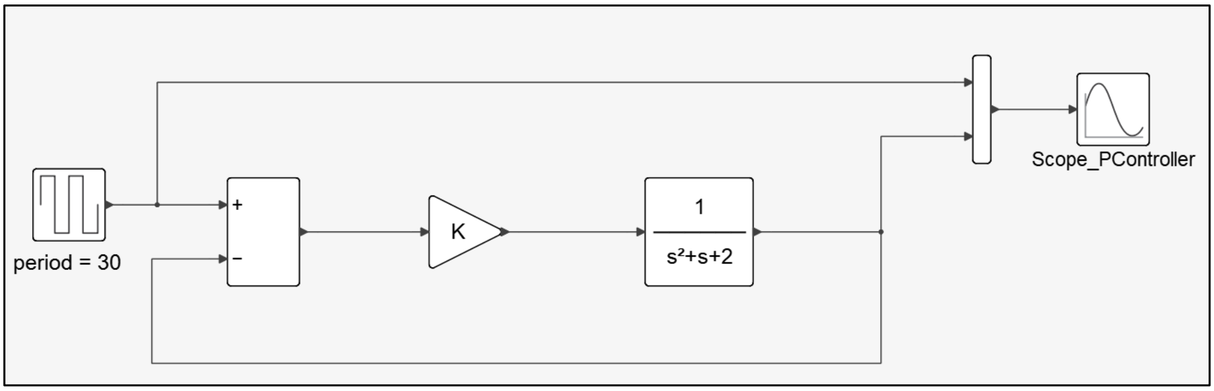
Cruise control: PID implementation

Activate TP/Project

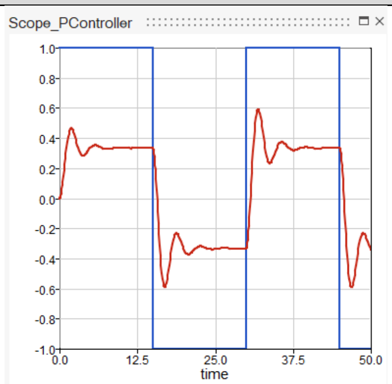
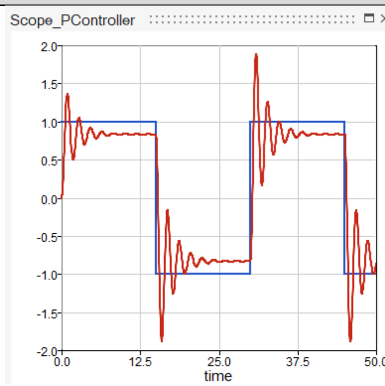
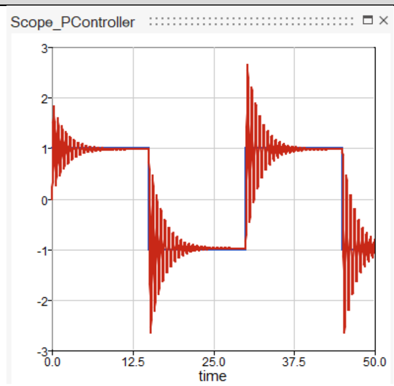
Student : MARÇAL Thomas

2.1.1. P Controller

On Activate, I made the following P Controller model:



To study this control strategy, we tested the model for different values of K and observed the response of the controlled system.

K = 1	K = 10	K = 100
		

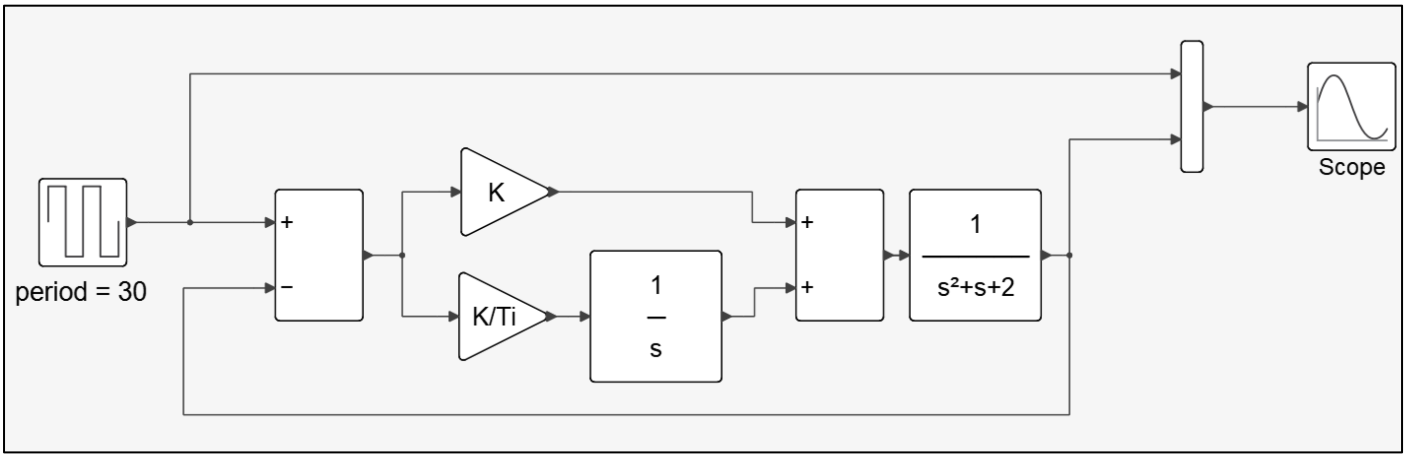
We can see that for different values of K:

