

ÉTUDE D'UN SYSTÈME DE CONTRÔLE ÉLECTRONIQUE DE LA STABILITÉ ET D'ANTIPATINAGE (ESC/ESP)

SUJET ASSOCIÉ AUX THÈMES :
SCIENCES INDUSTRIELLES -
AUTOMATIQUE ; PHYSIQUE –
MÉCANIQUE &
INFORMATIQUE -
INFORMATIQUE PRATIQUE

AOUDIA SAMMY - 12315



SAMMY AOUDIA

SOMMAIRE

Introduction

Modèle Physique

Modèle Informatique

Asservissement

Synthèse

I - HISTORIQUE

~1995 par Bosch

-80% Accidents par perte de controle

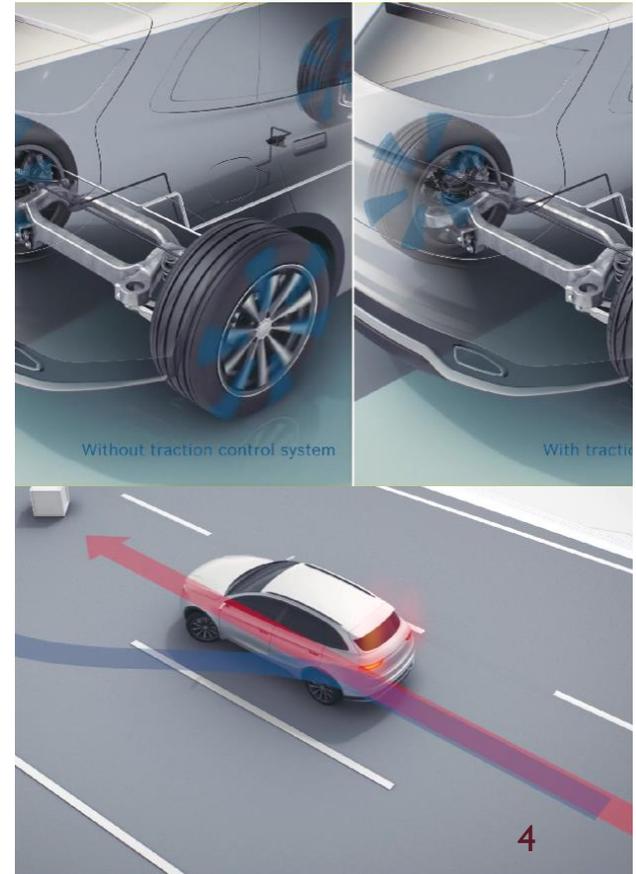
-4000 morts/an (Commission européenne)

Obligation : 2012

I – CONTEXTE

Contrôle de traction

Correction de trajectoire



I – CONTEXTE

Contrôle de traction

- Eviter les pertes d'adhérence

Correction de trajectoire

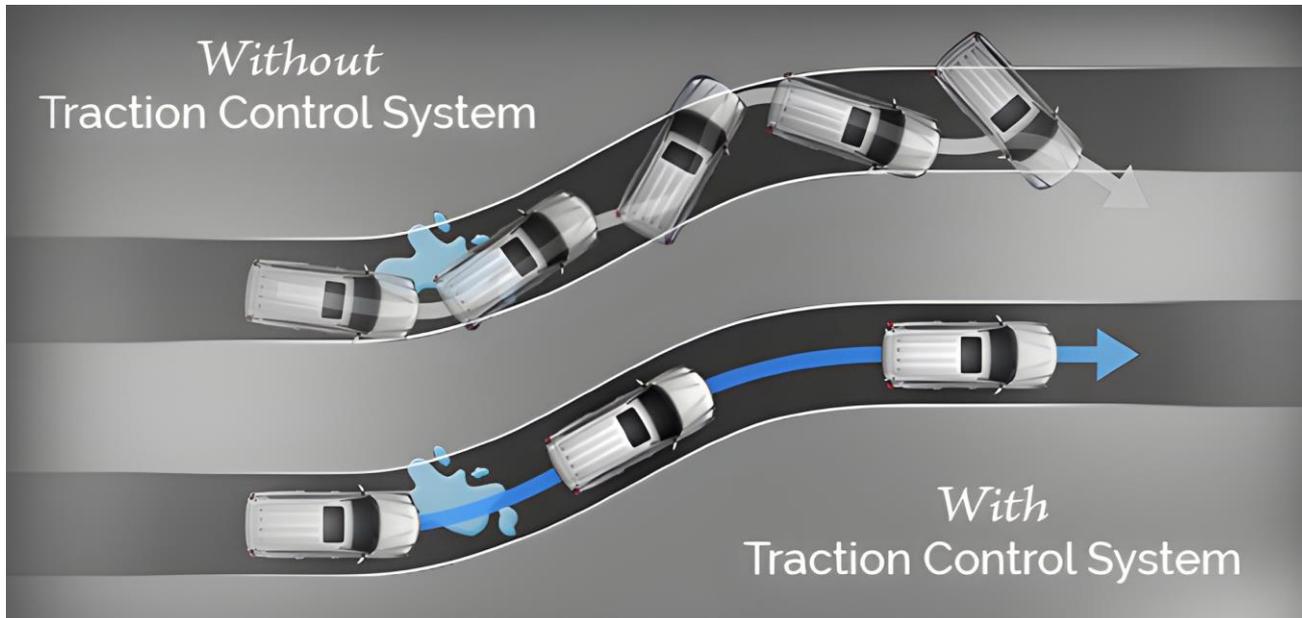
- Maintenir une trajectoire en virage

I – CONTEXTE

Sous-virage ou
Survirage



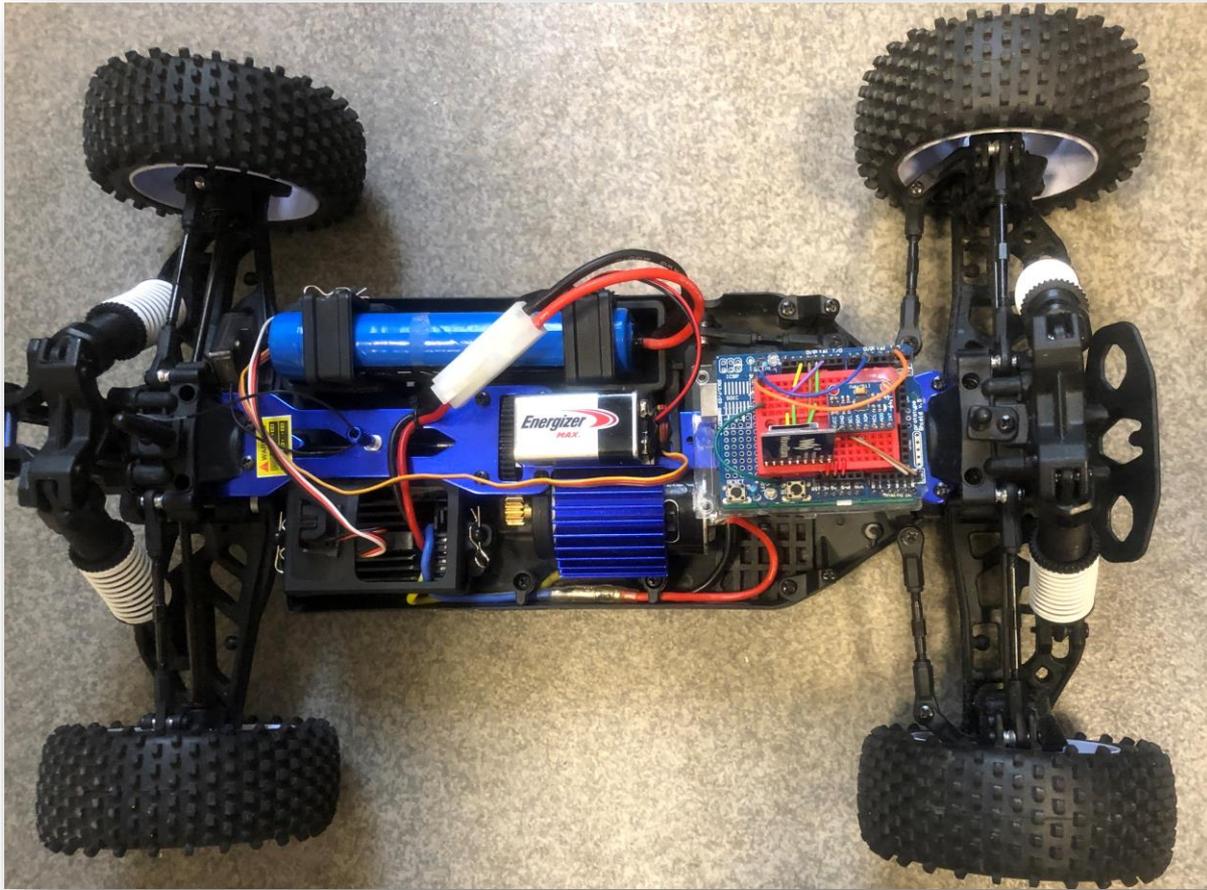
Risque de perte de
contrôle voire
d'accident



