# Model-Based Software Design, A.Y. 2022/23

# Laboratory 2 Report

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

* David Costa, 317665
* Miriana Blangero, 318925

## External interfaces of the plant

|  |  |  |
| --- | --- | --- |
| **Name** | **Direction** | **Type** |
| **Requested\_Torque\_Nm** | Input | CAN |
| **Vehicle\_Speed\_km\_h** | Output | CAN |
| **Automatic\_Transmission\_Selector** | 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

A picture containing text, line, diagram, font

Description automatically generatedA picture containing text, diagram, line, receipt

Description automatically generatedImmagine che contiene testo, schermata, diagramma, linea

Descrizione generata automaticamenteDraw the I/O block diagram of the plant and of the controller, showing how they interact to each other.

Figure 6 – Speed/Torque Controller (Tuned Pi Controller)

Figure 5 - Overview of the Controller: including the Transmission and Torque Controller (FSM) part and the Speed\Torque Controller (PI) part.

Figure 4 - Top view of the interaction between the Driver, Controller and Plant

The one pedal controller is split in two different parts:

The Transmission and Torque Controller part is a simple FSM realized with Stateflow: it outputs primarily the torque proportional to the pedal and the automatic transmission gear, it also outputs two enable signals for the brake and starting control.

The Speed/Torque Controller instead, is a simple (but perfectly tuned) PI controller that, based on a speed reference and the current speed, computes the optimal torque (taking into account the saturation limits) to reach the reference in the smallest time and with very little oscillations.

Finally, a small logic, based on switches and the Max element, selects which of the two computed torques is the right one to send to the motor depending on the situation (specified by the two enable signals).

# Controller SW Unit specifications

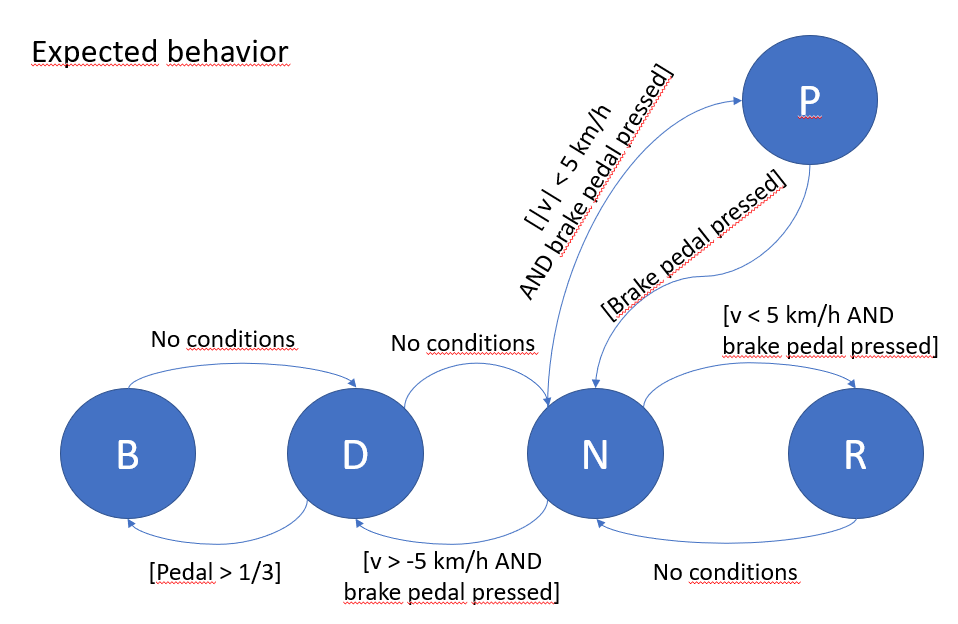
Provide a brief description of the Controller functionalities and its interfaces.

The controller for this item must deliver two different functionalities.

Firstly, it must provide the motor with the right amount of torque needed to perform the action desired by the driver and specified by the pedal position, according to the current gear selected. Notice that inside this functionality there are also additional thigs that need to be ensured:

* The negative torque applied to the vehicle in B mode must not make the car go backwards.
* When stationary in B mode the car must stay still even on positive or negative inclines.
* When the brake is pressed and released in B and D mode, the vehicle must start to move forward slowly with the same speed, regardless of the road incline.

Secondly, it must provide the right Transmission State to the automatic transmission, following the request of the driver’s selector and the conditions summarized by this diagram:



Then if the transition is accepted, it must display the Current Transmission State in the dashboard.

