Alexandre Sprich - Dmytro Savchuk - Yana Krasnova

Event control
Distributed control of a vehicle queue

The objective of this TP is to find control laws with event sampling for the automatic piloting of a queue of vehicles, which move in a straight line.

I. Presentation

The system to be studied is therefore composed of cars, whose speed and acceleration are represented by dynamic equations.
The dynamics of the 1st vehicle (associated with the index 0) are as follows :

$$\begin{aligned} \dot{v}_0 &= a_0 \\ \dot{a}_0 &= -\frac{1}{\tau}a_0 + \frac{1}{\tau}u_0 \end{aligned}$$

And the dynamics of the following vehicles (associated with indexes i) are :

$$\begin{aligned} \dot{v}_i &= a_i \\ \dot{a}_i &= -\frac{1}{\tau}a_i + \frac{1}{\tau}u_i, \end{aligned}$$

With “v” the speed, “a” the acceleration, “τ” the time constant, and “u” the input command.

1. This input command u_i is called the desired acceleration because in the dynamic expression of acceleration, this input command must be equal to the acceleration a_i to stabilize the system at the desired acceleration value.

Then, the objective is to maintain a constant passage time between 2 successive vehicles. This means that depending on the speed of the vehicles, the distance between them must be adapted to keep the same time difference.

So, we define the error e, which represents the difference between the desired distance dr_i and the real distance d_i, for a vehicle i and i-1.

$$e_i \quad := \quad d_i - d_{r,i}.$$

2. Concretely, the goal is to keep "e" at 0 over time, for each pair of vehicles, which then means that the real distance measured between these 2 vehicles corresponds to the desired distance.

So each vehicle i (excluding the lead vehicle) can be modeled like this :

$$x_i \quad := \quad (v_{i-1}, a_{i-1}, u_{i-1}, e_i, v_i, a_i, u_i)^\top .$$

Which means that each vehicle i is modeled according to the speed and acceleration parameters of the vehicle i-1, and which make it possible to determine the parameters of the vehicle i.

Now, in addition to controlling the distance between vehicles, we want to avoid the accordion phenomenon (successive slowdowns and acceleration of traffic for no reason).
Thus, the control law for a vehicle i is as follows :

$$\chi_i \quad := \quad k_p e_i + k_d \dot{e}_i + u_{i-1}.$$

This command law corresponds to a PD controller, which, thanks to its proportional and derivative actions (1st and 2nd term), makes it possible to stabilize the inter-vehicle distance at the desired value. And the 3rd term allows us to prevent the accordion phenomenon from occurring.

Finally, as we want to implement a wireless communication between the system and its corrector, we have to adapt the control law like this :

$$X_i := k_p e_i + k_d \dot{e}_i + \widehat{u}_{i-1}$$

II. Analysis and synthesis of transmission laws

The next objective is to determine a state feedback control law sampled in an event-driven manner.

First, regarding the accordion phenomenon, we consider the following inequality :

$$\frac{\partial V}{\partial z_i}(A_z z_i + B_z \chi_{i-1} + E_z \hat{u}_{i-1}) \leq \mu \left(|\chi_{i-1}|^2 - |\chi_i|^2 \right),$$

which implies the absence of this accordion phenomenon.

Then, we introduce the sampling error : $e_{ui} = \hat{u}_{i-1} - u_{i-1}$, when transmitting u_{i-1} to vehicle i. In this case, the last inequality becomes :

$$\frac{\partial V}{\partial z_i}\left(\widetilde{A}_z z + B_z \chi_{i-1} + E_z e_{u_i}\right) \leq \mu \left(|\chi_{i-1}|^2 - |\chi_i|^2 \right) - \rho u_{i-1}^2 + \gamma^2 e_{u_i}^2$$

7. Therefore, if there aren't communication constraints, this means that the wireless communication takes place continuously. In this case, the error $e_{ui} = \hat{u}_{i-1} - u_{i-1}$ doesn't exist, so the last term of the expression above is null. In this way, the inequality brings us back to the first inequality.

8. Finally we can define the sampling law. Since the objective is to have the following inequality :

$$\frac{\partial V}{\partial z_i}(A_z z_i + B_z \chi_{i-1} + E_z \hat{u}_{i-1}) \leq \mu \left(|\chi_{i-1}|^2 - |\chi_i|^2 \right),$$

We can easily deduce the sampling law :