I – OBJECTIFS DE L'ÉTUDE

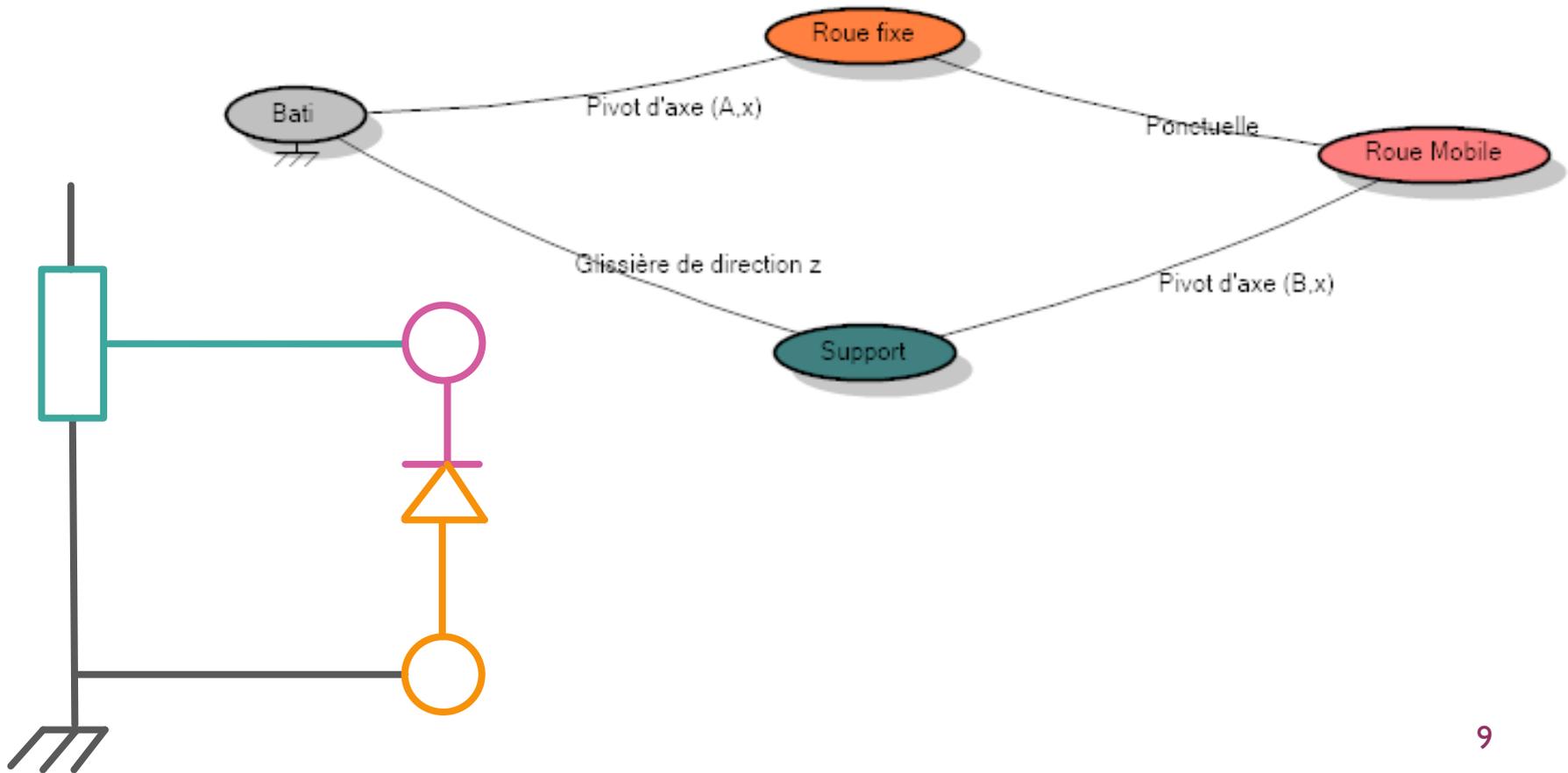
- Modéliser le système
- Détecter un glissement
- Asservir le système sans modifier la consigne