- When the value of K is large in a P controller, the system's response is faster and more accurate in response to an error, leading to a larger adjustment of the output variable. However, too high of a value for K leads to instability in the system, manifested as undesirable oscillations around the desired value.
- When the value of K is small in a P controller, the system's response is slower and less precise in response to the error. This means that the controller will adjust the value of the output variable more slowly and to a lesser degree in response to the error. However, too small of a value for K leads to oscillations around the desired value, as the controller is not powerful enough to quickly correct errors.

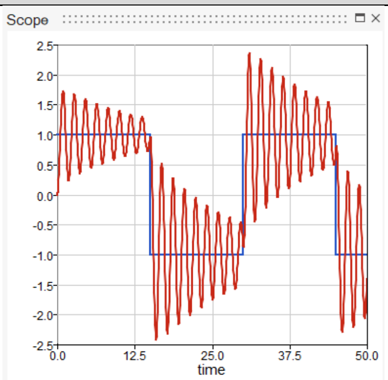
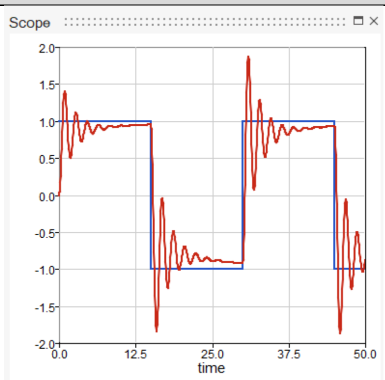
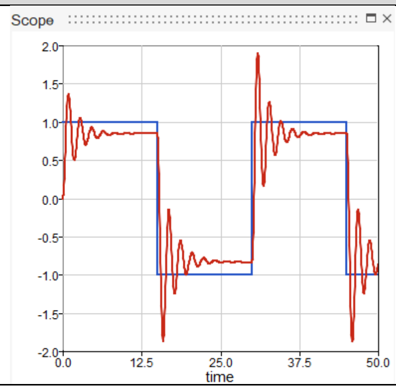
Thus, the P controller has several limitations. Firstly, we can see that it is necessary to opt for precise tuning of the parameter in order to achieve optimal performance for the model being implemented. Consequently, one must choose and find the right balance between system speed and stability. It is also difficult to limit the error because we can see that the output is not equal to the desired initial value. This means that there will always be a difference between the actual output and the desired value. Furthermore, the P controller is sensitive to disturbances, which leads to various oscillations.

2.1.2. PI Controller

On Activate, I made the following PI Controller model:



To study this control strategy, we tested the model for different values of K and Ti, and observed the response of the controlled system. For K = 10, we have :

Ti = 1	Ti = 10	Ti = 100
		

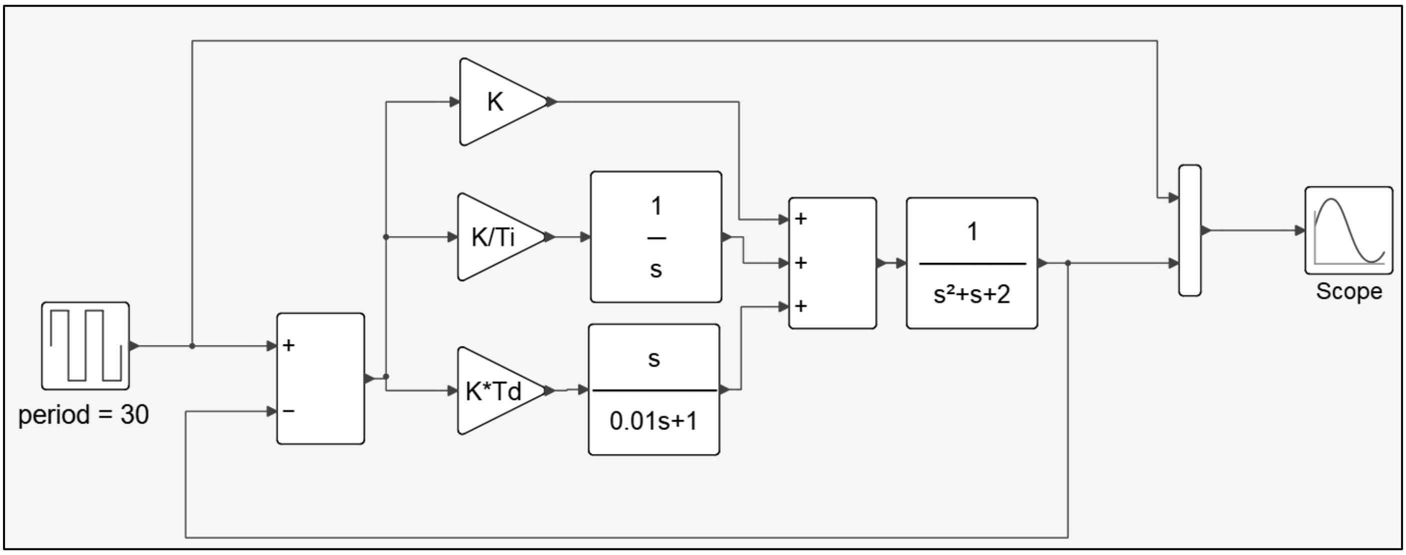
In the PI Controller, the contribution of the integral term depends on the time constant Ti used to adjust the controller.