$$-\sigma\rho u_{i-1}^2 + \gamma^2 e_{ui}^2 \leq 0$$

or

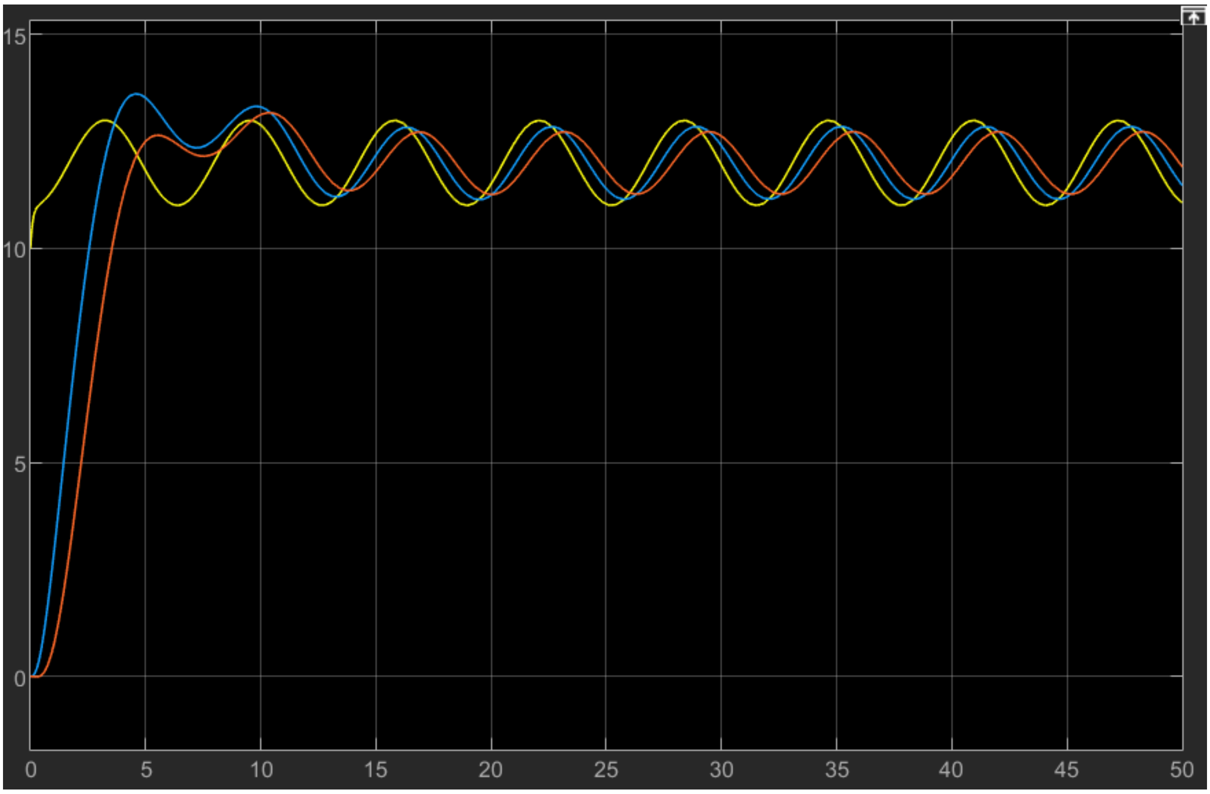
$$\gamma^2 e_{ui}^2 \leq \sigma\rho u_{i-1}^2$$

III. Simulations

Now, we want to simulate the control laws obtained using Matlab and Simulink, by considering a system with N = 3 vehicles.

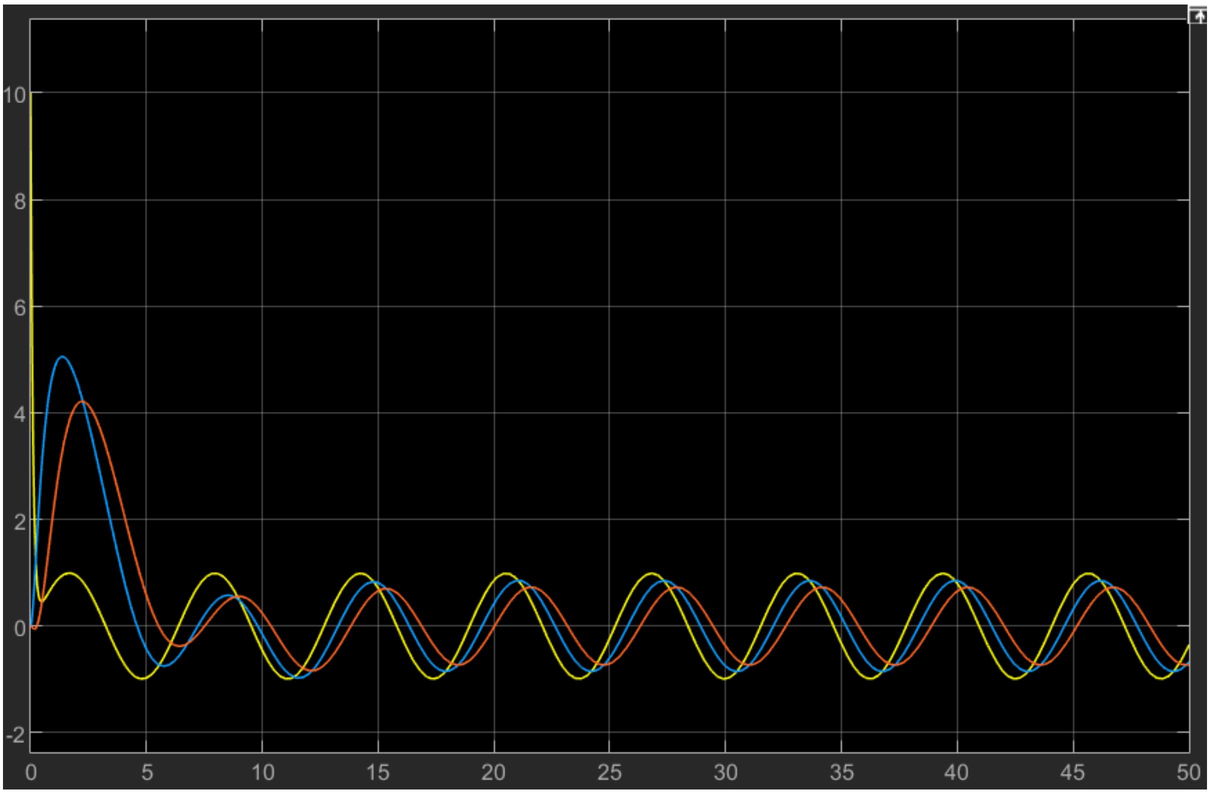
First, we want to start by simulating the operation of each vehicle, by representing the evolution of its speed and its acceleration over time.
Vehicle 0 will serve as a reference, while vehicles 1 and 2 will be the vehicles to which we will apply the control laws, based on information from the previous vehicle.