I - MAQUETTE DU SYSTÈME

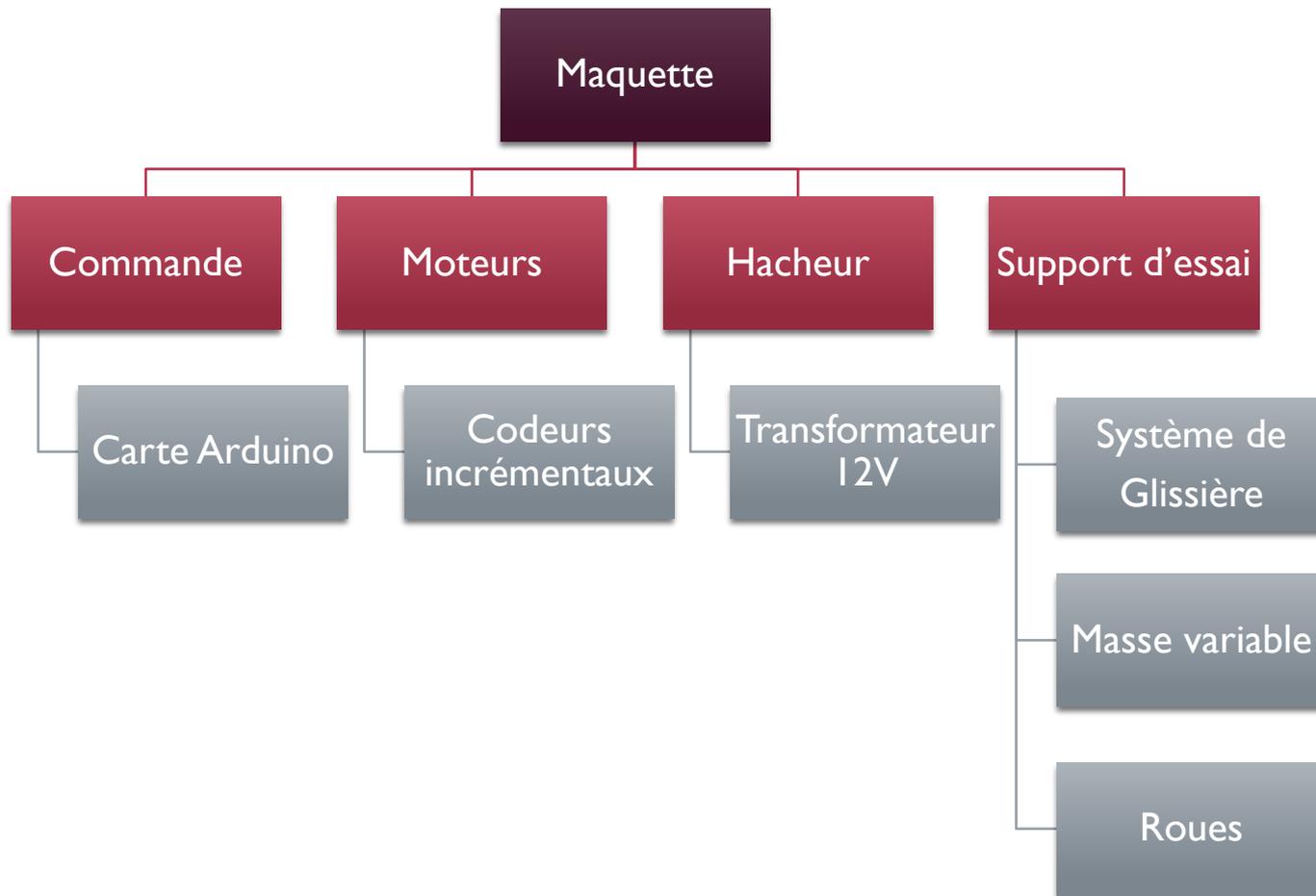


SAMMY AODIA

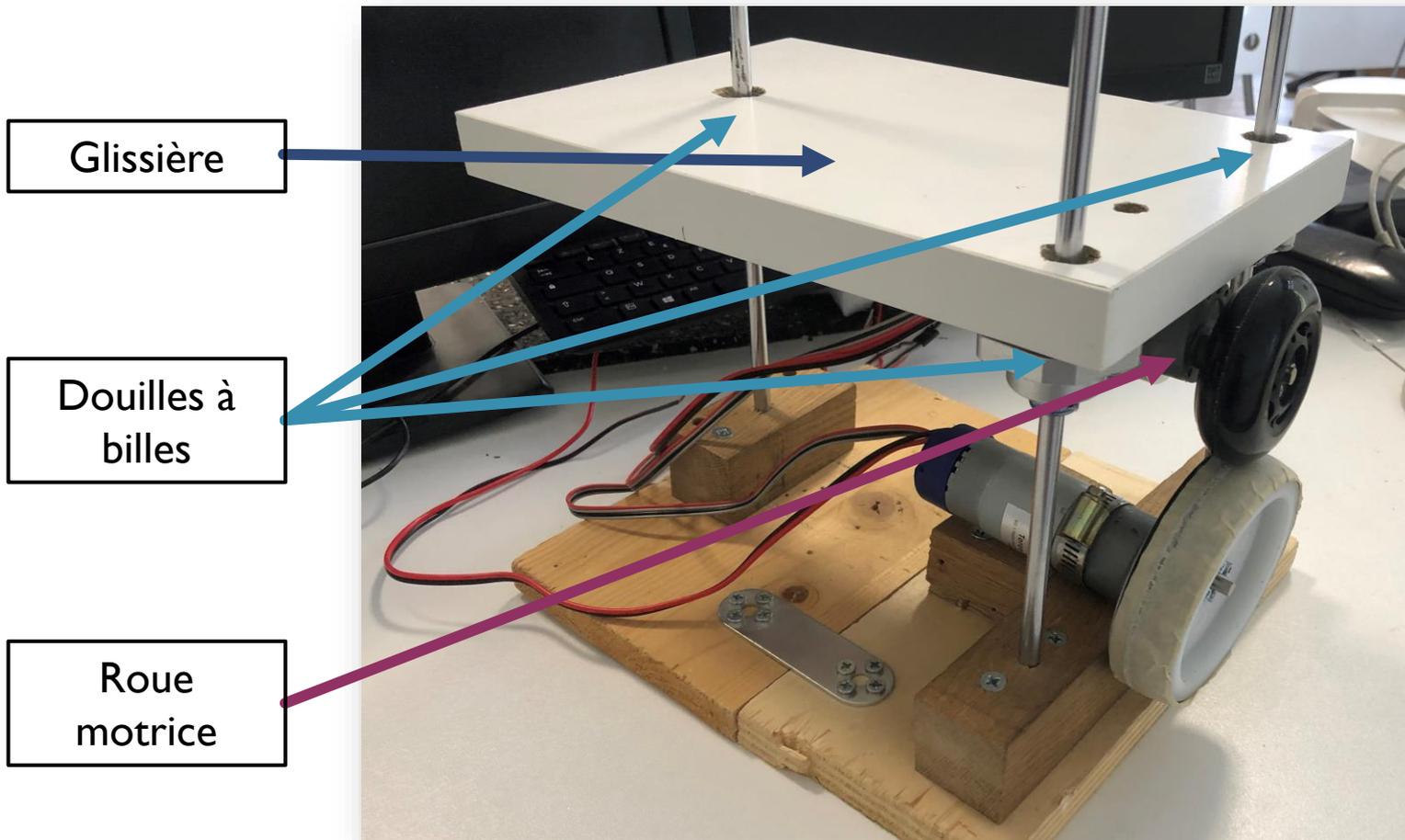
I - MAQUETTE DU SYSTÈME



II - MAQUETTE DU SYSTÈME



I - MAQUETTE DU SYSTÈME



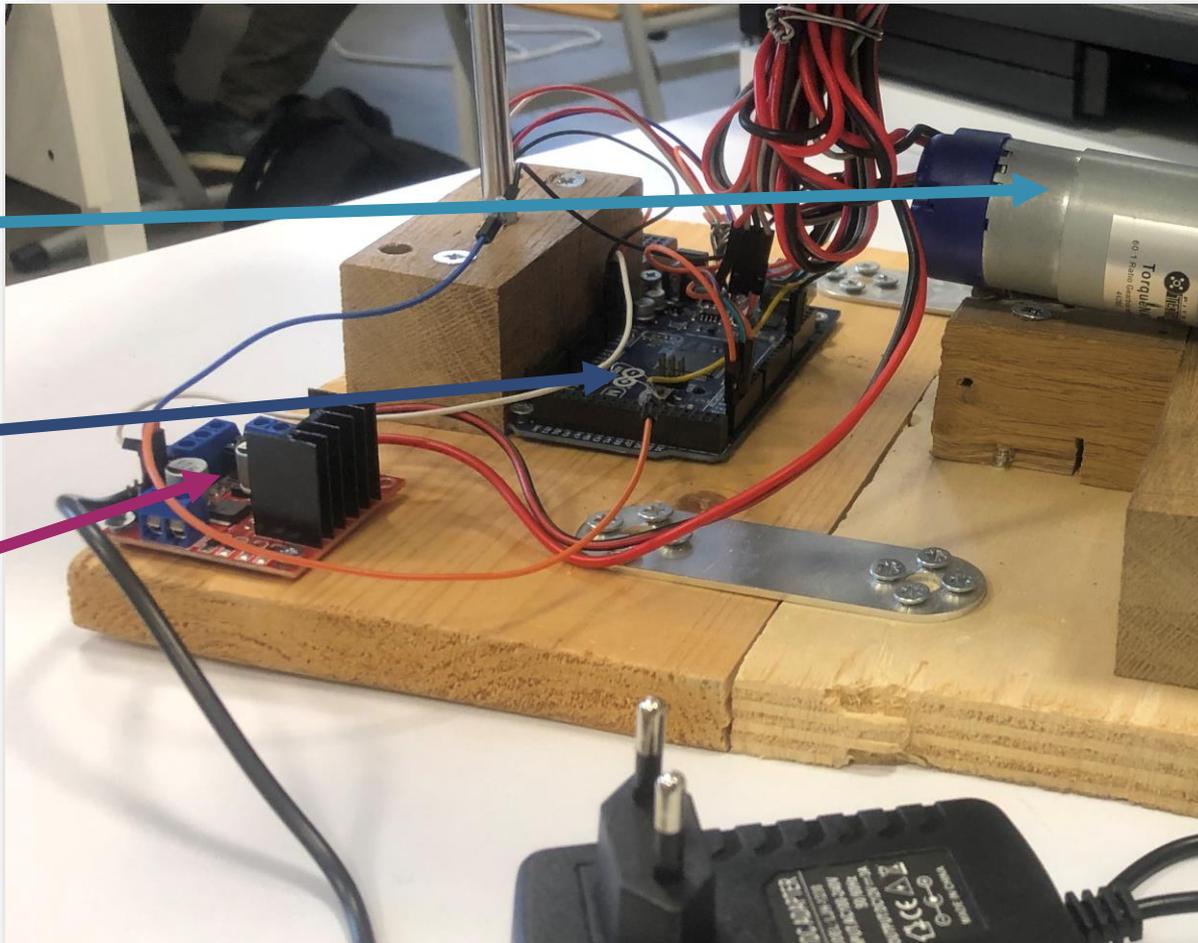
SAMMY AODIA

II - SCHÉMA ÉLECTRONIQUE

Moteur
/
Codeur

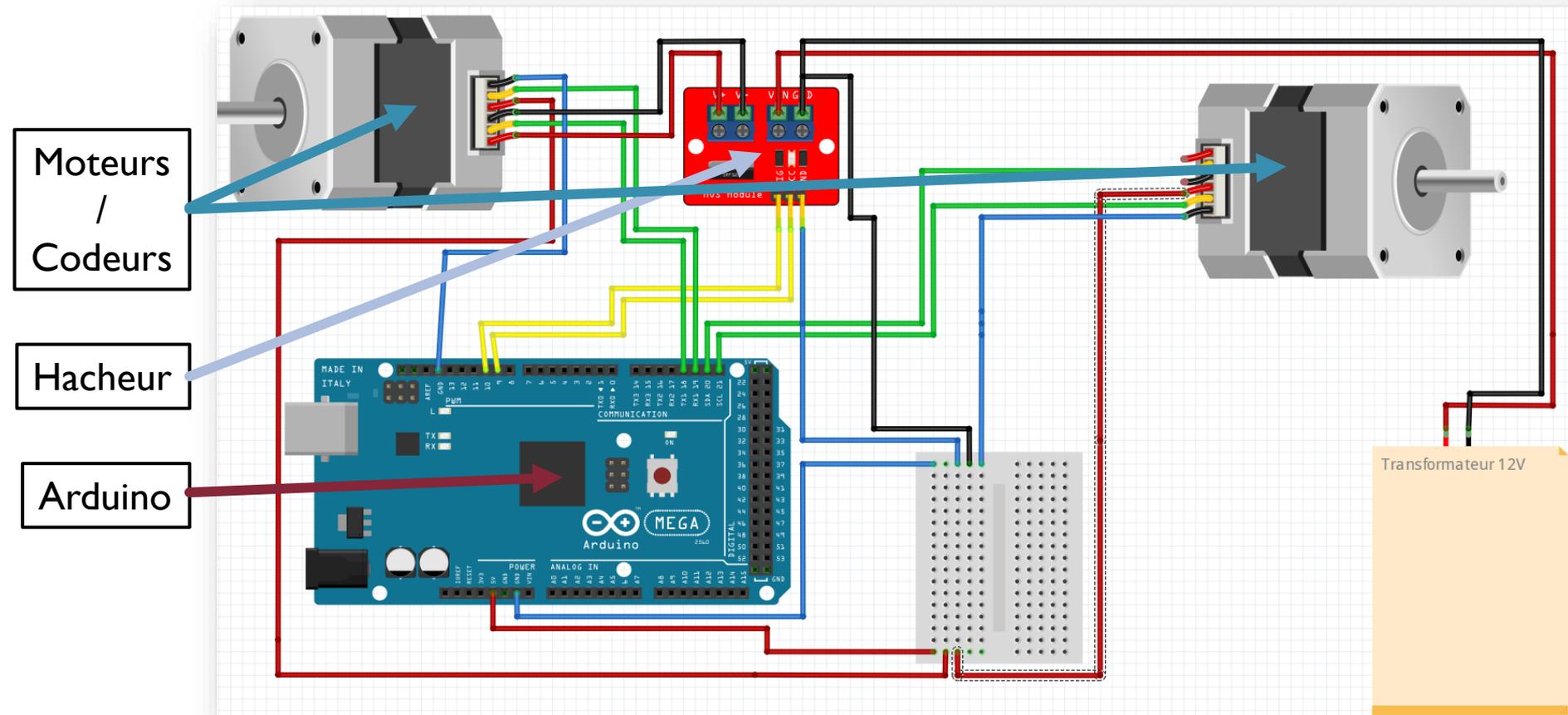
Arduino

Hacheur