We can see that :

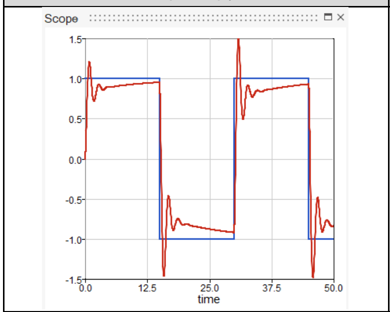
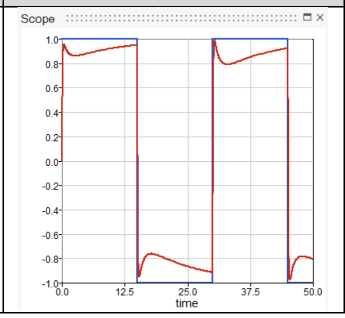
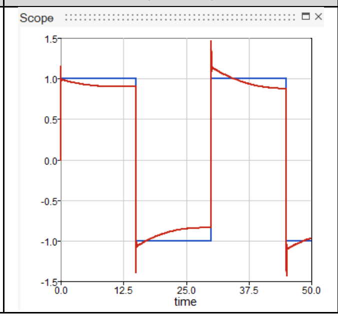
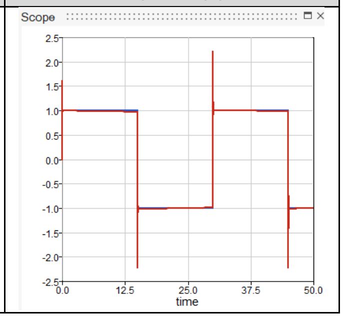
- A higher Ti value means that the integral effect occurs more slowly, which can increase the system response time in terms of time taken to reach the setpoint, but can also improve stability by reducing sensitivity to disturbances. This means that the system will be less likely to experience sudden and undesirable variations in the output control.
- On the other hand, a lower Ti value can result in a faster response, but can also make the system more unstable, with significant oscillations. Indeed, a lower Ti value means that the integral effect occurs more quickly and error accumulation occurs more quickly, which can lead to oscillations in the output control around the setpoint.

2.1.3 PID controller

On Activate, I made the following PID Controller model:



To study this control strategy, we tested the model for different values of K and observed the response of the controlled system. For K = 10 and Ti =10, we have :

Td = 0.1	Td = 1	Td = 10	Td = 100
			

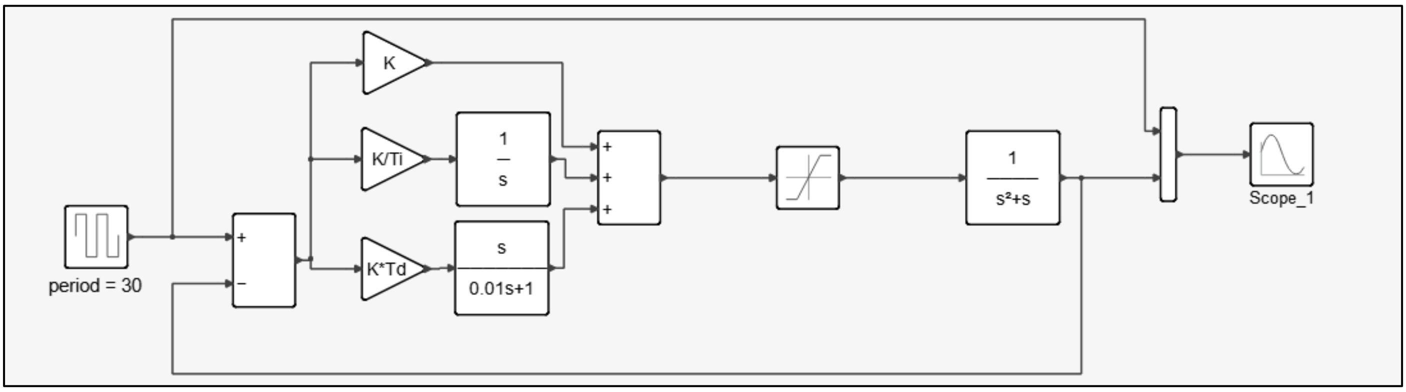
In a PID controller, the contribution of the derivative term depends on the time constant Td used to adjust the controller.