By applying a "classic" control law and considering a sinusoidal signal for the control input u0, the speed curves of the 3 vehicles are as follows :



There is therefore a stabilization around a certain speed value. Vehicle 0 (in yellow), vehicle 1 (in blue) and vehicle 2 (in red) follow each other, keeping a constant delay over time.

And regarding the acceleration curves of the 3 vehicles :



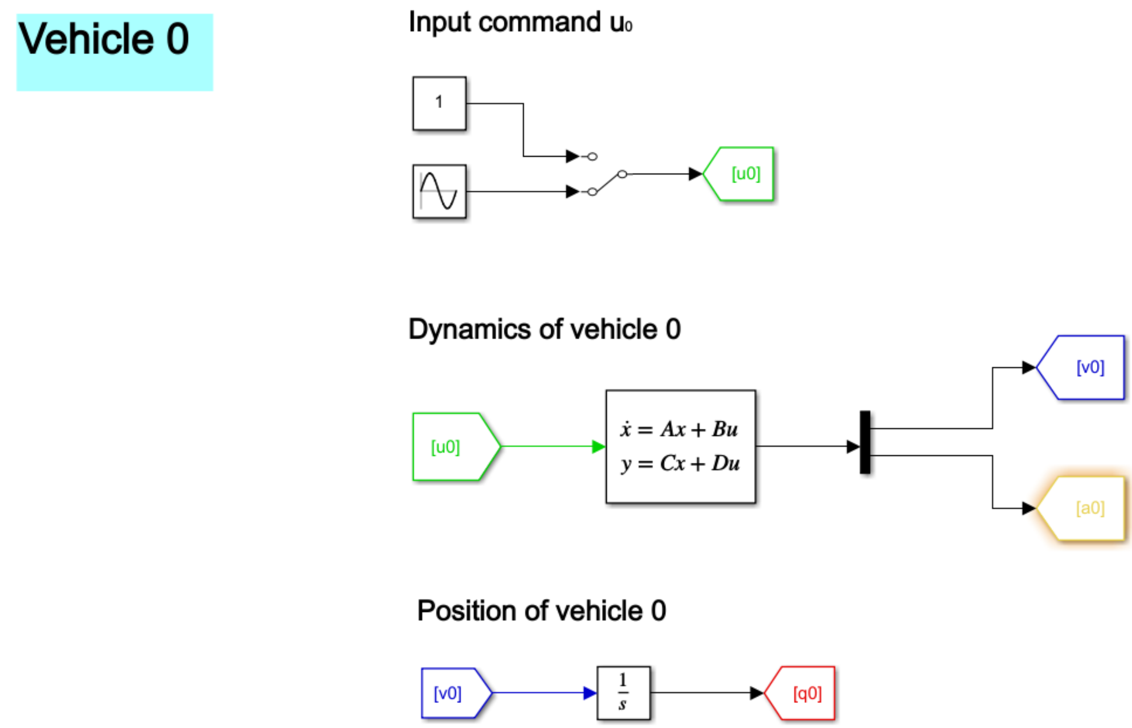
The principle remains the same, the 3 vehicles follow each other, each time keeping a constant delay over time. The acceleration values oscillate around 0, since once the cars stabilize at a certain speed, the acceleration becomes zero.

Finally, to know if the control law works correctly, we need to check that :

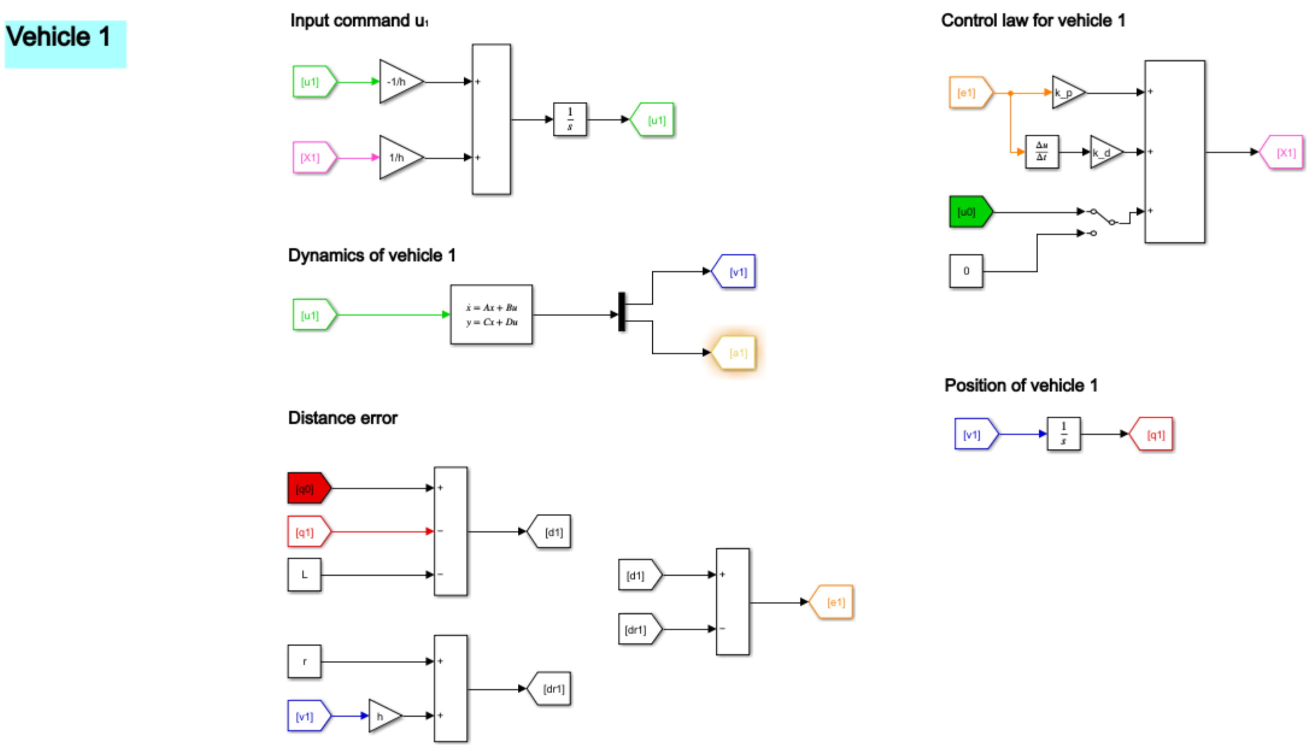
1) The error ei is equal to 0, which means that the distance between 2 vehicles is equal to the desired distance. This is the case for the 2 vehicles, the error is well stabilized at 0.