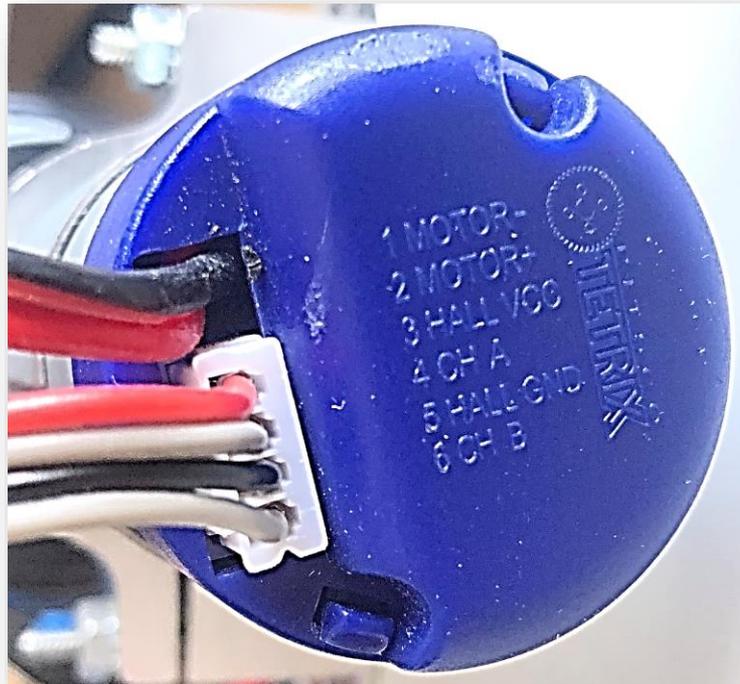
SAMMY AODIA

II - SCHÉMA ÉLECTRONIQUE

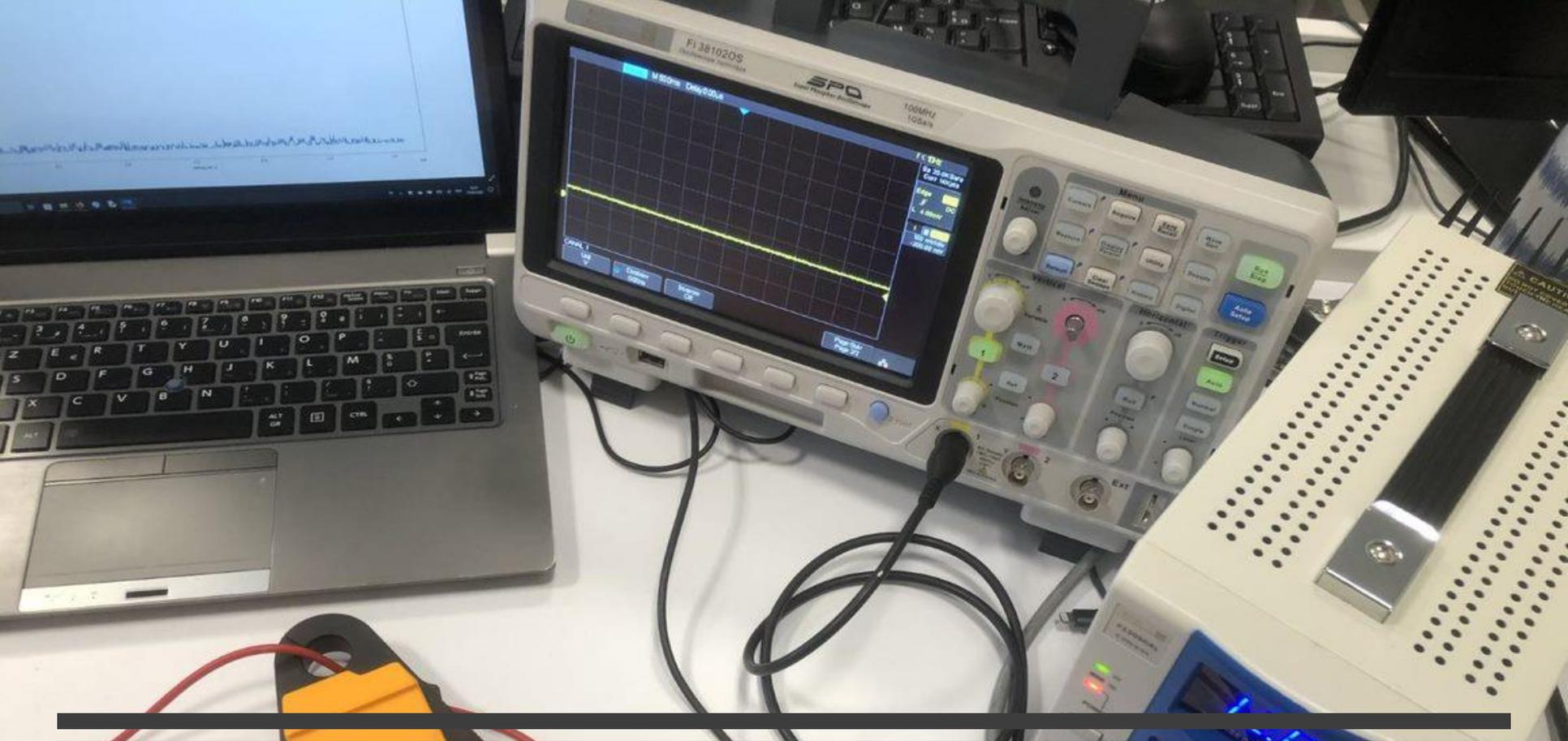


II – IMPLÉMENTATION DU CODEUR INCRÉMENTAL

$$\omega_i = \frac{Tops_i - Tops_{i+1}}{dt} \times \frac{2\pi}{Nombre_Tops_Par_Tour} \text{ [rad. s}^{-1}\text{]}$$

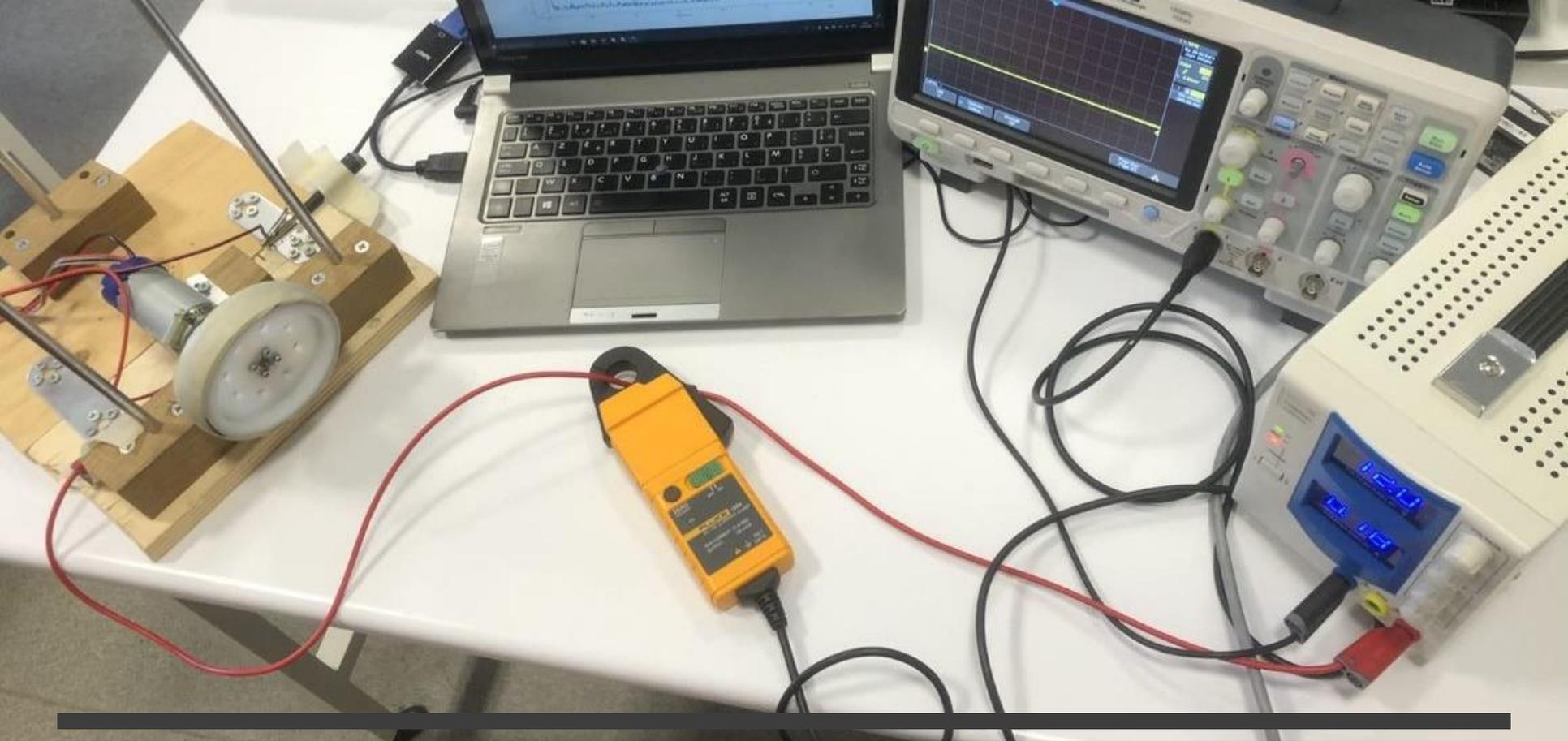


SAMMY AODIA



II – CARACTÉRISATION DU MOTEUR