We can see that:

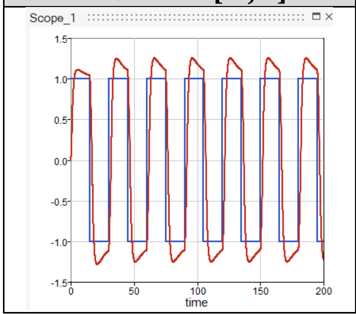
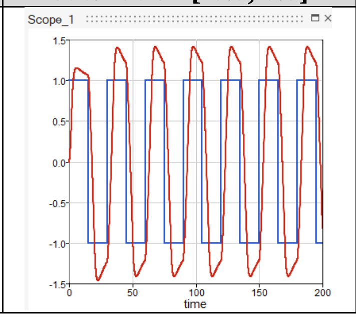
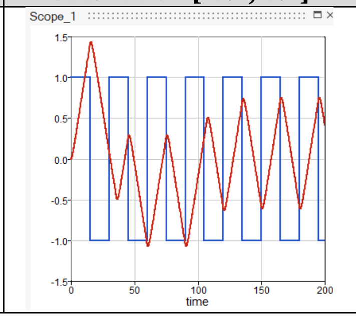
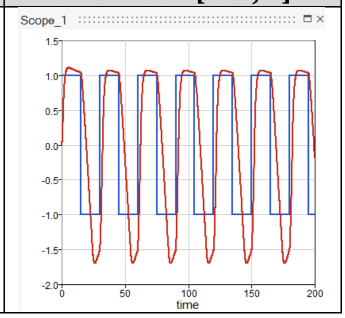
- A higher Td value means that the derivative effect occurs more slowly, which can improve stability by reducing sensitivity to disturbances and providing a smoother output. Thus, the system will be less likely to experience sudden, undesirable variations in output.
- On the other hand, a lower Td value may result in a faster response, meaning that the system will react quickly to changes. However, it can also make the system more unstable, with unwanted oscillations around the desired value.

2.1.4 PID controller with saturation

On Activate, I made the following PID Controller with saturation model:



To study this control strategy, we tested the model for different values of K and observed the response of the controlled system. For K = 10, Td = 1 and Ti = 10, we have :

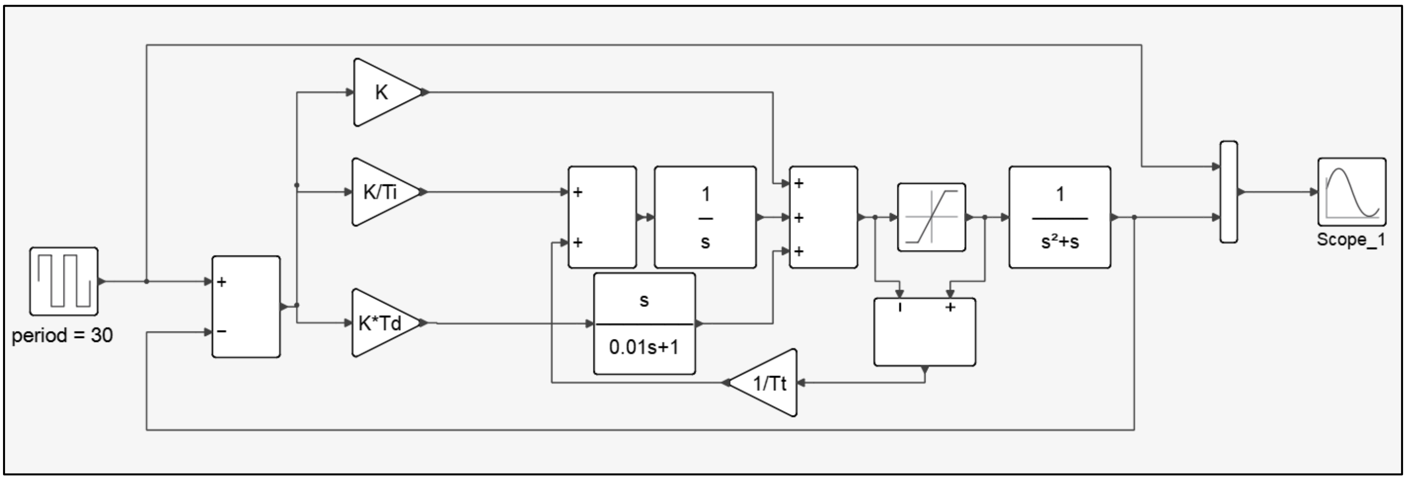
Saturation [-1, 1]	Saturation [-0.5, 0.5]	Saturation [-0.1, 0.1]	Saturation [-0.3, 1]
			

Thanks to these displays for different saturation intervals, we can see that there are several consequences to reducing the limits of the saturation block. A saturation block allows limiting the controller outputs to specified values, which also ensures stability in control loops. It can be seen that by reducing the saturation limits, the observed output range is impacted and reduced. Indeed, reducing the output range leads to a reduction in the maximum and minimum output range of the controller. This can lead to difficulties in reaching desired setpoint values as the range is then limited.