2) the accordion phenomenon doesn't occur, which it does.
   We can also try to simulate the system by removing the term ui-1 from the control law (which makes it possible to avoid the accordion phenomenon) and we observe the appearance of this phenomenon which disturbs the results.

The Simulink diagrams implemented to simulate the system described above are given below :

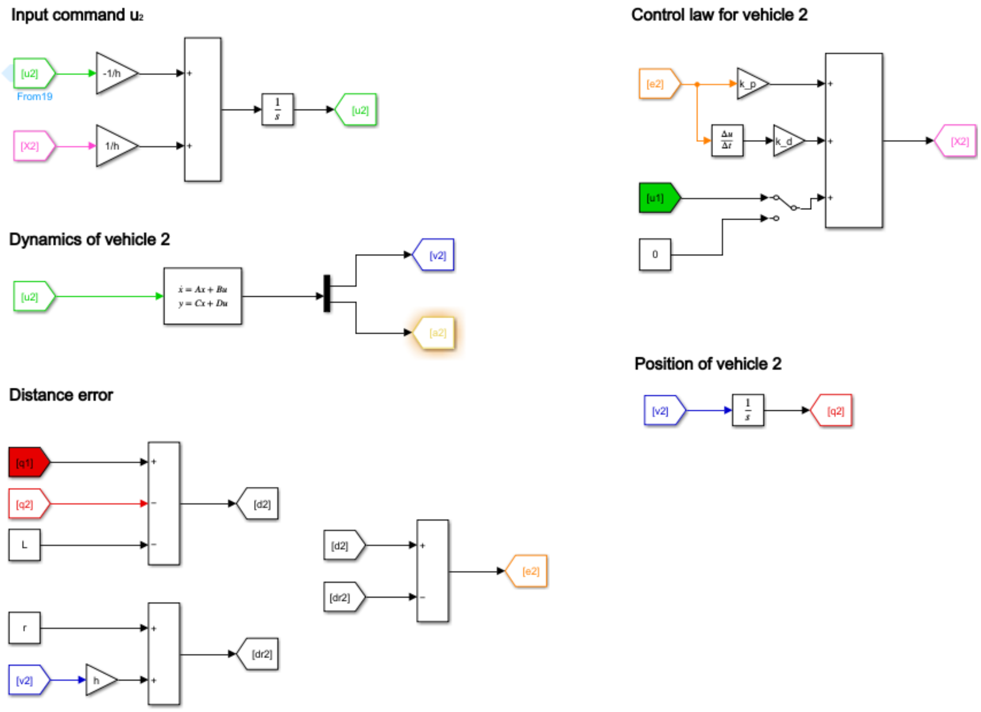