- $E = k \cdot \omega$
- $Cem = k \cdot I$
- $J \cdot \frac{d\omega}{dt} = Cem - Cr$
- $U = E + R \cdot I + L \cdot \frac{di}{dt}$



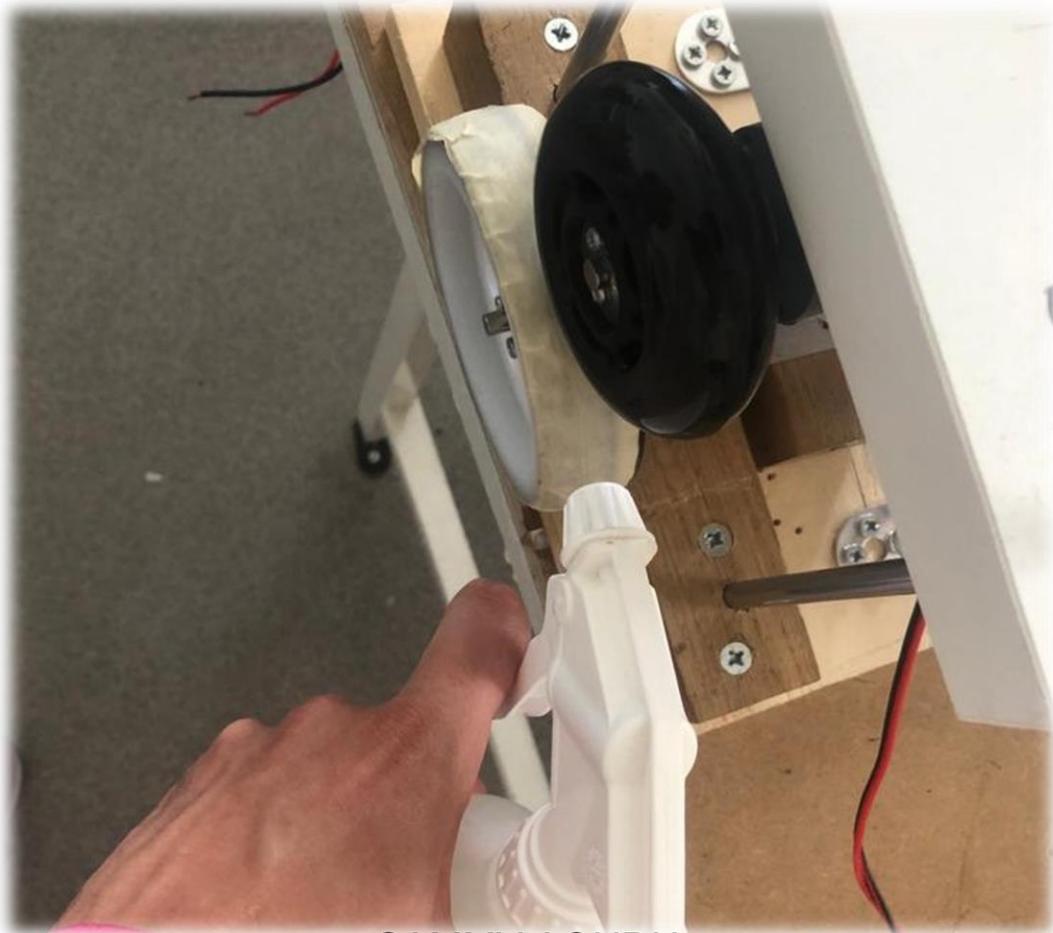
II – CARACTÉRISATION DU MOTEUR

- *Mesure rotor bloqué* $\rightarrow R = 2,29 \Omega$
- *Mesure à vide* $\rightarrow k = 1,11 V.s$
- *Mesure de $t_{5\%}$* $\rightarrow J = 1,2 \times 10^{-6} kg.m^2$

