MBSD Assignment #2 A.Y. 2024/25

# 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 4th 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. 2024/25

# Laboratory 2 Report

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

* Laura Scigliano, s331082
* Emanuele Giuseppe Siani, s330980

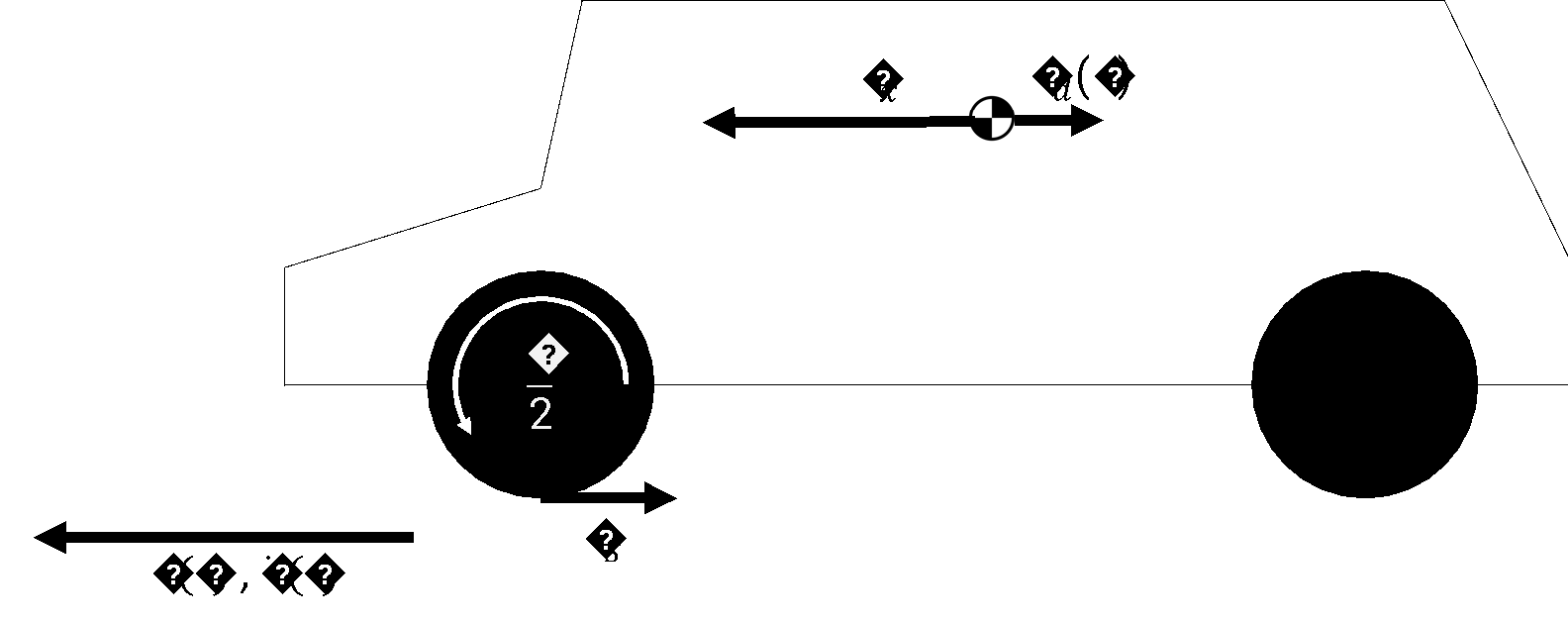
## External interfaces of the plant

| **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.

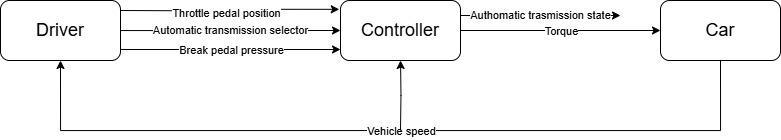
Graphical user interface, text, application