Blocks for vehicle 0



Blocks for vehicle 1

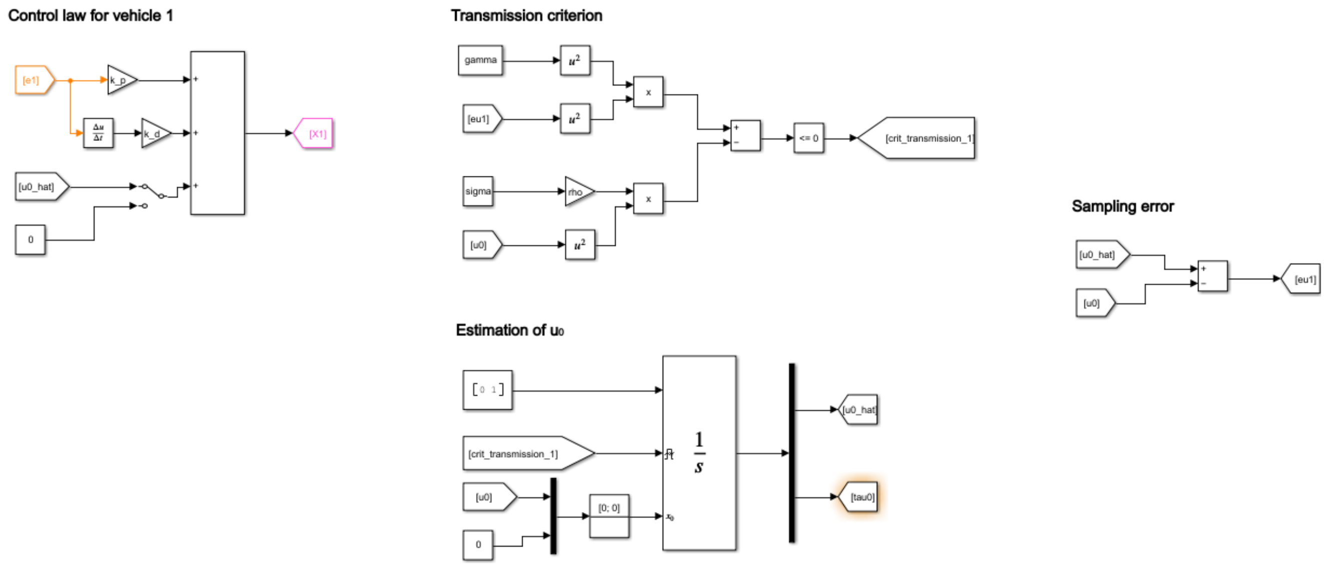
Vehicle 2



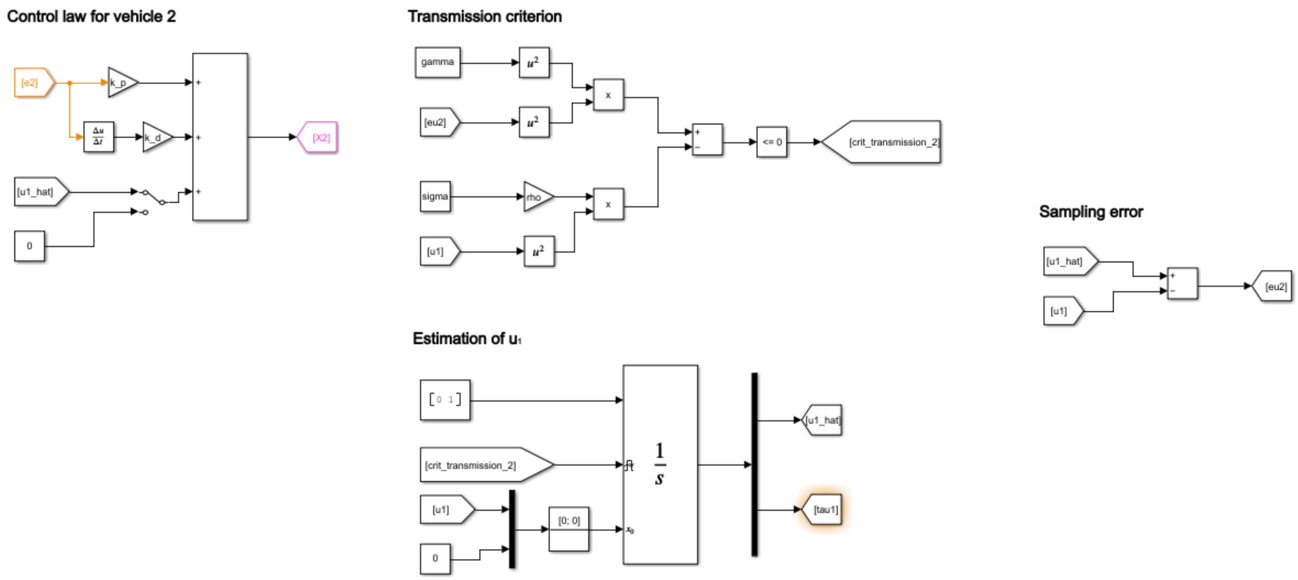
Blocks for vehicle 2

Now we want to simulate the system using the event sampling control law, using the transmission criterion we have defined.

To implement these changes in Simulink, we need to modify the blocks which concern the control law :



