* contains various diagrams and "boilerplate" elements prepared to help allocation and building a system architecture. The required steps along with the diagrams necessary to perform them are collected on *Adaptive Cruise Control / Overview*.

## Tasks - Allocation and system architecture

Specify the functional allocation and build a system architecture model.

c. Perform the following steps to specify a functional allocation and derive a new level of decomposition for the functional architecture.
   i. Allocate high-level components to platform elements.
   ii. Define a low-level (platform-specific) decomposition for the ACC function based on the allocation.
   iii. Assign the low-level functions to platform elements.
   iv. Display the platform functions inside the realizing platform elements to specify which function to run where.

d. *(for extra IMSc credits)* Perform the following steps to refine the new decomposition of the functional architecture and complete structural allocation.
   i. Specify how low-level functions interact with each other: add ports, introduce new logical signals, if necessary.
   ii. Display ports of the low-level functions and allocate them to ports of the enclosing platform elements.
   iii. Allocate signal and data types to platform data formats: create new subtypes of payload if necessary.
   iv. Add item flows to the platform connections to show which logical item travels where.