III – SITUATION ÉTUDIÉE

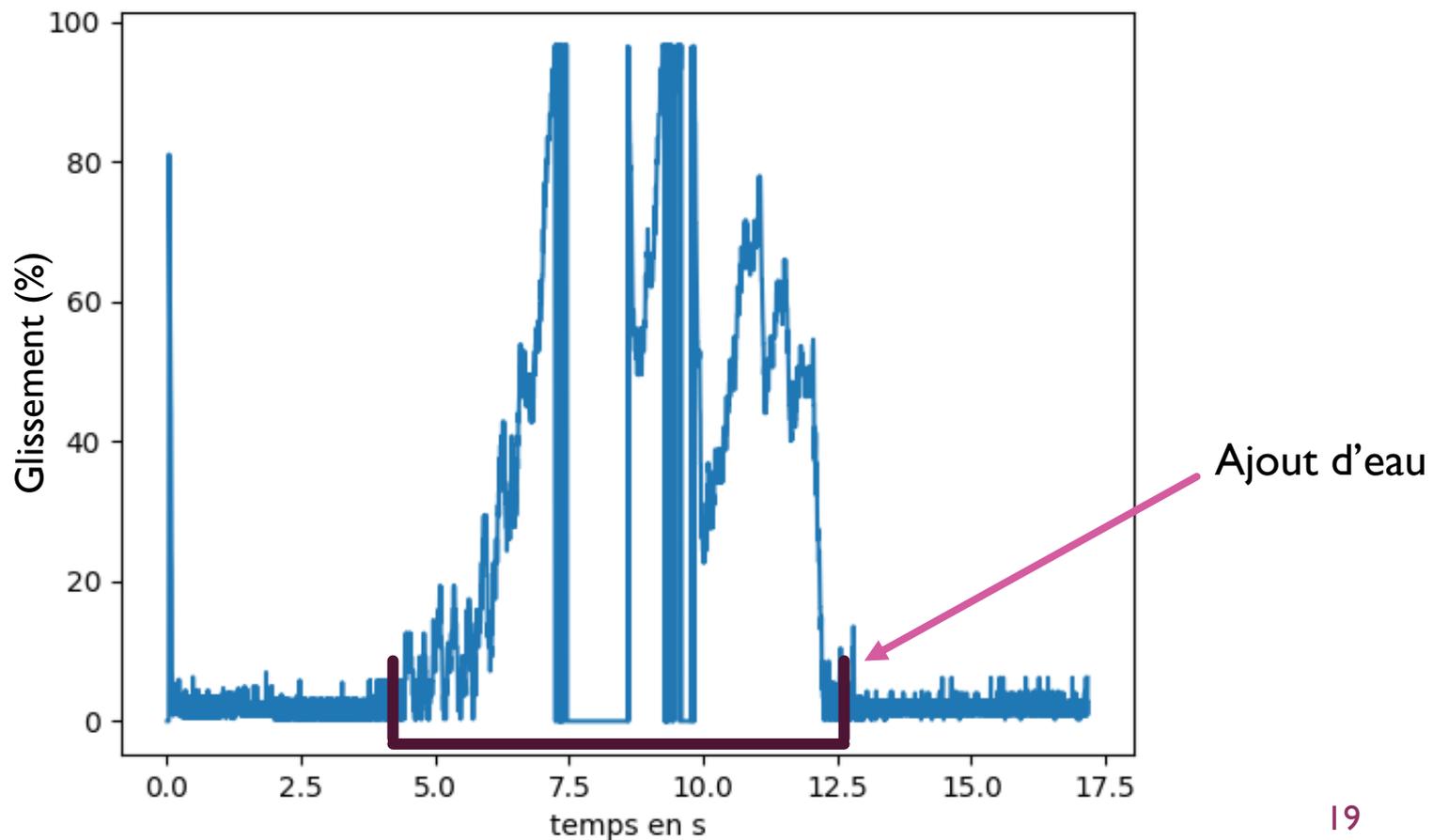
- Départ avec accélération maximale
- Phase à vitesse constante
- Changement de type de surface (Chaussée mouillée)
- Accélération jusqu'à la vitesse maximale
- Phase à vitesse constante

III - SITUATION SANS ASSERVISSEMENT



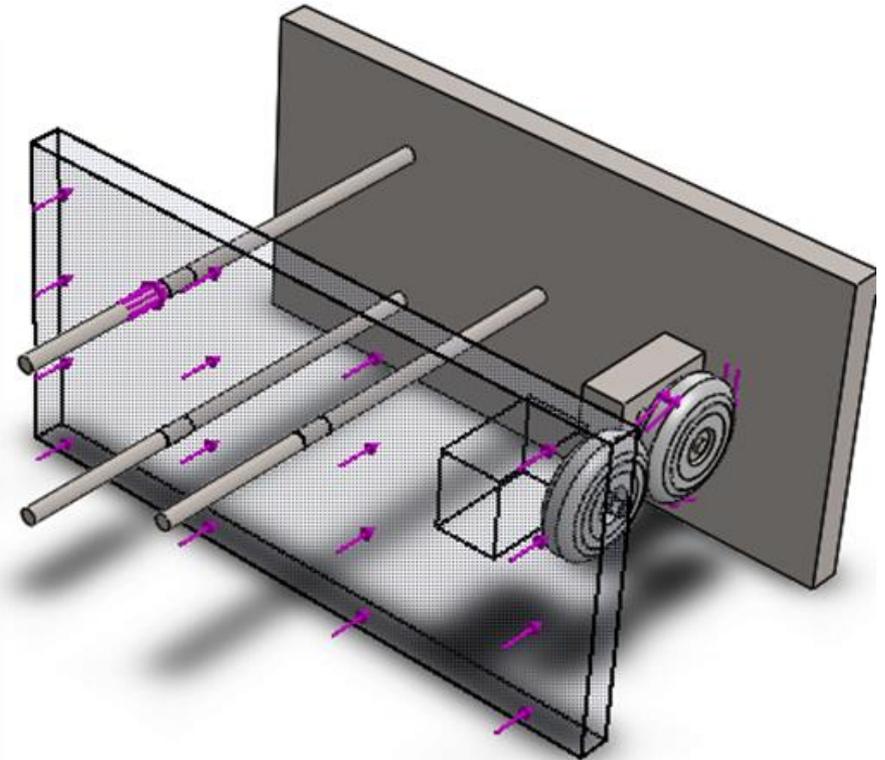
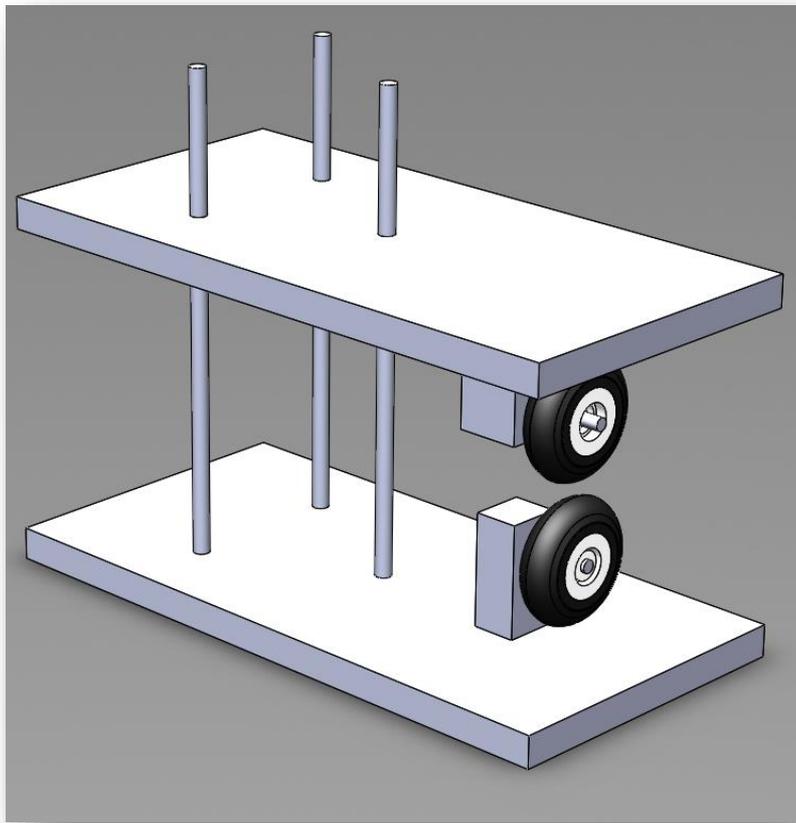
SAMMY AODIA

