MBSD Lab #3 A.Y. 2023/24

Purposes

- Perform some parts of the Functional and Technical Safety Concept analysis, according to ISO26262, of a “one pedal controller” for a car.
- Implement some of the safety concepts in the Simulink model of the controller developed in Laboratory #2.
- Perform unit and integration tests on the implemented safety-related functionalities.

It is available an example of a Functional Safety Concept for the item Front Light Manager (FLM).

The deliverable, composed of

- the report (the following pages of this document)
- the Simulink models on where the safety concepts have been implemented
- all the needed files to replicate the software testing results

has to be provided as a .ZIP file up to **June 23rd at 23:59.** It shall also contain a brief report explaining the design of the controller using the following template.
It is sufficient that only one of the group members uploads it.

Important hint:
For the following analysis, consider as ASIL C all the safety goals related to unintended acceleration (those leading to an increase of the vehicle’s speed modulus) and as ASIL B the warnings to the driver and the unintended deceleration (those leading to a decrease of the vehicle’s speed modulus).

Model-Based Software Design, A.Y. 2023/24

Laboratory 3 Report

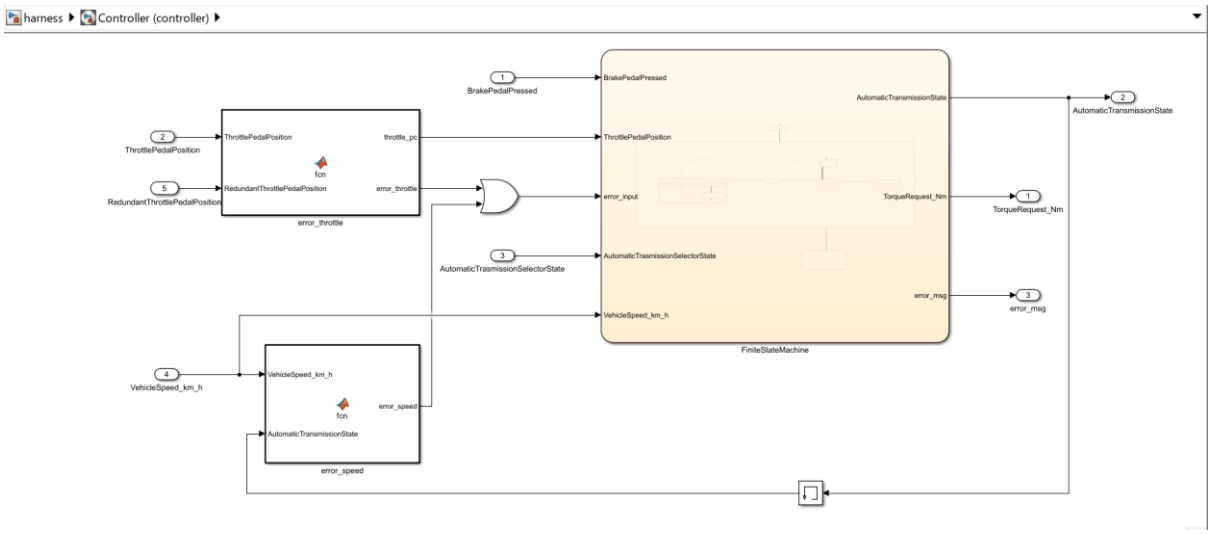
Components of the working group (max 2 people)

- Luca Pili s331500, Riccardo Solazzo s331337

Functional Safety Concept

One pedal

Functional safety architecture



Attributes of the safety goals

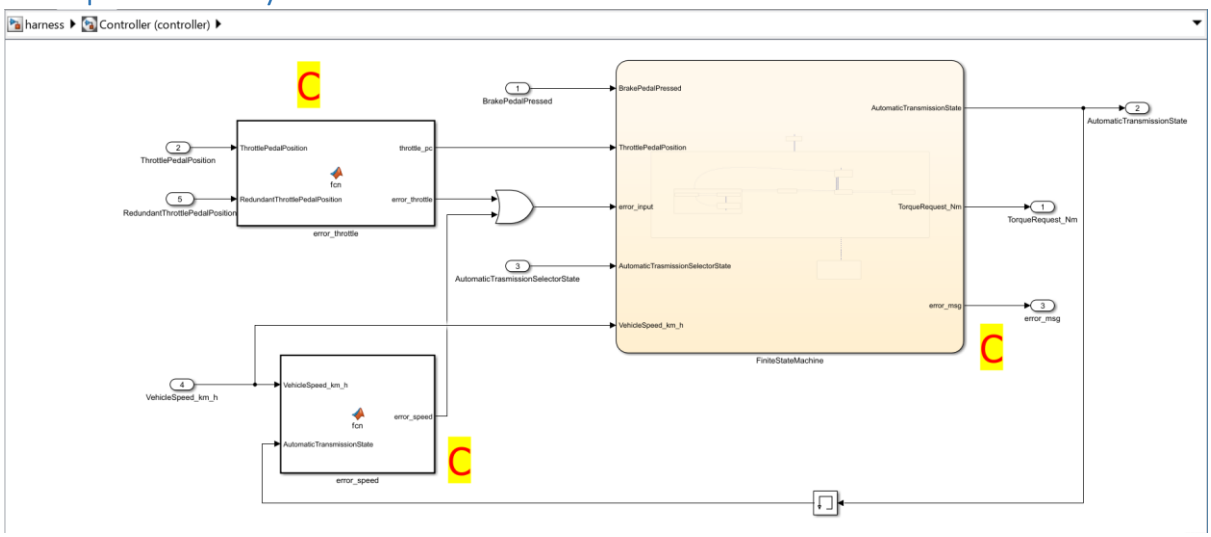
Fill in the attribute/parameters of the safety goal