Description automatically generated

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

## Description of the whole system

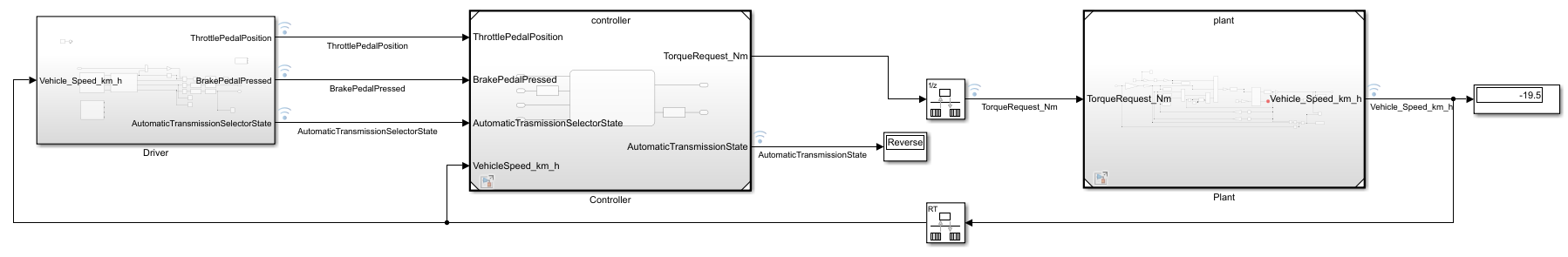
*Draw the I/O block diagram of the plant and the controller, showing how they interact.*

**

*Figure 1: Simple scheme of driving system*

The block diagram provides a simplified representation of a driving system, composed of three main components: Driver, Controller, and Car, the latter representing the system's *plant*.

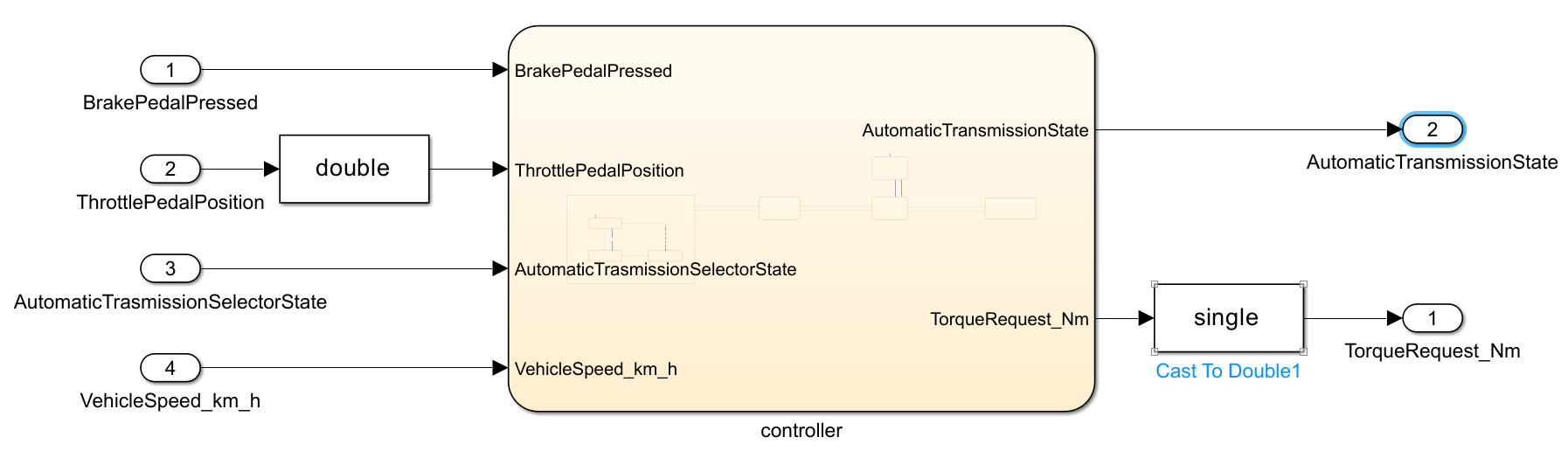
* The Driver block simulates the user's input and provides the Controller with information about the throttle pedal position, brake pedal pressure, and the gear selected via the automatic transmission.
* The Controller implements a Finite State Machine (FSM) responsible for managing gear shifting and requesting torque from the machine.
* The Car block represents the actual vehicle and simulates the dynamic behavior of the mechanical system in response to the torque requested by the Controller.