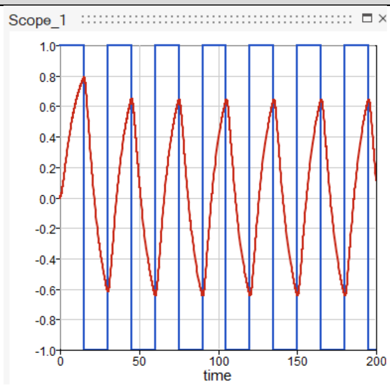
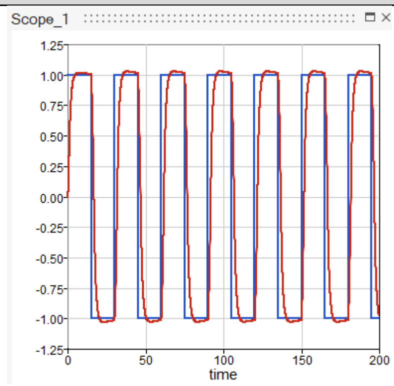
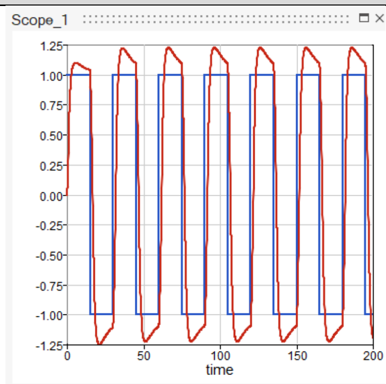
Moreover, instability may appear. If the saturation block limits are too narrow, it can lead to instability in the controller output (as seen for the saturation interval [-0.1, 0.1]).

2.1.5 Anti-Windup

On Activate, I made the following Anti-Windup model:



To study this control strategy, we tested the model for different values of K and observed the response of the controlled system. For K = 10, Td = 1, Ti = 10 and saturation between -1 and 1, we have :

Tt = 1	Tt = 10	Tt = 100
		

The best configuration of Tt for the output system to best respond to the input system is for Tt to be equal to 10.

We can see that:

- If Tt is too high, we have a slow response of the system with overshoots from the desired value.
- If Tt is too small, we have a fast response of the system with a very high instability.

The best value in this context to implement this Anti-Windup is to set Tt = 10.

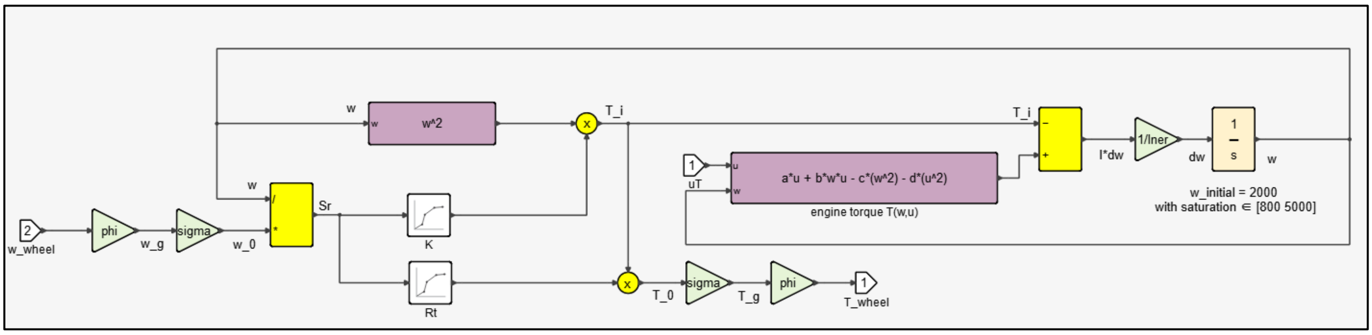
3. Project: Cruise control

Presentation of the vehicle model :

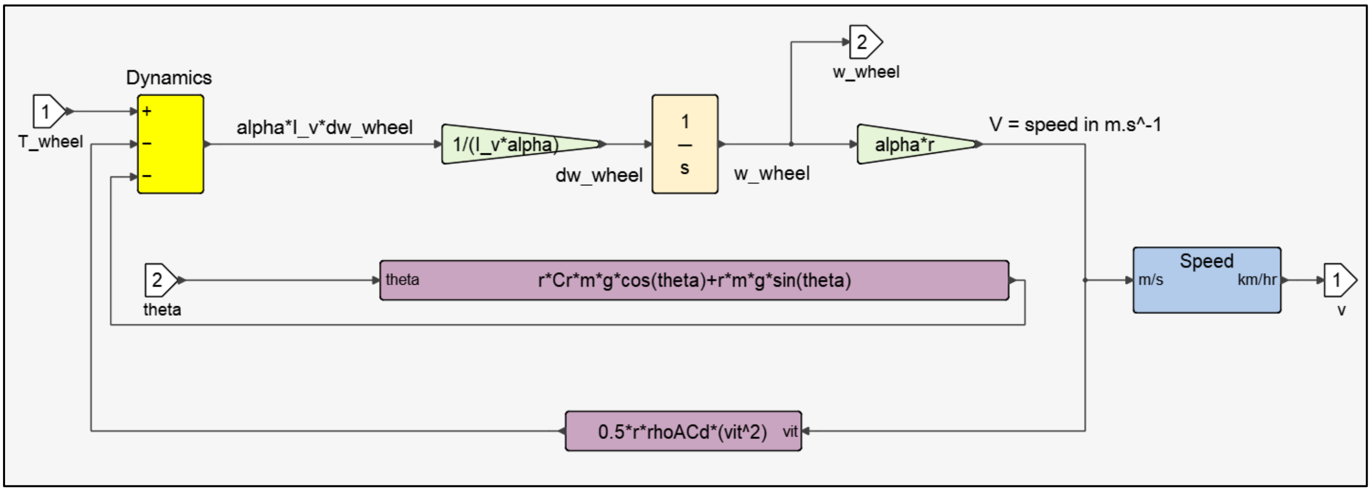
In this project, we aim to simulate the behavior of a vehicle on a road with a speed control system. To do this, we have developed a model that takes into account the dynamics of the vehicle and its engine. The vehicle model takes the throttle signal (uT) and the road gradient (θ) as inputs and provides the vehicle speed (V) in km/h as an output.

