

|  |
| --- |
| MATLAB PROJECT REPORT |
|  |
| 16th DECEMBER 2019  LORRIAUX Tristan  MAISONNAVE Lucas |



Project objective

         For this project we will focus on a sensorless speed regulator without a physical rotational speed estimator. This regulator works by estimating the rotational speed of the propeller: it will be developed on Matlab. This will allow the control of the thrusters to free itself from the gradual drop in battery voltage, while avoiding a voltage regulator, which would weigh down the robot.

I - Thrusters modelisation

First it is necessary to know the characteristics of the motor: *Lm , Rm , f ,  J* et *Kem* (respectively the motor inductance, internal resistance, friction coefficient, inertia and electromagnetic coupling constant). We calculate them with the technical documentation and formulas given, we get them:

# 

Figure 1- Charachteristics of the motor

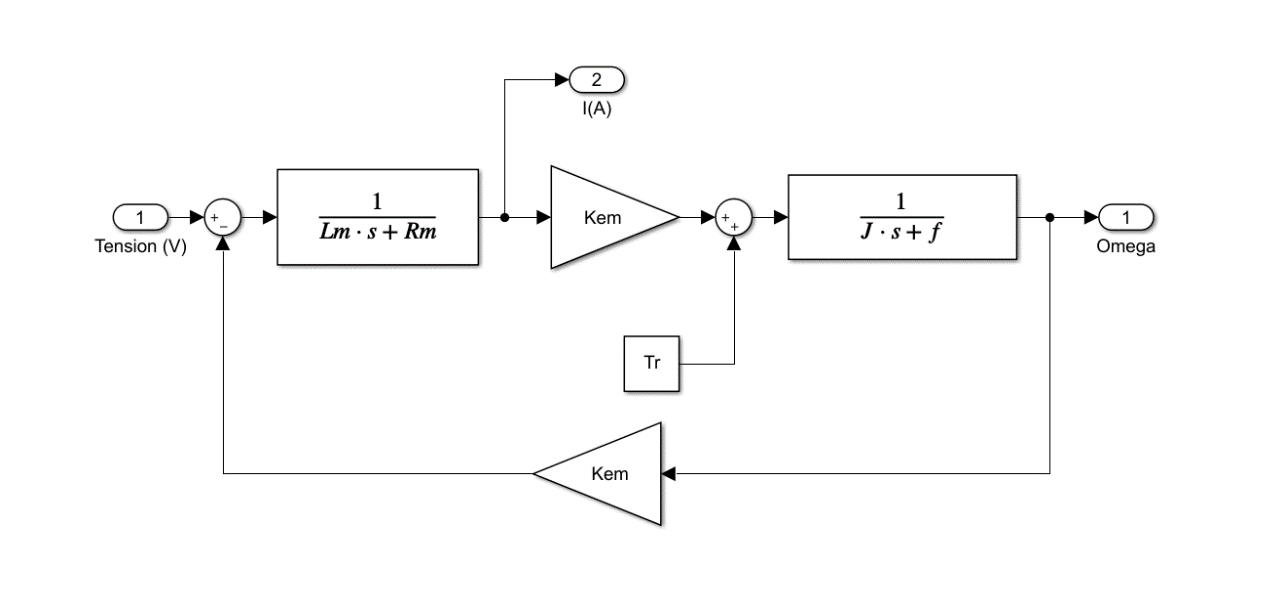
The dynamic behaviour of the thrusters in Matlab is then modelled using the following structure:

Figure 2 - Thrusters dynamic behaviour in Simulink

Tr is set to 0 because it does not consider disturbances. We get an output which is the speed of rotation of the motor as well as the armature current that will be required for the speed estimator. The motor is controlled in voltage in PWM (Pulse Width Modulation) which is generated by a chopper.