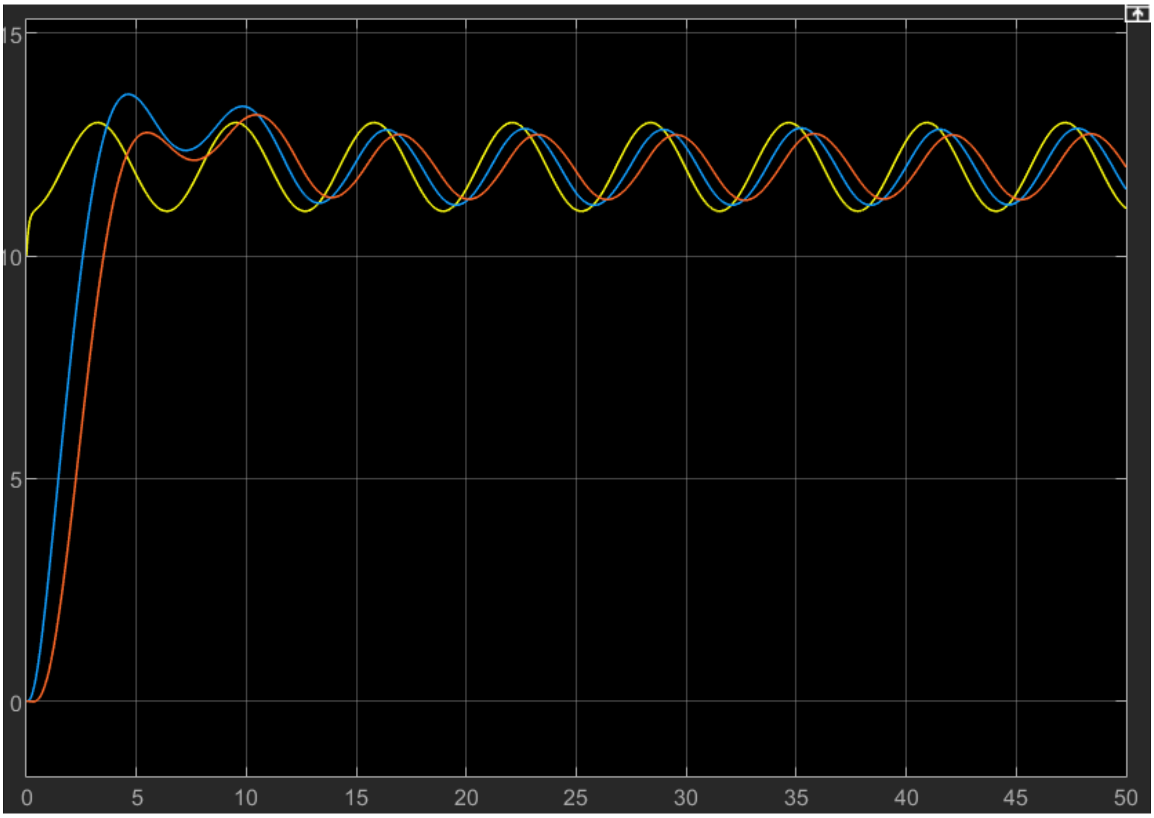
Blocks for vehicle 1



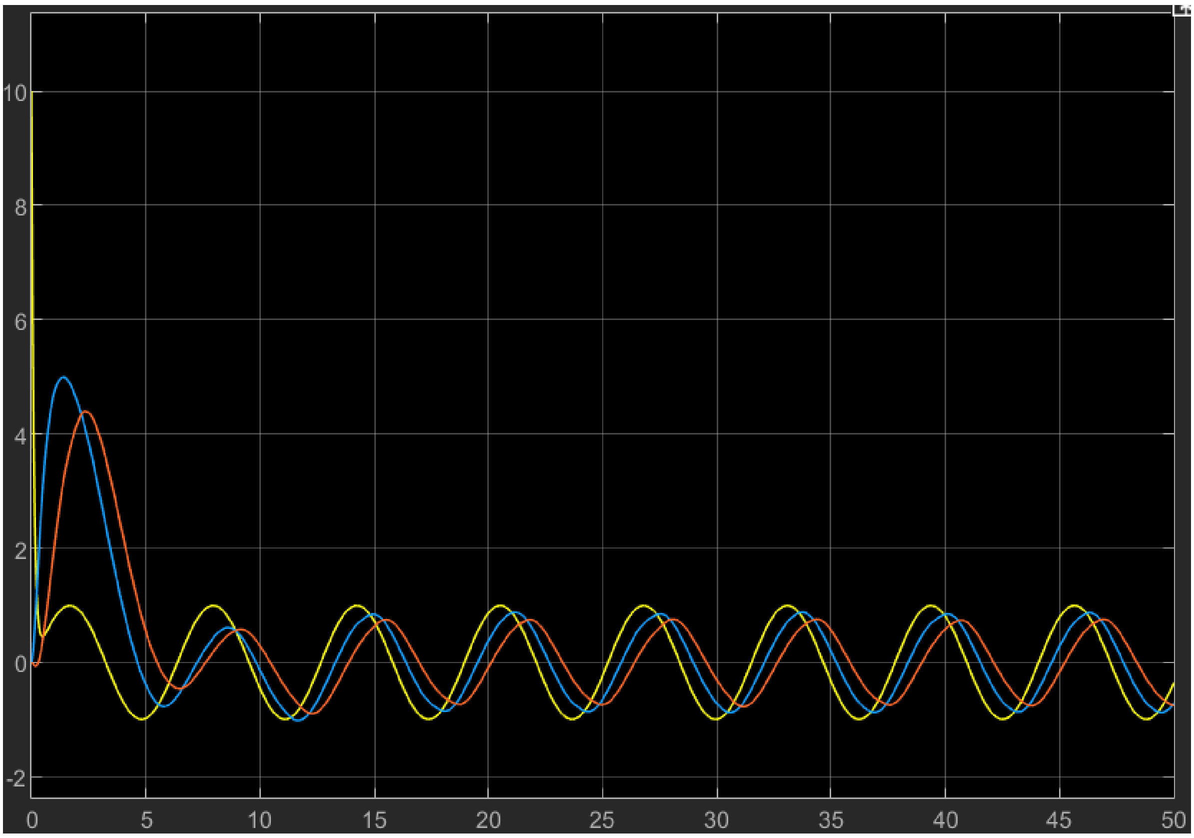
Blocks for vehicle 2

The simulation of the 3 vehicles and the plotting of the speed and acceleration curves gives us results similar to the previous test :

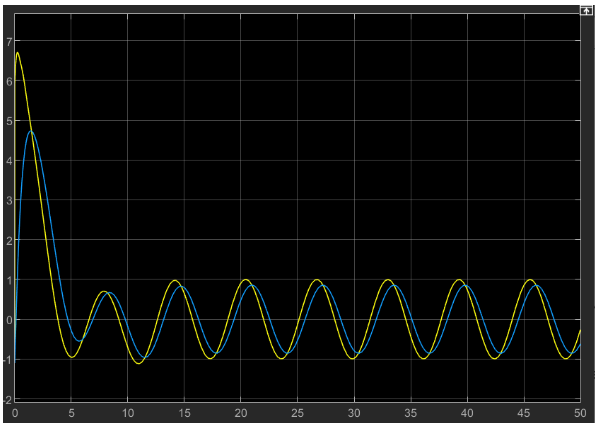
- Velocity :