The engine model is described by several torque equations that depend on the speed and acceleration control. The model also includes the effects of the transmission on the vehicle speed, assuming a fixed gear ratio.

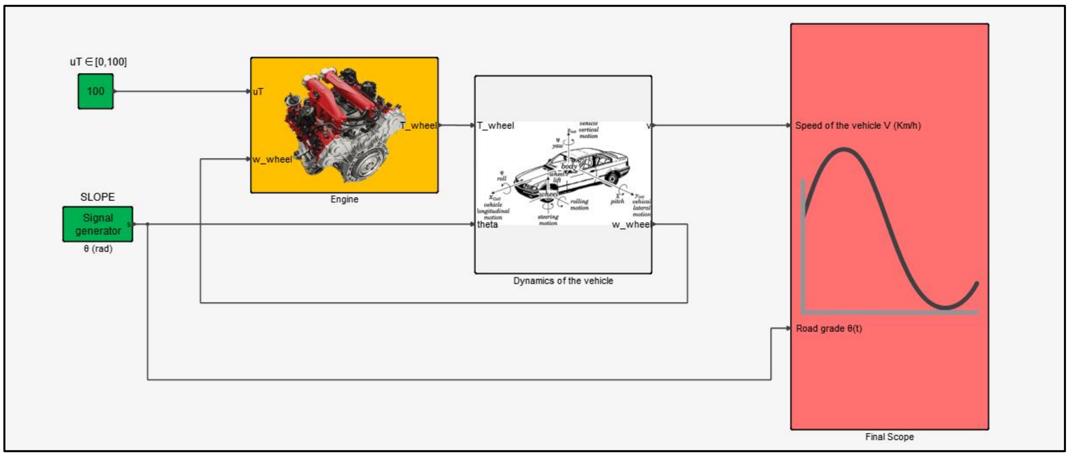
- In Activate, I implemented the engine model as a superblock, as shown above:



- For the general dynamics of the vehicle, I set up the following superblock in Activate:

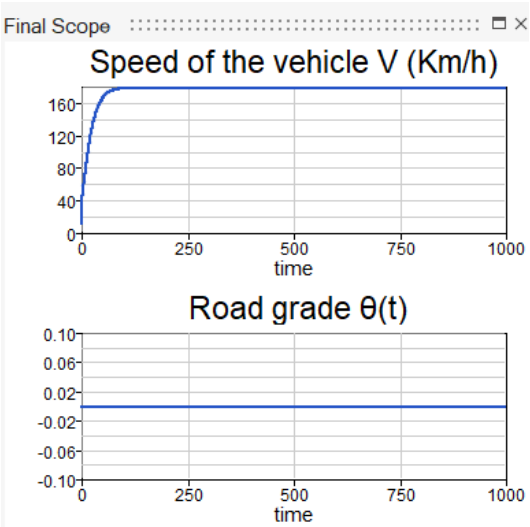


After establishing these two Super Blocks, I set up the global model of the vehicle without taking into account the action of the brakes :



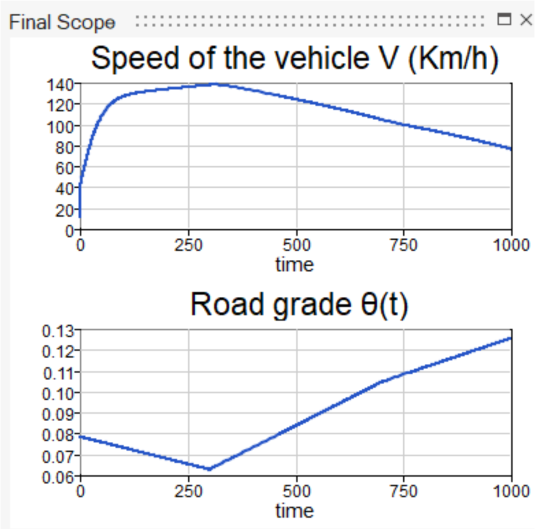
We will use our established model to perform tests to verify its accuracy and operation. In particular, we will test our model under zero slope conditions, i.e. on flat ground, while fully depressing the accelerator pedal (100% acceleration). These tests will allow us to evaluate the model's ability to predict the behavior of the system in extreme situations and to verify whether it is capable of providing consistent and reliable results. In sum, these tests will be essential to validate the relevance of our model and to ensure that it is suitable for the intended use.