**

*Figure 2: Simulink model of driving system*

Figure 2 shows the complete Simulink model of the driving system described above. In addition to the main blocks: Driver, Controller, and Car, the model includes additional subsystems and signals that enable a more realistic and detailed simulation of the vehicle’s behavior and control logic.

The output of the *plant*, that is, the vehicle speed, is expressed in kilometers per hour (km/h) and is used as a feedback signal for both the Driver and the Controller.

**

*Figure 3: Controller with input and output*

Please note the casting blocks needed for correct computation.

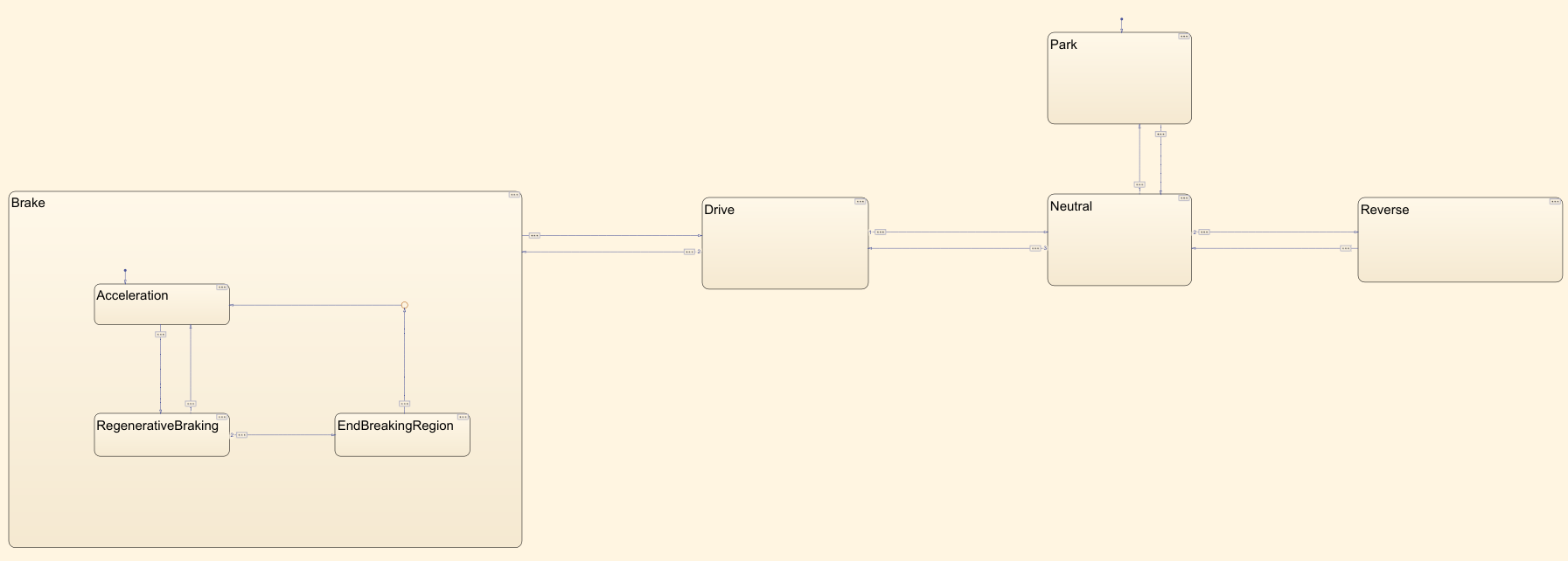
# Controller SW Unit specifications

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

## Interfaces

| **Name** | **Unit** | **Type[[1]](#footnote-0)** | **Data Type[[2]](#footnote-1)** | **Dimension** | **Min** | **Max** |
| --- | --- | --- | --- | --- | --- | --- |
| BrakePedalPressed |  | Input Data | boolean | 1 | 0 | 1 |
| ThrottlePedalPosition |  | Input Data | double[[3]](#footnote-2) | 1 | 0 | 1 |
| AutomaticTrasmissionSelectorState |  | Input Data | Enum: TransmissionState | 1 | 0 | 4 |
| VehicleSpeed\_km\_h | Km/h | Input Data | single | 1 | -60 | 240 |
| AutomaticTransmissionState |  | Output Data | Enum: TransmissionState | 1 | 0 | 4 |
| TorqueRequest\_Nm | Nm | Output Data | single[[4]](#footnote-3) | 1 | -40 | 80 |
| MAX\_TORQUE | Nm | Parameter data | double | 1 | 80 | 80 |
| MAX\_TORQUE\_REVERSE | Nm | Parameter data | double | 1 | 40 | 40 |
| coeff[[5]](#footnote-4) | N.A | Local Data | double | 1 | 0 | 1 |

