MBSD Lab #2 A.Y. 2023/24

# Purposes

* Implement the “one pedal controller” as a Simulink model.

To implement the system, follow the description already provided in the Laboratory 1 document considering also eventual assumptions already stated in this latter document.

The safety mechanisms are not required in this laboratory.

The Simulink project has to be split into 3 files:

* **Harness.slx**, containing reference models for the controller and plant and test stimuli generation
* **Controller.slx**, containing the controller (to be developed)
* **Plant.slx**, containing the car longitudinal physical model.

The longitudinal physical model and the test stimuli generators shall demonstrate the effectiveness of the implemented controller with respect to the expected functionalities.

Consider the plant model description as an example of the comments to be inserted in the Controller Software Unit description report.

Templates of these files are available alongside this document.

The deliverable, composed of the 3 Simulink models and a PDF file obtained by filling the following pages of this document (please delete this first page), has to be provided as a .ZIP file up to **May 24th at 23:59 through “Consegna Elaborati” of the Portale della Didattica.** 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.

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

# Laboratory 2 Report

## Components of the working group (max 2 people)

* Matteo Gravagnone, s319634
* Danilo Guglielmi, s318083

## External interfaces of the plant

|  |  |  |
| --- | --- | --- |
| **Name** | **Direction** | **Type** |
| **TorqueRequest\_Nm** | Input | CAN |
| **Vehicle\_Speed\_km\_h** | Output | CAN |
| **AutomaticTransmissionSelectorState** | Input (from the driver to the controller) | CAN {P, R, N, D, P} |
| **Selected mode/errors** | Output (to the driver) | CAN |

## Equations of the plant

The plant considered in this model is the so-called *Vehicle Longitudinal Dynamics*.

Considering:



* the vehicle acceleration, expressed in []
* the vehicle longitudinal speed, expressed in []
* the vehicle mass, expressed in []
* the longitudinal force applied to the vehicle center of gravity, expressed in []
* the longitudinal force applied to the wheel on the terrain, expressed in []
* the longitudinal force applied to the vehicle center of gravity due to the frictions with air and terrain, expressed in []
* the moment of inertia of each one of the wheels, expressed in []
* the radius of the wheel, expressed in []
* is the angular speed of the wheel, expressed in [
* is the angular speed of the engine/electrical motor, expressed in [
* is the angular speed of the engine/ electrical motor, expressed in [
* is the angular acceleration of the wheel, expressed in [
* is the frontal surface of the car, expressed in []
* is the automobile drag coefficient
* is the average density of air at sea level in standard conditions 🡪
* is the gearbox reduction ratio
* is the final drive reduction ratio
* is the total power train reduction ratio.

An extremely simplified model can be obtained as follow:

where is the vehicle acceleration, is its mass, is the longitudinal force applied to its center of gravity by the effects of the torque applied on the wheels, and is the sum of the friction forces on the vehicle due to wheel-terrain and vehicle-air interactions.

Considering that the torque is equally split between the two wheels (valid only on straight tracks)

the absence of slipping:

and considering the moment of inertia of the wheels , we can define the following equation, given that 🡪 .

The drag force that limits the maximum speed of the vehicle is equal to:

where:

and, as usually modeled:

By substituting the (2) equation in (1), and by integrating both sides, we obtain:

and, by substituting (3) in (6):

Remember that the integrator block of Simulink requires an initial condition corresponding to the vehicle's longitudinal speed at the beginning of the simulation, . A possible configuration of the integration block is shown in Figure 2.

During the model development, put all the needed gain to obtain as an output of the physical model a speed expressed in km/h.

To simulate the slope of the terrain, it is possible to add the gravity force as follows:

With the gravity acceleration on Earth.

Reasonable values for an electric compact car can be:

* The torque T (at the wheel) can vary in the range

Chart, line chart

Description automatically generated

Figure 1 Graph showing drag forces of tires (in orange) and air (in blue) at various speeds. It is possible to observe that, as imposed in equation (5), at 50 km/h. Below this speed, the tire drag is dominant, after that, the air drag is dominant. Moreover, it is possible to see the top speed of the car (around 230 km/h) when , with

With those values, the top speed on level ground reachable by the car is about 230 km/h, where the drag forces equal the traction force (3200 N).

Considering the reverse direction, the maximum speed reachable with a limitation of -60 Nm is about 45 km/h.

Graphical user interface, application, Teams

Description automatically generated

Figure 2 Settings window for the Integrator block of Simulink

Use these values (with a certain tolerance, for example, 10 %) to saturate the integrator block.

To make the model more realistic, it is possible to compute the torque request at the engine/motor. A typical ratio value for transmission of an electric car with a single gear can be around .

All the initialization parameters of the model are automatically loaded model by a callback of the function **init\_fn** as shown in Figure 3.

Graphical user interface, text, application

Description automatically generated

Figure 3 init\_fn callback configuration in the harness model properties.

## Description of the whole system

*Draw the I/O block diagram of the plant and the controller, showing how they interact.*

# Controller SW Unit specifications

*Provide a brief description of the Controller functionalities and its interfaces.*

## Interfaces

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Unit** | **Type[[1]](#footnote-2)** | **Data Type[[2]](#footnote-3)** | **Dimension** | **Min** | **Max** |
| **BrakePedalPressed** | Inherit | Input | Boolean | -1 | 0 | 1 |
| **ThrottlePedalPosition** | Inherit | Input | Single | -1 | 0 | 1 |
| **AutomaticTransmissionSelectorState** | Inherit | Input | Enum | -1 | 0 | 4 |
| **VehicleSpeed\_km\_h** | Km/h | Input | Single | -1 | -60 | 240 |
| **TorqueRequest\_Nm** | N\*m | Output | Single | -1 | -40 | 80 |
| **AutomaticTransmissionState** | inherit | Output | Enum | -1 | 0 | 4 |

*Draw the Finite State Machine (FSM) representing the controller logic*

*Comment on the design choices of the FSM, which are not trivial to be understood just by analyzing the controller logic.*

*Comment with plots of the results obtained from the test cases (it is suggested to use the Data Inspector)*

1. Input, Output, Local, Global, Volatile [↑](#footnote-ref-2)
2. Struct, Double, Integer, Enum, Boolean, etc… [↑](#footnote-ref-3)