We therefore obtain the following result:



The results obtained confirm the validity of our model in a classical case. Indeed, our model is able to validate the theory of the classical dynamics of a vehicle when the gas pedal is fully depressed. In this particular case, we were able to observe that on a zero slope, the car accelerates to a speed of 180 km/h, and this speed is maintained constant over time. These results confirm the relevance of our model, which is in agreement with the observed reality.

In order to validate the model, we can test our model for a road with several slopes:



We can therefore see that when going downhill, the car accelerates faster, while when going uphill, the maximum speed decreases. These results confirm the validity of our model in more specific situations, which are more faithful to reality. Indeed, our model takes into account the parameters that influence the speed of the car, such as gravity and air resistance, which allows it to reproduce the behavior observed in real conditions. This validation is important because it reinforces the relevance and applicability of our model in various contexts.

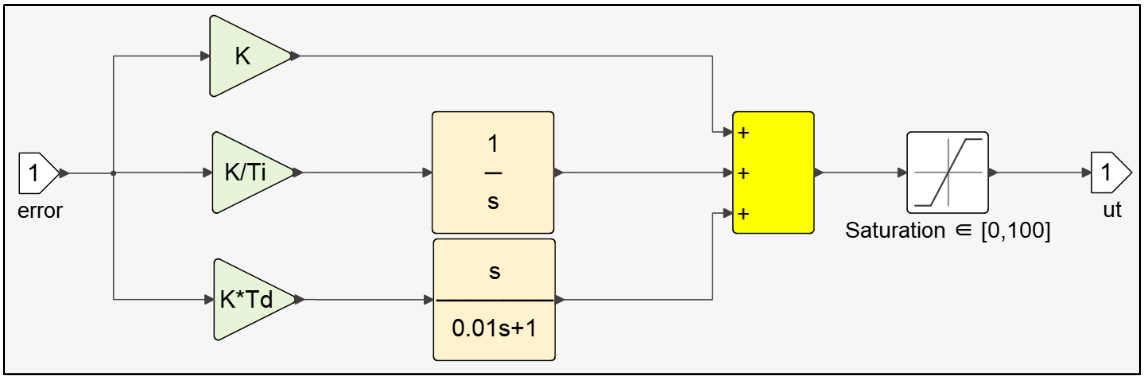
We can therefore conclude that our model is valid for representing a vehicle.

Controller presentation :

In the context of the model, the implementation of a controller is essential to regulate the behavior of the system. We have chosen to use a PID controller with saturation, which allows us to limit the action of the accelerator pedal between the interval [0, 100]. This limitation allows us to avoid oscillations and overshoots that could compromise the stability of the system, as well as to render a model faithful to reality. Indeed, adding a saturation to the controller output allows to limit the output command between the terminals 0 and 100, representing the vehicle acceleration command in percentage, because the acceleration command is limited between 0 and 100%.

Moreover, the PID controller with saturation is a means of regulating and controlling the speed of a vehicle according to the defined speed reference. In this model, the PID controller is used to control the first input of the motor dynamics, which is the acceleration command uT(t) (%).

On Activate, we have thus defined the following PID controller:



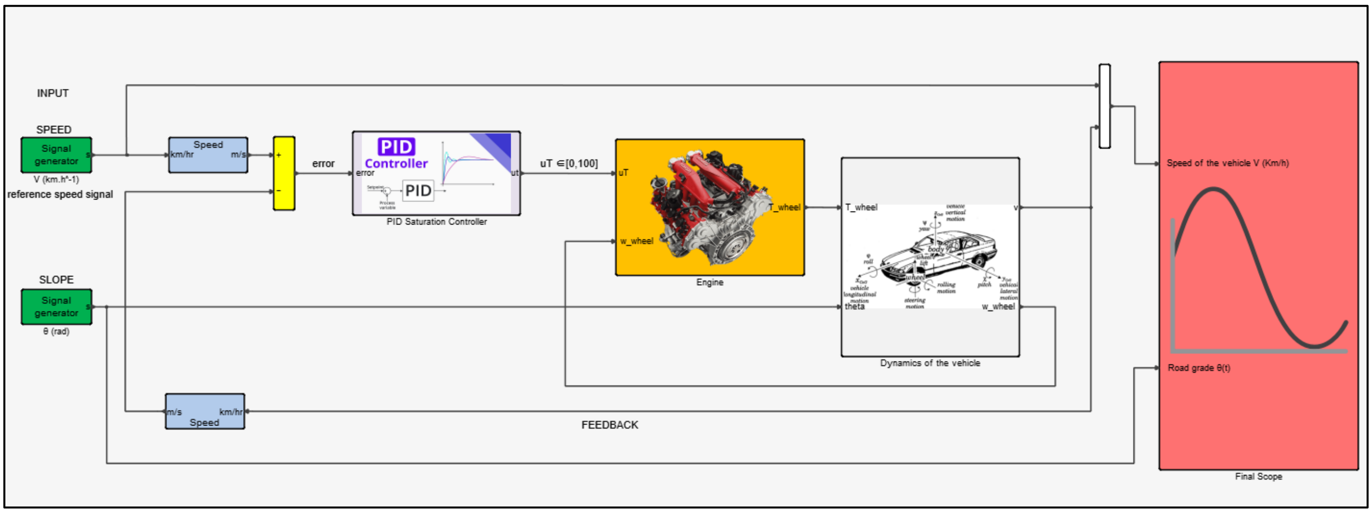
After several tests on our model, we decided to set the following PID values :

```
% PID Controller

K = 12;
Td = 1;
Ti = 20;
```