# Finite State Machine (FSM)

**

*Figure 4: Scheme of controller FSM*

The finite state machine (FSM), implemented in Stateflow, models the high-level behavior of the vehicle’s transmission logic. It is composed of several key states that represent the main driving modes: Park, Reverse, Neutral, Drive, and Break.

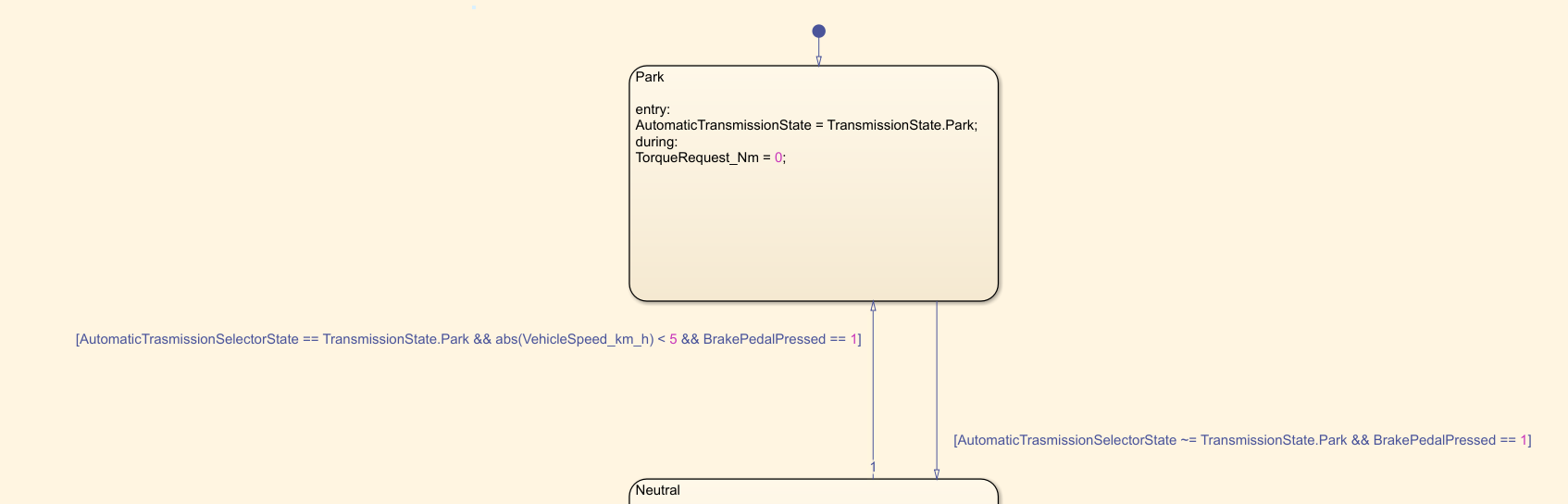
The system starts in the Park state. Transitions between states occur based on driver inputs (such as the position of the transmission selector) and other specific conditions. However, not all transitions are allowed directly, some require passing through intermediate states.

To reach other driving states, the system typically passes through Neutral, which acts as a transitional hub.

From Neutral, the FSM can move to Drive or Reverse, depending on the selected mode.

The Break state cannot be accessed directly from Neutral; it requires the system to transition through Drive first.