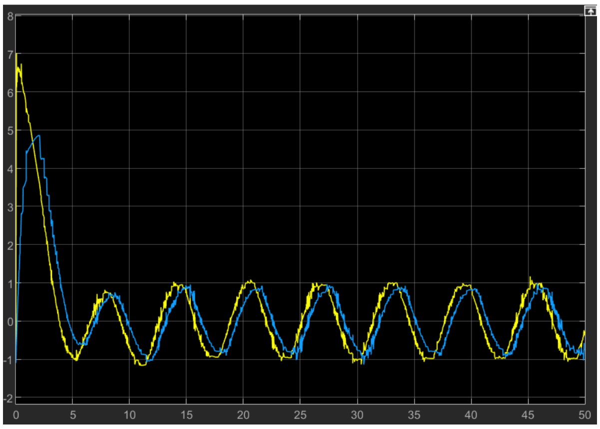
- Acceleration :



The difference between these 2 control laws is visible on the graphs which represent the evolution of the signals of the control laws for vehicles 1 and 2 over time :

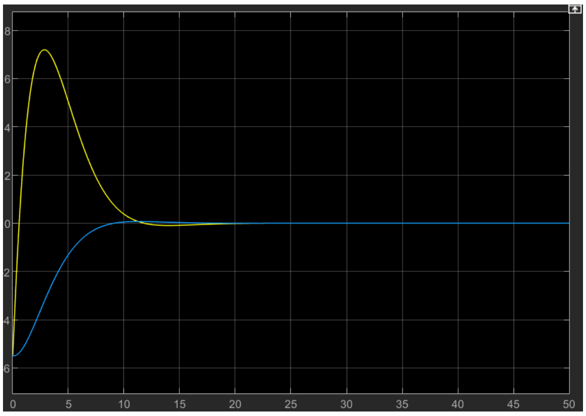


"Classic" control law

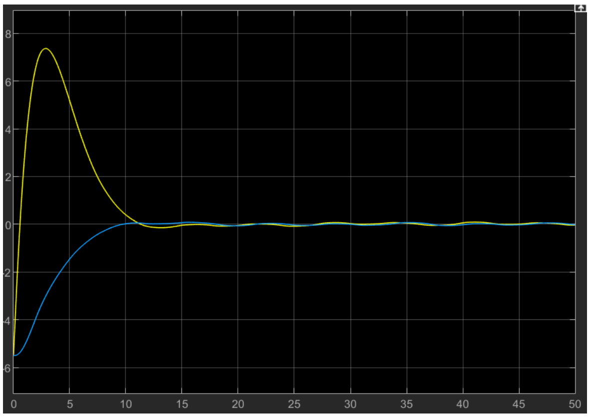


Sampling control law

And the consequences on the error :