The values of K, Td and Ti of my PID controller were chosen after a series of tests and adjustments to obtain the best possible performance for our specific system. I set the value of K to 12 because it allows for a fast system response while avoiding large overshoots of the desired value at the beginning. Similarly, the value of Td was set to 1 because it allows to limit the appearance of oscillations in the system. As for Ti, I chose the value of 20 because it best matches the input model, which allows for a better match between the output model and the desired value.
In sum, these values were determined empirically based on the specific characteristics of our system and are intended to ensure a fast, stable and accurate response.

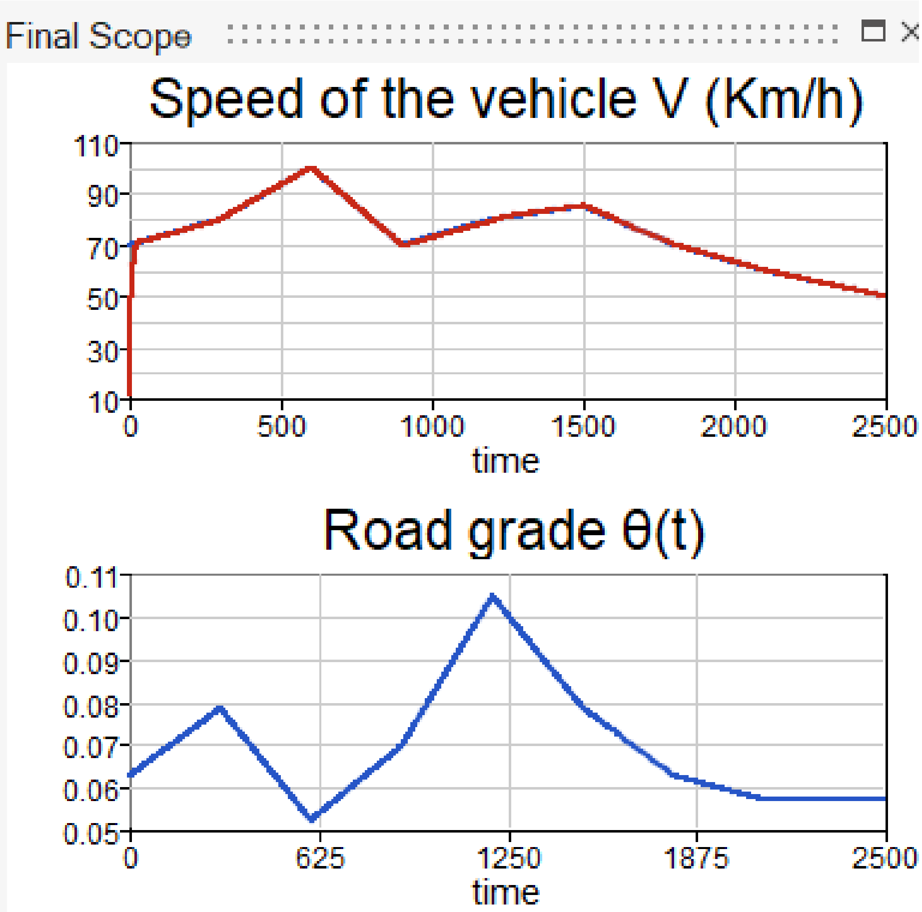
Finally, we created the final model on Activate which is the following:



So, the implementation of the PID controller with saturation allows the vehicle speed to be regulated according to the defined speed reference. The controller is connected to the output of the reference speed command to control the vehicle speed according to the road and driving conditions.

The behavior of the controlled system shows a fast and accurate response to the reference speed command. The system quickly reaches the reference speed and keeps it constant despite external disturbances. The limitation of the acceleration command between 0 and 100% prevents dangerous situations for the vehicle and passengers. In sum, the use of the PID controller with saturation is an effective method to regulate the speed of a vehicle while ensuring the safety and stability of the system.

In conclusion, we obtain the following result:



Thus, after implementing the PID controller with saturation in our model, we could observe the behavior of the system. The results obtained seem to be in accordance with our expectations.

Indeed, by examining the graph representing the reference speed in blue and the speed corrected by our controller in red, we can see that the corrected speed follows the curve of the reference speed. This means that our cruise control model is able to regulate the vehicle speed according to the defined speed reference.

Moreover, if we look closely at the graph, we can also notice that the corrected speed does not show any oscillations around the reference speed, which confirms the efficiency of our PID controller to avoid oscillations, despite the fact that we have some small overruns.

In conclusion, we can say that our cruise control model, based on a PID controller with saturation, is able to regulate effectively the vehicle speed according to the defined speed reference, while limiting oscillations and overshoot.