Safety goal	**Attributes/Parameters of the safety goal**				
	Integrity (ASIL)	Safe state	Fault tolerance time	Warning concept	Degradation concept
SG1: avoid moving in the wrong direction or moving when parked	C	Output torque = 0	100 ms	Driver must be informed properly	In case of malfunction, torque is not provided
SG2: avoid unintended acceleration	C	Output torque = 0	100 ms	Driver must be informed properly	In case of malfunction, torque is not provided
SG3: avoid unintended braking	B	Output torque = 0	100 ms	Driver must be informed properly	In case of malfunction, torque is not provided

Functional (and technical) safety requirements and allocation

		Define functional safety requirements		Allocation of requirements on systems and elements	
		Safety requirements	**Remark**	**If applicable, allocate the safety requirements to other Items / Systems**	**If applicable, allocate the safety requirements to equipment other technologies to minimize risk. That could be e.g. hydraulic, mechanical equipment**
Safety goals	**SG1, SG2, SG3**	**SR1:** plausibility checks: velocity must be compliant with the current state	No	Warning lamp in the Cockpit-Display	No
		SR2: redundancy on the throttle pedal position measurement	No	Warning lamp in the Cockpit-Display	No

ASIL preliminary architecture[1]



[1] See document `02-iso26262.pdf`, slides 89, 90, 91, 92, 93.

Implementations[2]

Functional redundancies

A double sensor is used to implement redundancy on the measurement of the throttle pedal position.
If the discrepancy between the two measurements is more than 5% of the total range, a transition to the error state is performed: by doing so, the torque provided to the vehicle is set to zero.
The above procedure is carried out by the “error_throttle” function in the Simulink model of the controller.

Implemented plausibility checks

The following plausibility checks are considered:

- If the transmission state is set to reverse, the speed must be less than 5km/h;
- if the transmission state is set to park, the absolute value of the speed must be less than 5km/h;
- if the transmission state is set to drive, the speed must be greater than -5km/h;
- if the transmission state is set to brake, the speed must be greater than -5km/h .

If at least one of these conditions is not satisfied, a transition to the error state is performed: by doing so, the torque provided to the vehicle is set to zero.
The above procedure is carried out by the “error_speed” function in the Simulink model of the controller.

Note: the plausibility checks are chosen accordingly to the block scheme contained in the “explanations.pdf” file provided in the lab 2 kit.

[2] In the ISO26262 the implementations are based on a document called *Technical Safety Concept*, but for simplicity we move straight from the *Functional Safety Concept* to software implementations.
A guideline for the implementation phase can be found in the document `02-iso26262.pdf` from slide 81, in particular slide 86.

Software testing

Implemented unit tests

Unit “error_throttle” is tested by means of a fault injection test, considering the following scenarios:

- discrepancy > 5% of the total range;
- discrepancy = 5% of the total range;
- discrepancy < 5% of the total range.

This fault injection test can be found in the model “throttle_error_unit_test.slx”.

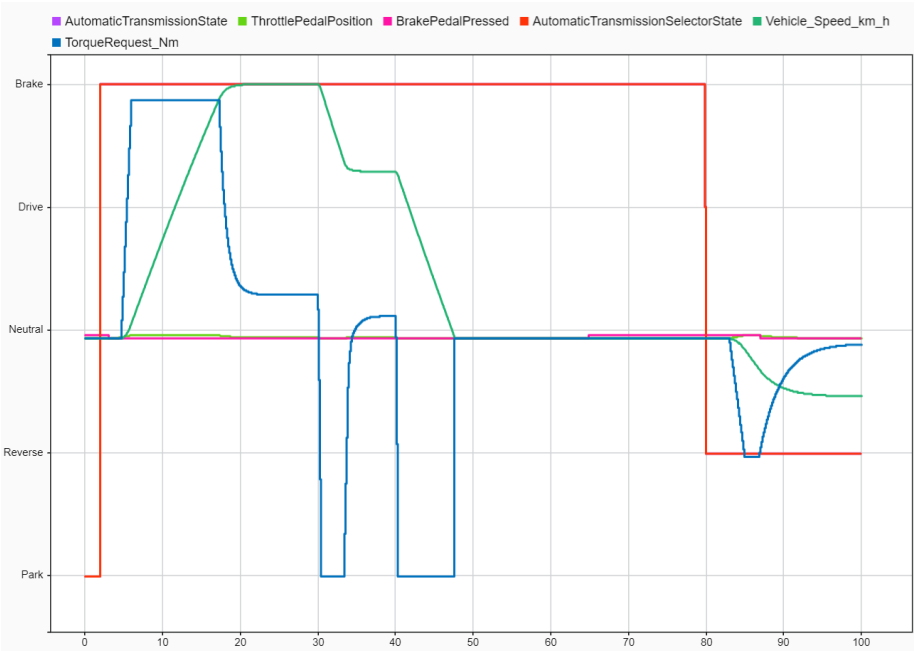
Unit “error_speed” is tested by means of a fault injection test, considering the following scenarios:

- transmission state = park, speed = -6, -5, -4, 4, 5, 6;
- transmission state = reverse, speed = 4, 5, 6;
- transmission state = drive, speed = -6, -5, -4;
- transmission state = brake, speed = -6, -5, -4.

This fault injection test can be found in the model “speed_error_unit_test.slx”.

Implemented integration tests

First of all, the faultless scenario is tested exploiting the input references provided by the “driver” block coming from the previous lab, obtaining the following results:



Then, the scenarios described in the unit test section are repeated for the integrated system, obtaining the expected results.