III - SITUATION SANS ASSERVISSEMENT



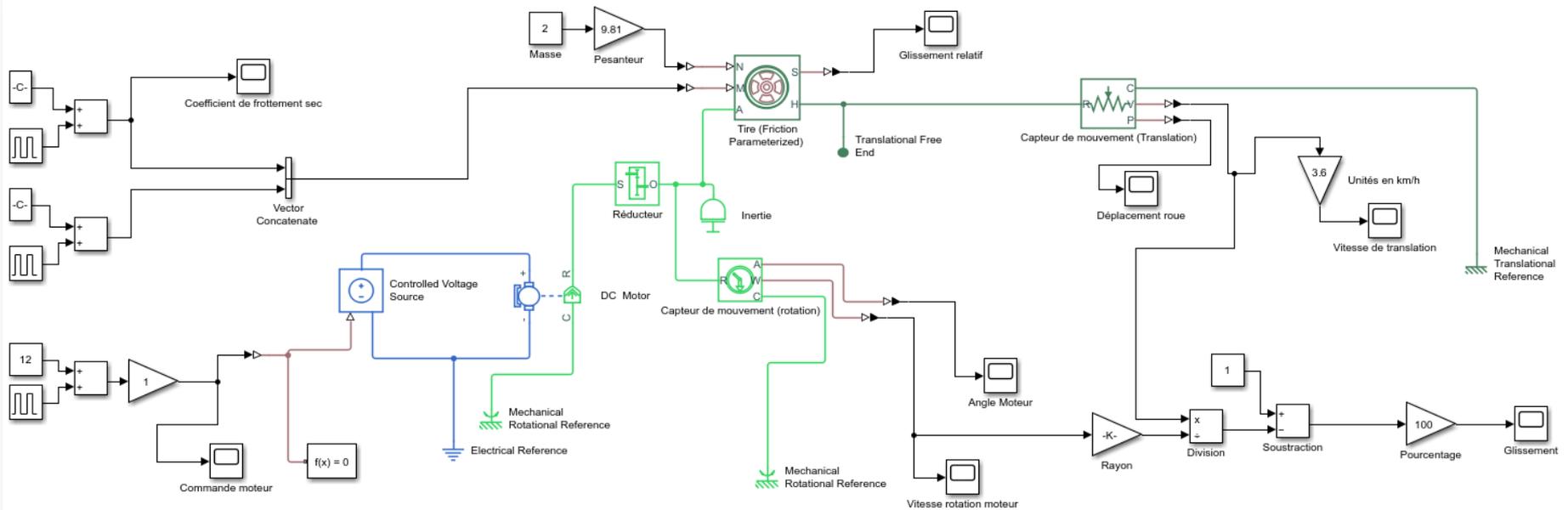
III -
MODÉLISATION
INFORMATIQUE –
1^{ER} ESSAI :
SOLIDWORKS

III - MODÉLISATION INFORMATIQUE – I^{ER} ESSAI : SOLIDWORKS

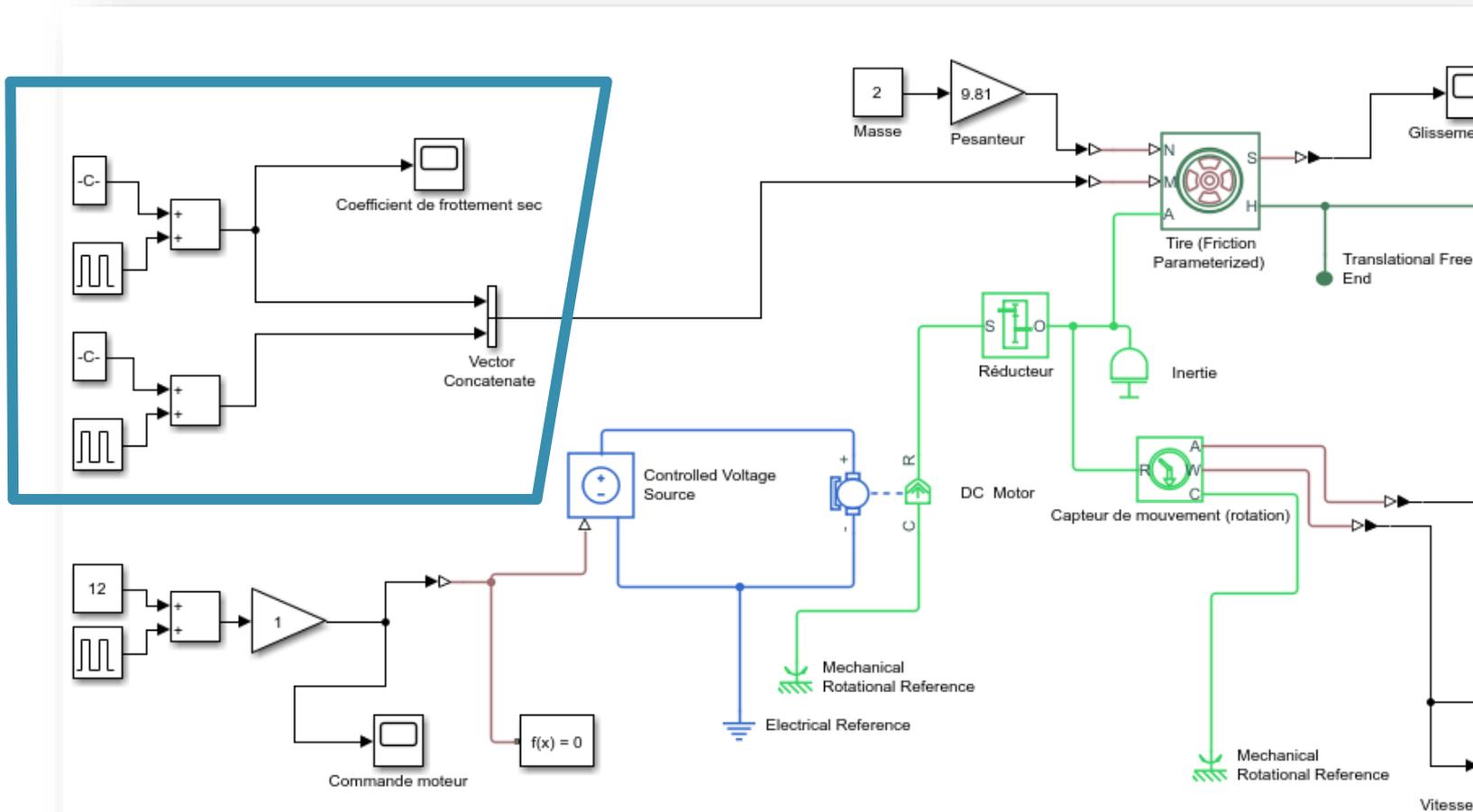


III -
MODÉLISATION
INFORMATIQUE
– 2ND ESSAI :
MATLAB

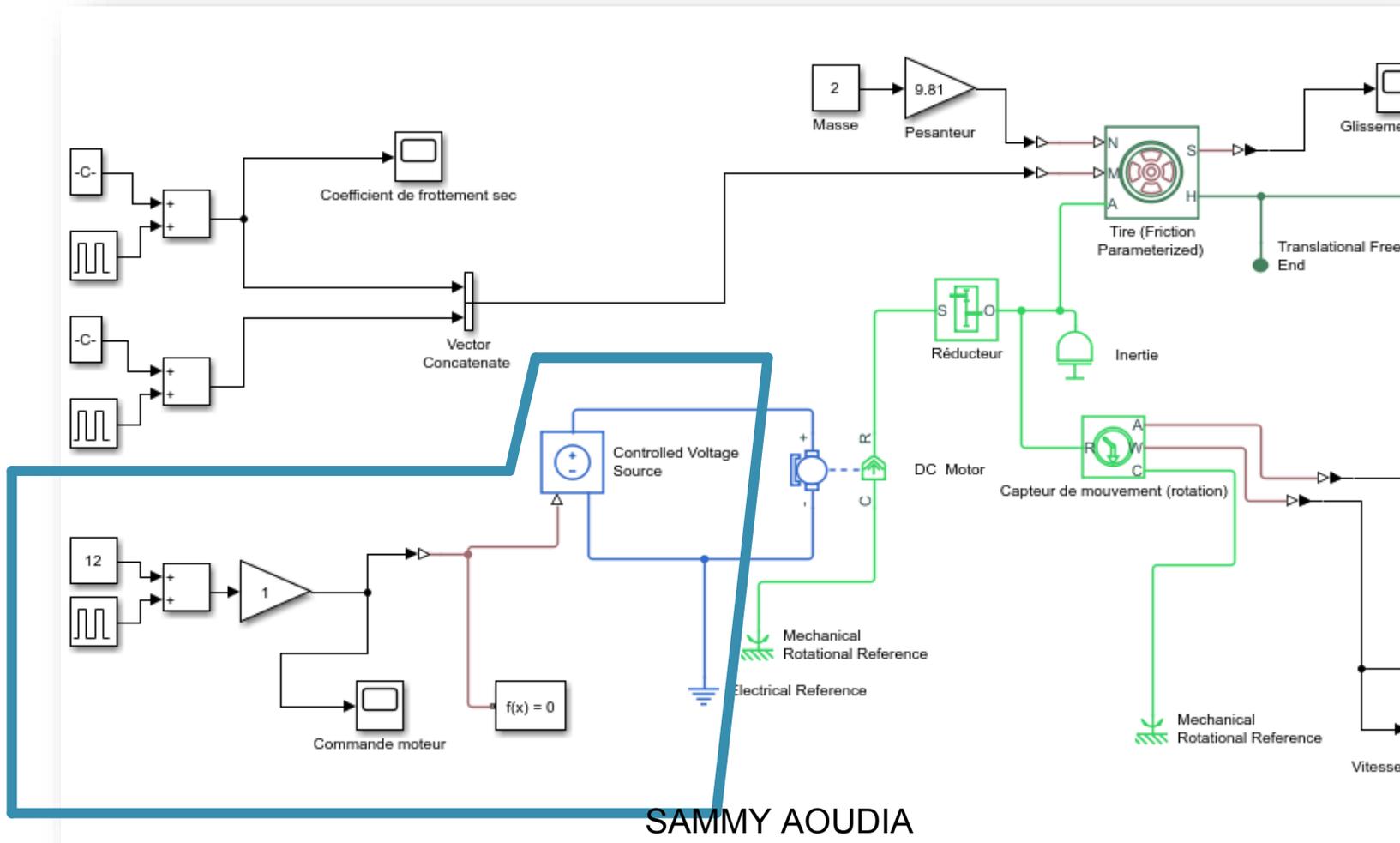