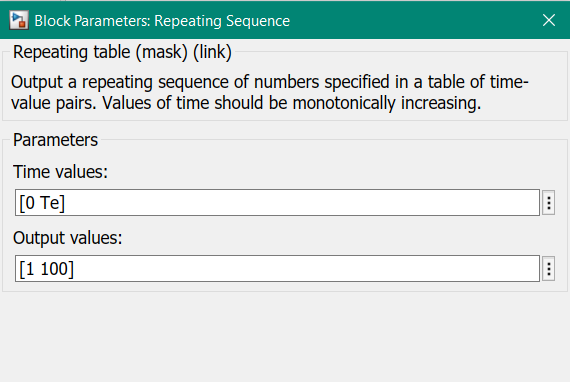
**II - Chopper’s modelisation:**

The chopper's role is to generate a PWM signal to modulate the motor input voltage. To do this, we will have to take into account the decrease of the battery’s voltage that will be modelled by a ramp. For this simulation we will consider that the battery decreases by 1.1V in 5s (15 min in reality) to have a reasonable time scale. The initial battery’s voltage Ubat is 8.2V and is configured as follows:

# 

Figure 3- Ramp parameters and triangle generator parameters in Simulink

Figure 4 - Ubat Slope



The PWM’s frequency is set to 20 kHz with much bigger than the electrical constant of the motors (62.5Hz). The structure of the chopper is then modeled as following:

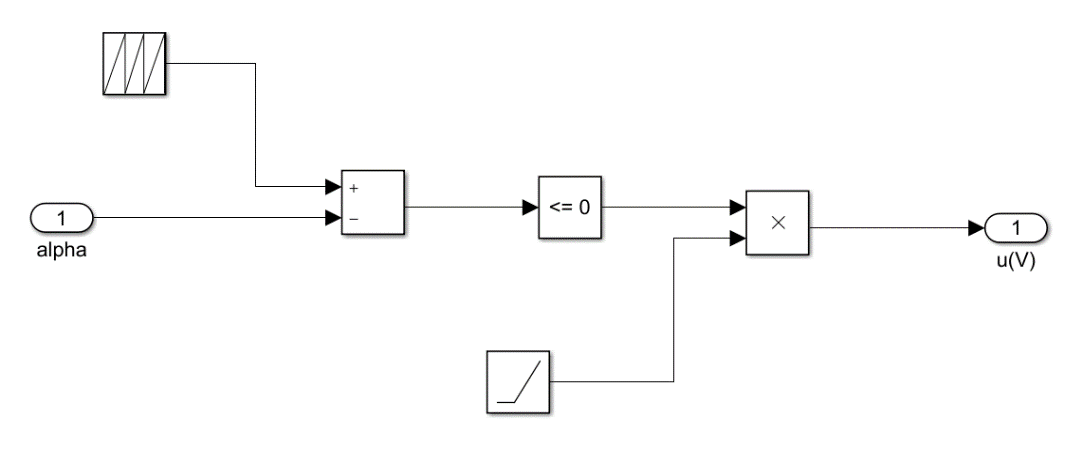


Figure 5 - Chopper's structure in Simulink

We set the triangle generator to a frequency of Fe and a value between 0 and 99. With this model we control the duty cycle of the PWM, and we obtain for different values of alpha:

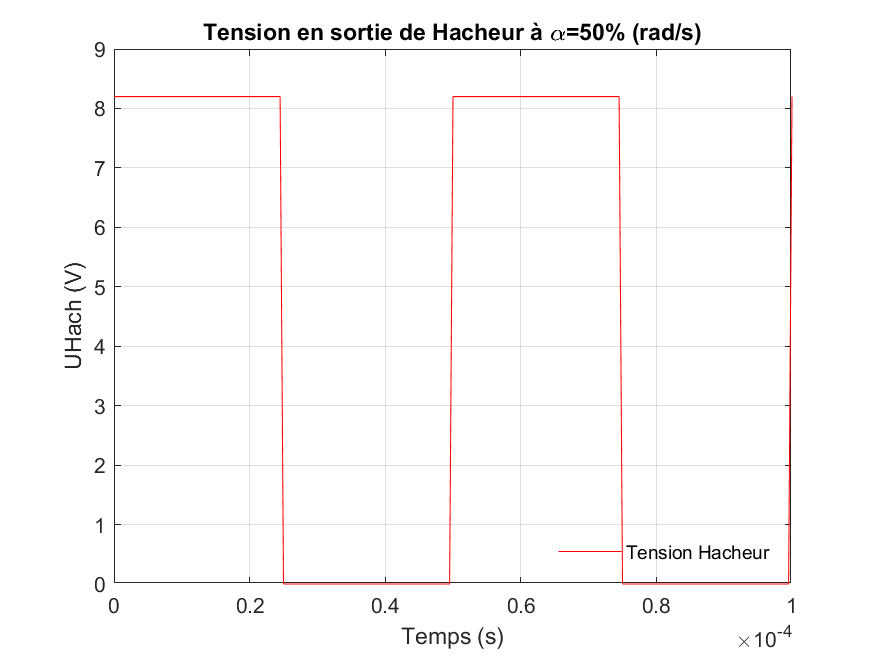
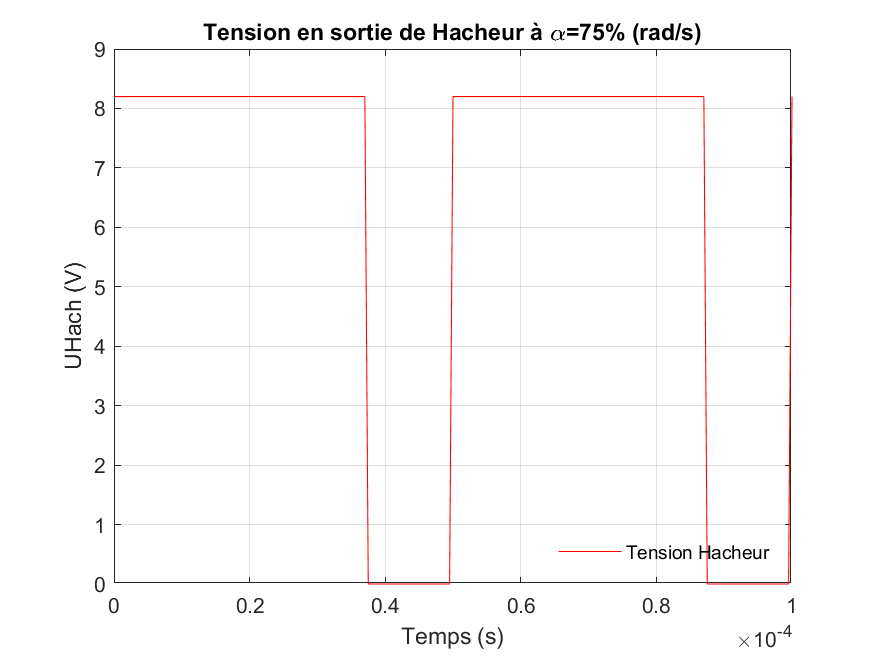
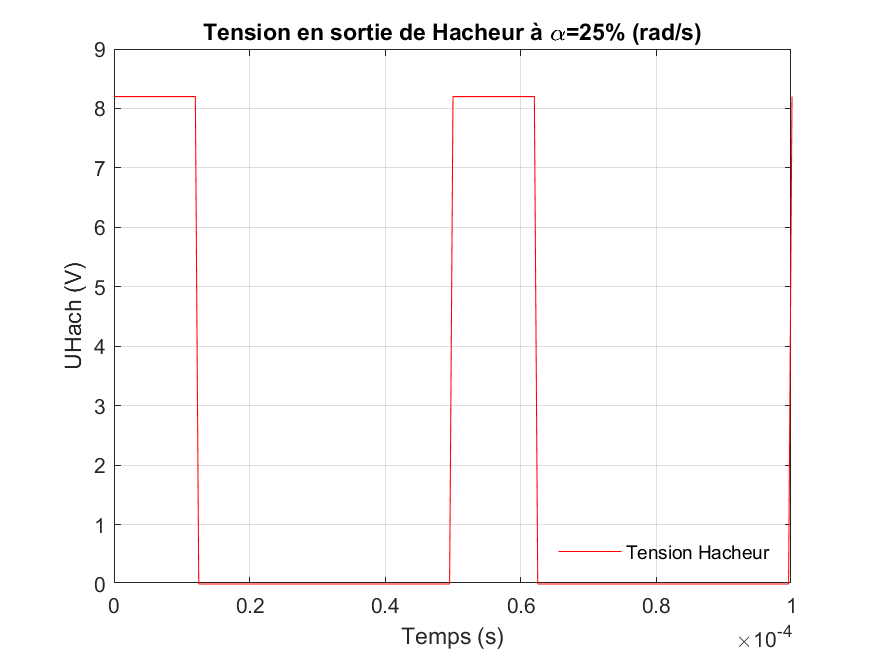