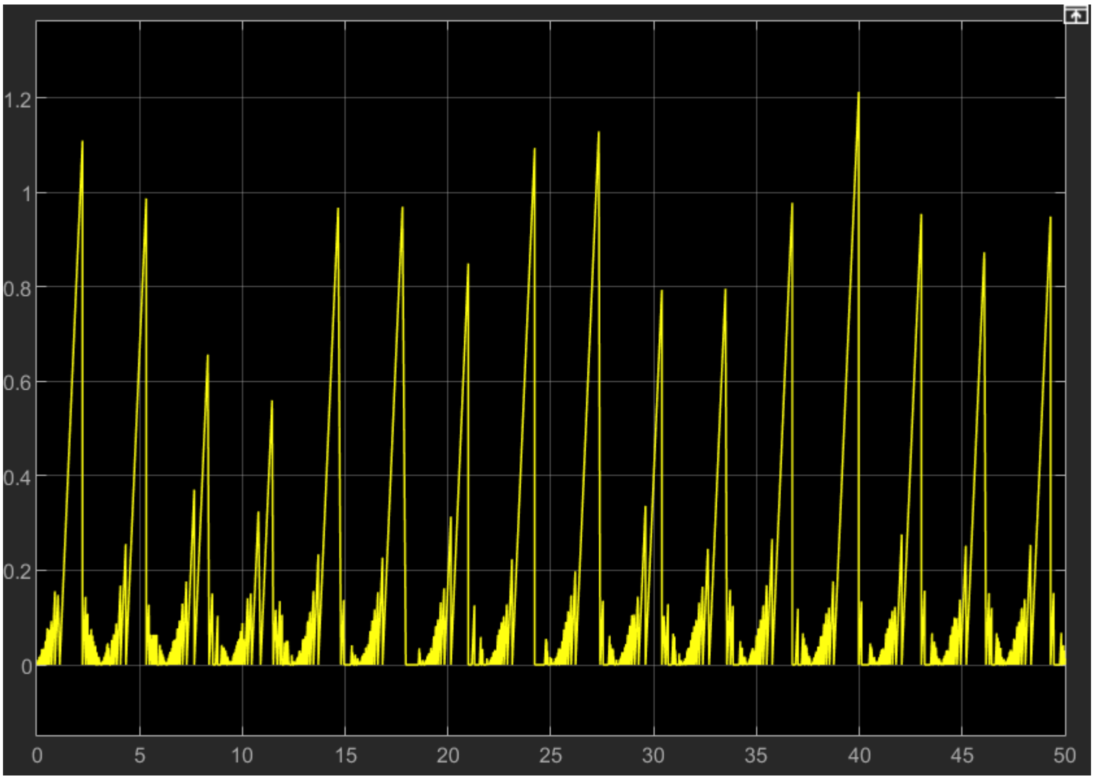
"Classic" control law



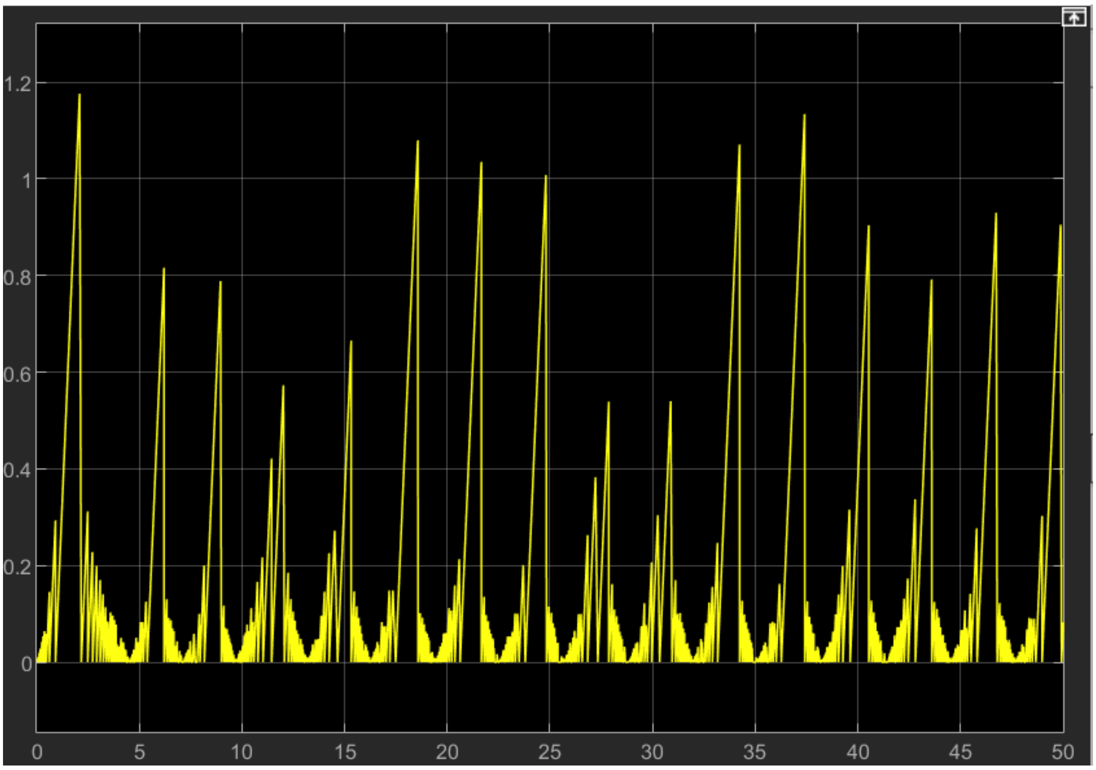
Sampling control law

We can observe oscillations during the simulation with the sample control law, due to the fact that we are not transmitting continuously, but indeed on events defined by the transmission criterion.

We can also represent the transmission times :



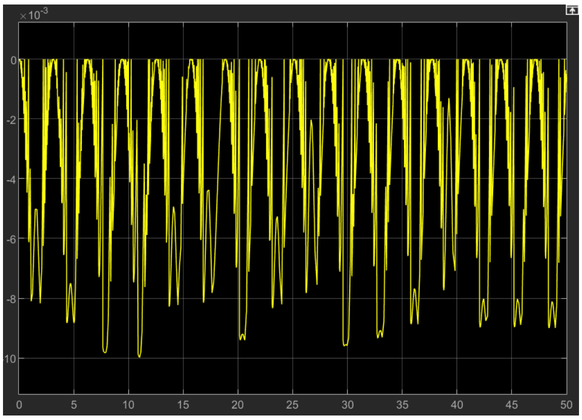
Vehicle 1



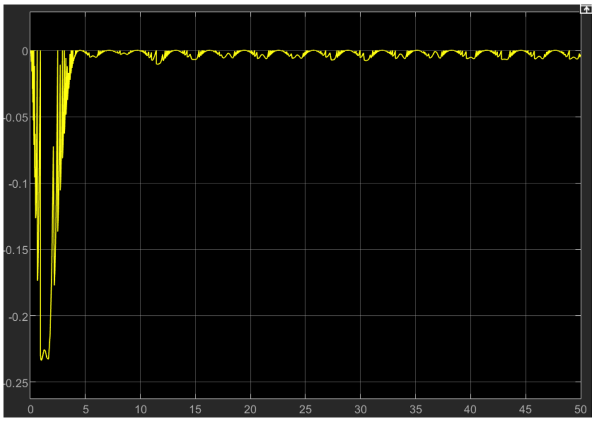
Vehicle 2

Where we can observe periodic transmission peaks which are repeated periodically, approximately every 3 time units.

Between them, we can observe weak transmission peaks. However, the transmission seems to be taking place at all times, as the transmission criteria are always true, as shown in the following graphs



Vehicle 1



Vehicle 2