Park State



*Figure 5: State PARK of Controller*

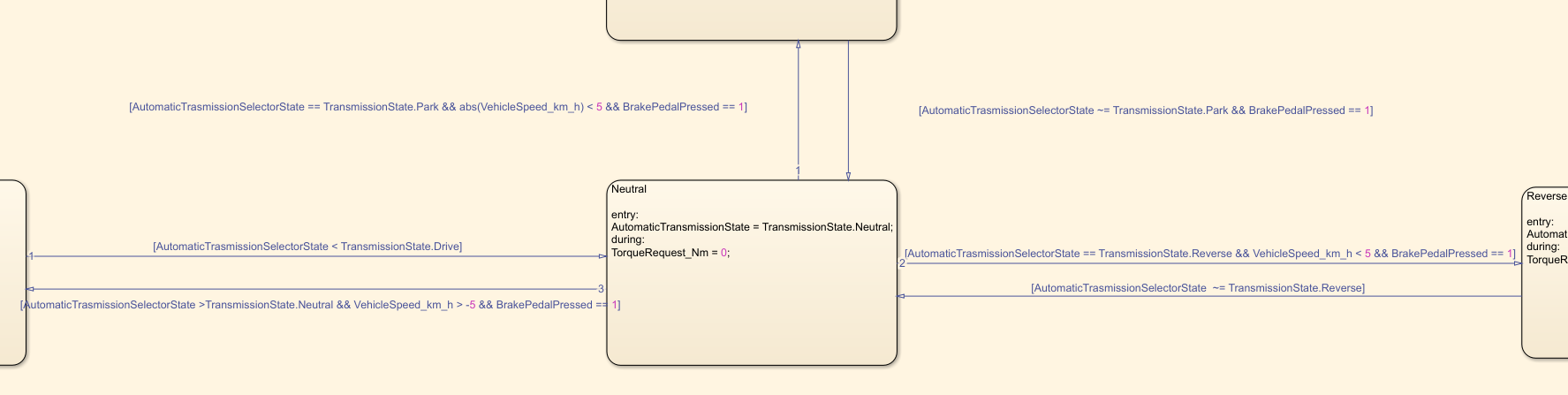
The Park state represents the initial resting state of the vehicle's transmission system. It is the default state when the system starts, ensuring that the vehicle is safely stationary.

To move from Park to another state, the system must pass through the Neutral state. This transition occurs only if the transmission selector is set to a mode different from Park, and the brake pedal is pressed. These conditions ensure that the vehicle will not inadvertently switch modes without the proper safety measures in place.

On the other hand, to return to Park from Neutral, the system requires that the transmission selector be set back to Park, the vehicle speed must be below 5 km/h, and the brake pedal must be pressed. These conditions prevent the vehicle from accidentally shifting into Park while in motion at significant speed.

When entering the Park state, the system automatically sets the transmission state to Park, and at every timestep the issued torque request is set to zero regardless the ThrottlePedalPositon value.

Neutral State



*Figure 6: State NEUTRAL of Controller*

The *Neutral* state it’s characterized by setting the TransmissionState to Neutral at the entry and to issue 0 torque at every timestep, causing the vehicle to move freewheel.   
As clearly evident from Figure6 the controller allows direct transitions from *Neutral* to *Reverse* (and vice versa) and from *Neutral* to *Drive.*

The transition to *Reverse* state is made accepted only if the AutomaticTransmissionSelectorState is set to Reverse, the VehicleSpeed\_km\_h is smaller than 5 km/h and the brake’s pedal is pressed. The limit on the vehicle’s speed ensures no sudden variation in the direction of rotation of the wheels when driving at high speed.

The Transition to the *Drive* state happens, on the other hand, when the brake pedal’s is pressed, the vehicle’s speed is greater than -5 km/h (ensuring the vehicle is not moving backwards at high speed) and the AutomaticTransmissionSelectorState is set at a value higher than Neutral. This ensures that the transition to *Drive* state happens both in case the Driver selects it or if he selects *Brake* state. This forces the controller to switch from drive first than *Brake*, then the transition to *Brake* is further regulated by the condition the throttle pedal’s position.

Reverse State

Immagine che contiene testo, linea, Carattere, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.