Figure 6 – Outputs chopper’s voltages function of alpha

The calculation step of the simulation has to meet the demands of the subject: a calculation accuracy a calculation accuracy on the simulation variable of 1%. So, we calculate Ts:

III - Regulation loop

A - Rotational speed estimator:

In order to create a regulation without sensors we need to build a rotational speed estimator. This estimator is modeled with the following structure:

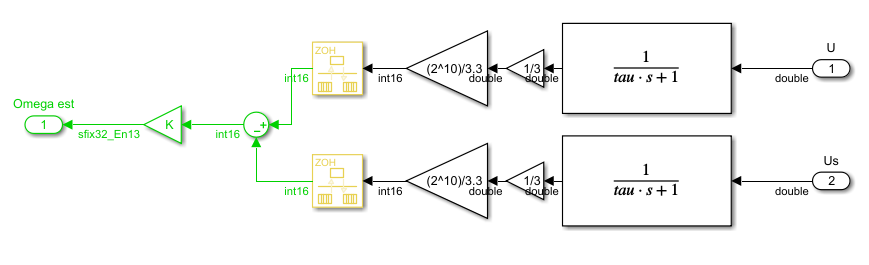


Figure 7 - Rotational speed estimator's model

The 2 inputs are the motor control voltage and the armature voltage, the low-pass filter is configured to let pass the average voltage of the PWM with . Now we configure the rate transition at 1 ms as specified by the subject. In order to set the value of K we look at the value of the regulator and the output of the dynamic model for the same instruction and we set K to reduce the gap between the two. We get :

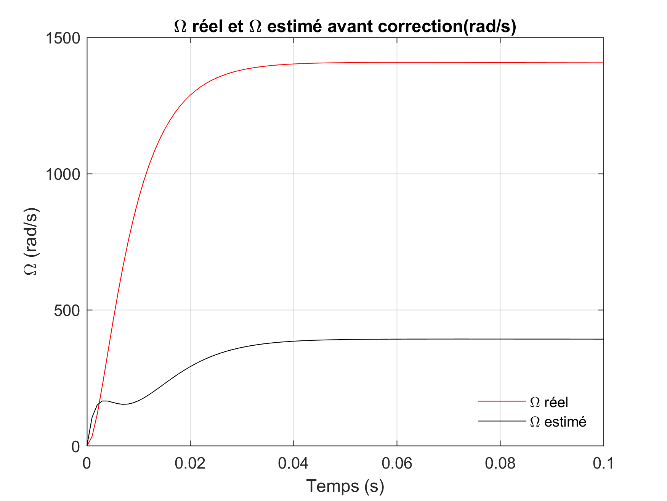
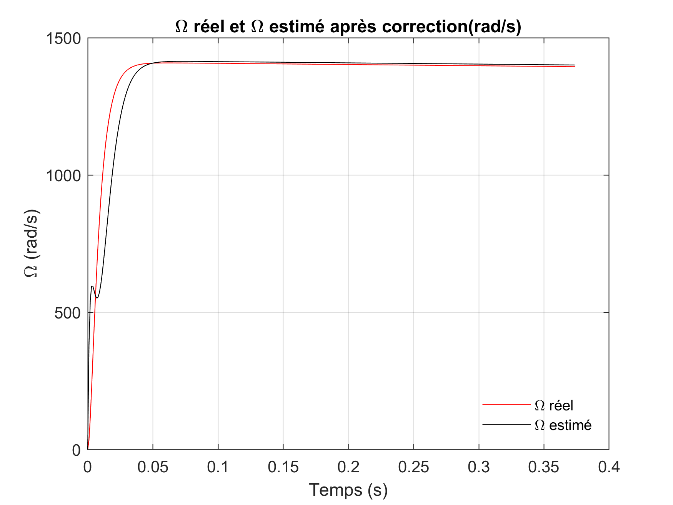