The input interfaces are:

* **ThrottlePedalPosition**: specifies the position of the throttle pedal and it is used to calculate the required torque.
* **BrakePedalPressed**: specifies if the brake pedal is pressed or not and it is used for the shifting conditions and for the automatic starting.
* **AutomaticTransmissionSelectorState**: specifies the AT gear selected in the shifter selector and desired by the user.
* **VehicleSpeed\_km\_h**: specifies the vehicle speed coming from the CAN network, it is used to calculate the speed error that is fed to the PI controller to enable a speed control and for the shifting conditions.

The output interfaces are:

* **TorqueRequest\_Nm**: specifies the torque fed to the motor (plant). It is calculated directly as a function of *ThrottlePedalPosition* or generated by the PI controller inside the Speed\Torque controller.
* **AutomaticTransmissionState**: specifies the gear to impose to the Automatic Transmission.

Below is a further Specification on the interfaces and symbols used in the FSM/Stateflow model and in the PI controller.

# Transmission and Torque Controller (FSM) Interfaces and constants

In parentheses the name coming in or to the harness.slx Simulink model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Unit\*** | **Type** | **Data Type** | **Dim.** | **Min** | **Max** |
| STARTING\_SPEED | Km/h | Constant | Double | 1x1 | 3 | 3 |
| brakePedalPressed  [BrakePedalPressed] | N/A | Input | Boolean | 1x1 | 0 (*false*) | 1 (*true*) |
| throttlePedalPosition  [ThrottlePedalPosition] | N/A | Input | Single | 1x1 | 0 | 1 |
| automaticTrasmissionSelectorState  [AutomaticTrasmissionSelectorState] | N/A | Input | Enum: TransmissionState | 1x1 | 0 (*Park*) | 4 (*Brake*) |
| vehicleSpeed\_km\_h  [VehicleSpeed\_km\_h] | Km/h | Input | Single | 1x1 | -60 | 240 |
| Desired\_Speed\_km\_h | Km/h | Input | Single | 1x1 | 0 | STARTING\_SPEED |
| Fault\* | N/A | Local (Input)\*\* | Boolean | 1x1 | 0 (*false*) | 1 (*true*) |
| automaticTransmissionState  [AutomaticTransmissionState] | N/A | Output | Enum: TransmissionState | 1x1 | 0 (*Park*) | 4 (*Brake*) |
| Desired\_Torque\_Nm  [TorqueRequest\_Nm] | N\*m | Output | Single | 1x1 | -40 | 80 |
| brakeControllerEnable | N/A | Output | Boolean | 1x1 | 0 (*false*) | 1 (*true*) |
| startingControllerEnable | N/A | Output | Boolean | 1x1 | 0 (*false*) | 1 (*true*) |

\* For this laboratory it is not used.

\*\* Supposed to be Input type but for now it is set to Local to not interfere with the compiler.

# Speed/Torque Controller (PI) Interfaces

In parentheses the name coming in or to the harness.slx Simulink model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Unit\*** | **Type** | **Data Type** | **Dim.** | **Min** | **Max** |
| Desired\_Speed\_km\_h | Km/h | Input | Single | 1x1 | 0 | STARTING\_SPEED |
| vehicleSpeed\_km\_h  [VehicleSpeed\_km\_h] | Km/h | Input | Single | 1x1 | -60 | 240 |
| IdealTorqueRequest\_Nm  [TorqueRequest\_Nm] | N\*m | Output | Single | 1x1 | -40 | 80 |

# Controller Logic

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

Immagine che contiene Rettangolo, diagramma, linea, Parallelo

Descrizione generata automaticamente

Figure 7 - Overview of the Transmission and Torque and FSM

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Figure 8 - Starting function (If Loop).

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

The AT controller is trivial, while the Torque controller works in three different modes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Mode** | **Signal Enable** | **Use** | **Torque Control** |
| Normal | None | Drive Reverse gears | Proportional to the throttle pedal position. |
| Brake | brakeControllerEnable | Regenerative Brake region of Brake gear | The maximum (smallest) of the inverse torques between the PI and FSM. |
| Starting | startingControllerEnable | Starting  Feature | Ideal to achieve the reference speed. |

The stop and start nodes are used for the starting feature. Accessible only when the car is stopped and the brake pedal get pressed, it enables the starting controller mode then cycles indefinitely the *starting(speedValue)* function. If the brake pedal is released, the reference for the PI controller becomes STARTING\_SPEED and the car start to move slowly, otherwise the reference is set to zero to keep the car still even on incline

Immagine che contiene testo, schermata, diagramma, Parallelo

Descrizione generata automaticamente

Figure 9 – Detailed view of the Transmission and Torque Controller (FSM)