*Figure 7: State REVERSE of Controller*

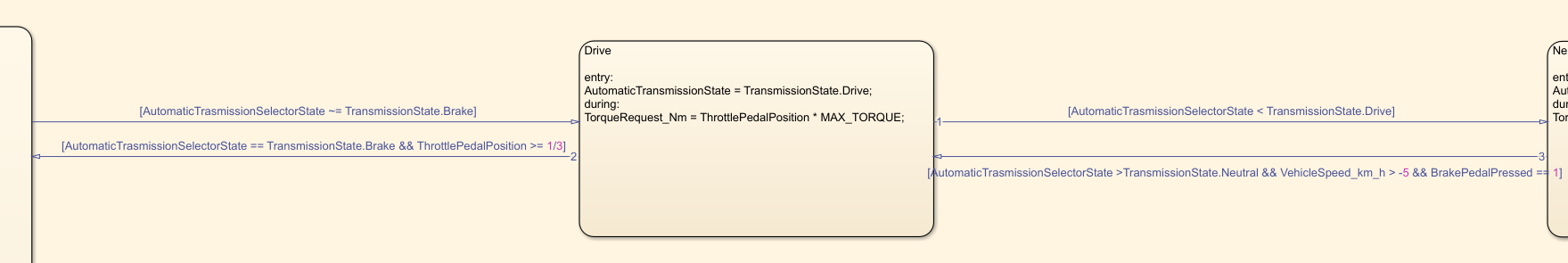
The *Reverse* state mimics the regular reverse motion in passenger cars. For this reason the TransmissionState is, of course, set to Reverse when entering the state.

When it comes to the torque request, this is a negative torque proportional to the throttle pedal’s position and multiplied by the maximum possible reverse torque.

This state can only be directly entered from the *Neutral* state, and only when the following conditions are met: the AutomatictrasmissionSelectorState is set to *Reverse*, the vehicle speed is below 5 km/h, and the brake pedal is pressed.

To transition from *Reverse* back to *Neutral*, the AutomaticTrasmissionSelectorState must be set to a value other than *Reverse*. This reflects the positioning of the reverse gear within our automatic transmission system.