Figure 8 - Ω estimated and real

So, we set:

B - Corrector:

The final step is now to configure the corrector of the regulation loop with an IP corrector, we also add a saturation and a rate transition block. The rate transition block is set to Ts and the saturation to 100 in order to avoid the case at the beginning when alpha is greater than 100. The gain of the corrector is determined by taking care of the overshoot which have not to exceed 5% of the final value. So, we tried different value and we finally get

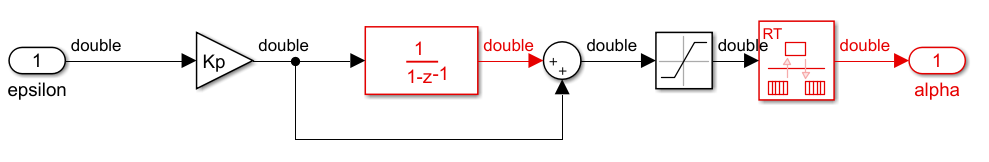


Figure 9 - Corrector's model

IV - Robustness test:

Now we are able the gather the different blocks together the get the full regulation and we can test the model:

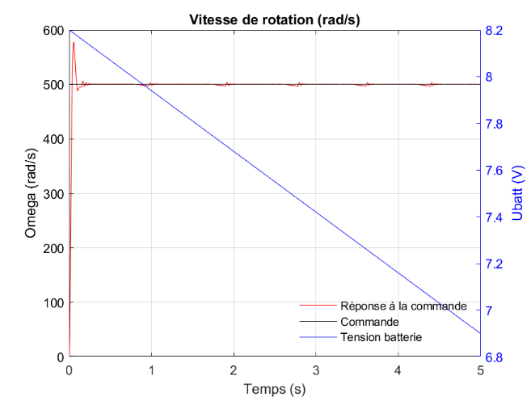
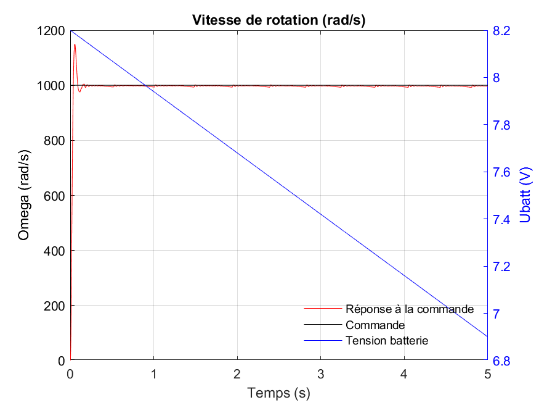
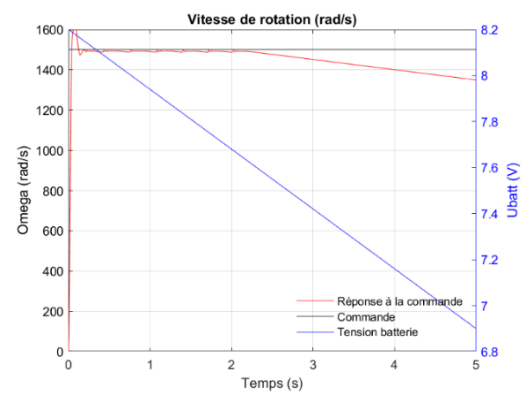
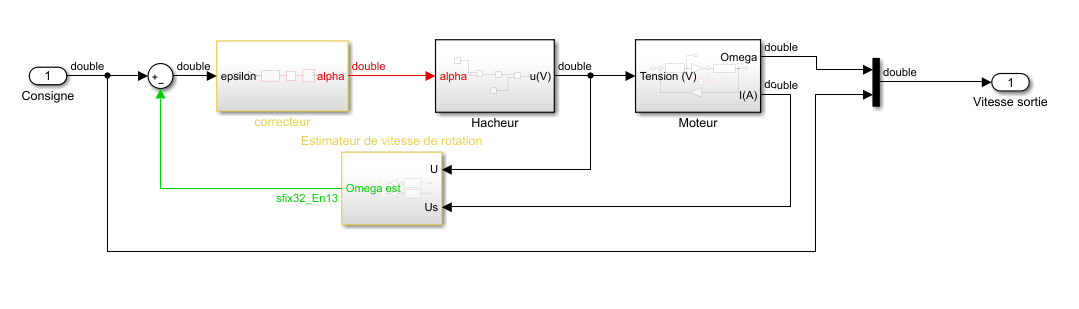
Then we are going to test the model for different rotation speeds: 500, 1000 and 1500 rad/s:

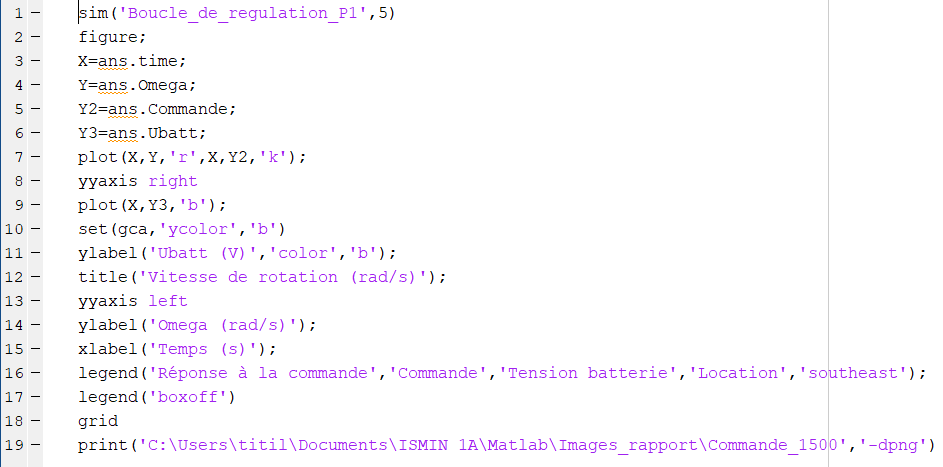
Figure 11 - Output speed for different inputs

Figure 10 - Final model

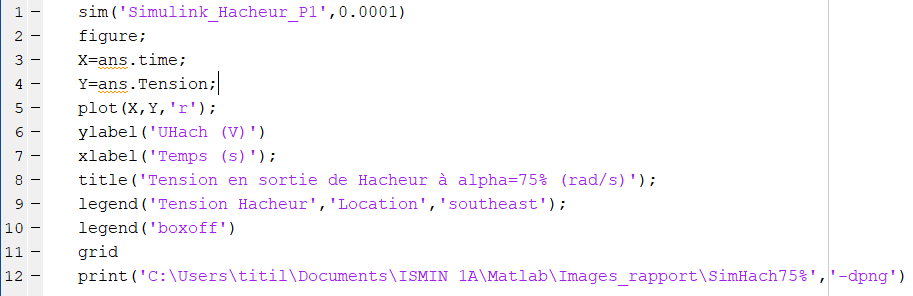
We notice that for a rotation speed of 1500 rad/s.

ANNEXES

*MATLAB code to display final graphs:*

**

*MATLAB code to display chopper’s graph:*



*MATLAB code to display the difference between actual and estimated omega:*