Drive State



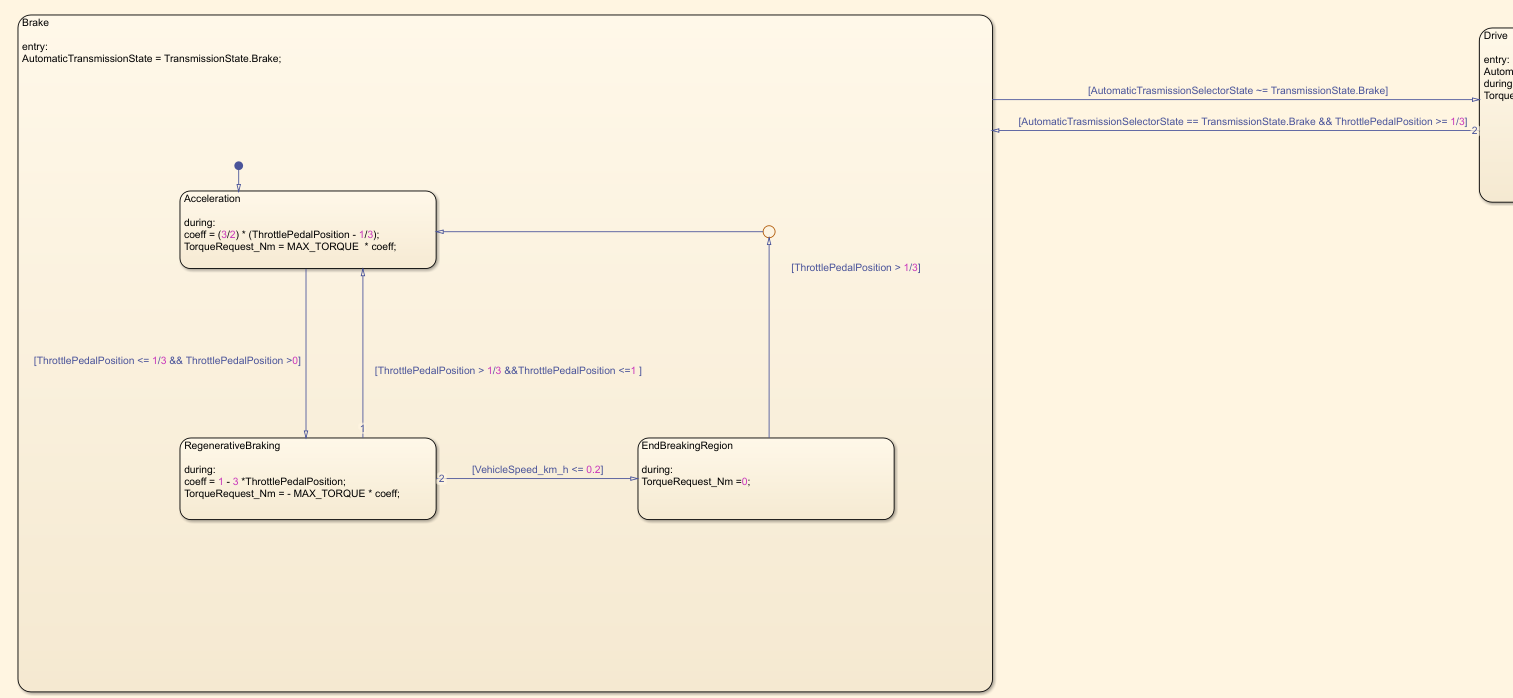
*Figure 8: State DRIVE of Controller*

The Drive mode reflects the normal driving behaviour of a car. When the throttle pedal is pressed, a positive torque is requested; conversely, when the pedal is released, no torque is requested.  
This state can be entered through the *Neutral* state, only when the AutomatictrasmissionSelectorState is set to either *Drive* or *Brake*, since the *Brake* state can only be reached via *Drive*.

Upon entering the *Drive* state, the AutomatictrasmissionSelectorState is set to *Drive*, and the torque request is calculated as the throttle pedal position multiplied by the maximum available torque.

The transition from *Drive* to *Brake* occurs when the AutomatictrasmissionSelectorState is set to *Brake*, and the throttle pedal position is greater than or equal to one-third of its maximum value.

Brake State



*Figure 9: State BREAK of Controller and its internal logic*

The *Brake* state replicates the vehicle following behavior as defined by the corresponding control equations:

Upon entering this mode, the AutomaticTrasmissionState is set to *Brake*. To implement the behavior described by the equations, an internal state machine is defined, beginning in the *Acceleration* state. This sub-state is active when the throttle pedal position is greater than ⅓. In this condition, the issued torque request follows Equation (2).

A transition to the *Brake* sub-state occurs when the throttle pedal position is smaller or equal to ⅓. In this state, the torque request is computed according to Equation (1).

*Please note that the* ***coeff*** *variable is introduced just for code readability, it does not imply any physical meaning.*

From the *Brake* sub-state, it is possible to return to *Acceleration* if the throttle pedal position rises above ⅓. Alternatively, if the vehicle speed becomes less than or equal to 0.2 km/h, the state transitions to *EndBrakingRegion*, where the torque request is set to zero.

The *EndBrakingRegion* serves as an essential intermediate state that allows the controller to exit the *Brake* state without continuing to apply negative torque. Without this state, the controller would keep requesting braking torque, potentially causing the vehicle to move backwards instead of stopping completely.

The system exits *EndBrakingRegion* and returns to the *Acceleration* state once the throttle pedal position exceeds ⅓.

# Test results

*Detailed description of the observed test results.*

*Test Pattern 2*

*A screen shot of a graph

AI-generated content may be incorrect.*

*Figure 10: Test Pattern 2: Signals evolution in time*

A graph on a white sheet

AI-generated content may be incorrect.

*Figure 11: Test Pattern 2: BrakePedalPressed and ThrottlePedalPosition evolution in time*

The figures above illustrate the time evolution of the input and output signals of the controller. This behavior results from selecting *Test Pattern 2* in the *Driver* block.  
It is important to note that the input signals of the controller are ThrottlePedalPosition, BrakePedalPressed, and AutomaticTransmissionSelectorState, shown in orange, yellow, and purple, respectively.

The objective of this graph is to demonstrate the correct operation of the controller when driven by these input signals. The controller’s response is represented by its output signals: AutomaticTransmissionState and TorqueRequest\_Nm, displayed in blue and light blue, respectively. Additionally, the correct behavior is also reflected in the Vehicle\_Speed\_km\_h signal, which, although not an output of the controller itself, is influenced by the torque request and generated by the plant model as a result.

**0–2 seconds**

At the beginning of the simulation, the default state of the controller is Park, which is consistent with the AutomaticTransmissionState being in the Park mode (though not clearly visible due to the selector signal), and both the vehicle speed and the torque request being equal to zero. The latter is determined by the execution of the *during* action in the Park state.

**2–5 seconds**

After a few seconds, the AutomaticTransmissionSelectorState changes to Brake, but the actual AutomaticTransmissionState transitions to Drive. This occurs because, as clearly observable, the ThrottlePedalPosition does not exceed the one-third threshold, thereby not fulfilling the transition condition to the Brake state. Once in the Drive state, the controller computes the torque request as the product of ThrottlePedalPosition and the constant MAX\_TORQUE. This is confirmed by the light blue and orange signals sharing the same shape, differing only in scale due to the multiplication.

**5–30 seconds**

When the ThrottlePedalPosition eventually exceeds one-third (around 5 seconds), the controller transitions the AutomaticTransmissionState to Brake. According to the controller logic, this leads to entering the Acceleration substate, which is the default substate of the Brake state. In this region, a positive torque request is continuously generated following the Equation 2’s formula defined for the Acceleration substate.

The torque request remains positive and proportional to the ThrottlePedalPosition until approximately 30 seconds, when the latter falls below the one-third threshold.

**30–33 seconds**

Since the ThrottlePedalPosition has fallen below the one-third threshold, a transition to the RegenerativeBraking substate happens, where the torque request becomes negative, as specified by Equation 1. As shown in the graph, this causes a decrease in vehicle speed proportional to the depression of the throttle pedal.

**33–41 seconds**

A noteworthy segment occurs between approximately 33 and 41 seconds: during this interval, the ThrottlePedalPosition again exceeds one-third, resulting in a small yet positive torque request. Consequently, the vehicle speed stops decreasing and stabilizes.

**41–47 seconds**

Following this, the pedal position decreases once more, leading to a negative torque request of maximum magnitude, which brings the vehicle to a stop around 47 seconds.

**47–80 seconds**

From this point until approximately 80 seconds, the controller maintains the expected behavior. As observed, the throttle pedal remains at 0, and the torque request is also 0. This is correct, as a negative torque request would undesirably cause the vehicle to move backward. In the Stateflow chart, the system enters the EndBrakingRegion state, where TorqueRequest\_Nm is held at 0 at every time step.

**80–End of simulation: Reverse state and backward motion**

At 80 seconds, the AutomaticTransmissionSelectorState is set to Reverse. Since all conditions for the transition are satisfied (selector in reverse, brake pedal pressed, and vehicle speed below 5 km/h), the controller correctly switches the AutomaticTransmissionState to Reverse. This leads to a negative torque request proportional to the ThrottlePedalPosition, persisting until the end of the simulation. As a result, the vehicle begins to move backward, as indicated by the negative value of the corresponding speed signal.

*Test Pattern 1*

A graph with lines and numbers

AI-generated content may be incorrect.

*Figure 12: Test Pattern 1: Signals evolution in time*

A graph with red lines

AI-generated content may be incorrect.

*Figure 13: Test Pattern 1: BrakePedalPressed and ThrottlePedalPosition evolution in time*

The figures above depict the time evolution of the controller’s input and output signals, resulting from the selection of *Test Pattern 1* in the *Driver* block.  
As clearly observable, this test pattern exhibits behavior that is essentially equivalent to *Test Pattern 2*, with the sole notable difference being the duration of the initial stationary phase of the vehicle, which is significantly shorter in this case, lasting approximately 2 seconds.  
The evolution of TorqueRequest\_Nm, AutomaticTransmissionState, and Vehicle\_Speed\_km\_h remains consistent with the input conditions and aligns well with the results observed in the previous test.  
For this reason, no further detailed description is required.

1. Input, Output, Local, Global, Volatile [↑](#footnote-ref-0)
2. Struct, Double, Integer, Enum, Boolean, etc… [↑](#footnote-ref-1)
3. Casted from single, see Fig. [↑](#footnote-ref-2)
4. Casted from double, see Fig. [↑](#footnote-ref-3)
5. Scalar coefficient to be multiplied by MAX\_TORQUE in the Brake state. Used just for making the equations of the torque provided in Brake state more readable. [↑](#footnote-ref-4)