III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB



III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB

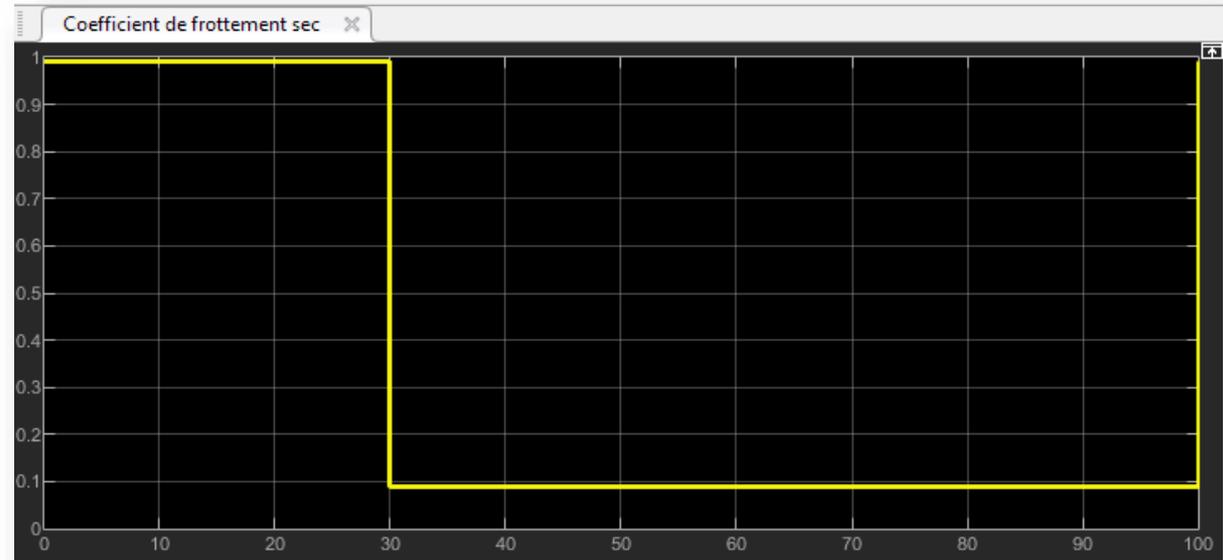


III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB



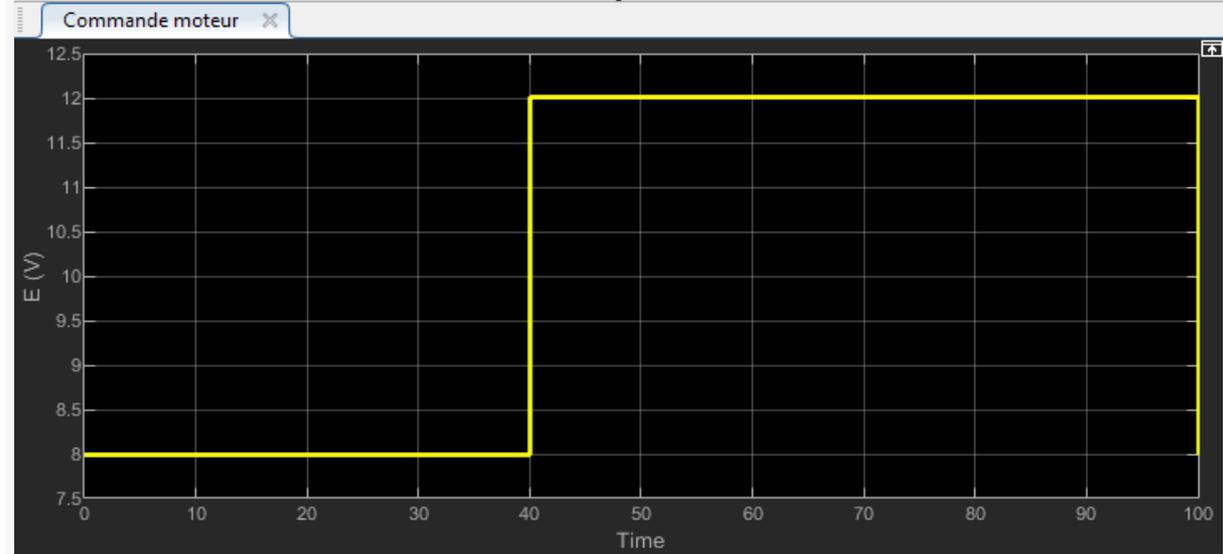
III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB

- Coefficient de frottement
- Commande Moteur

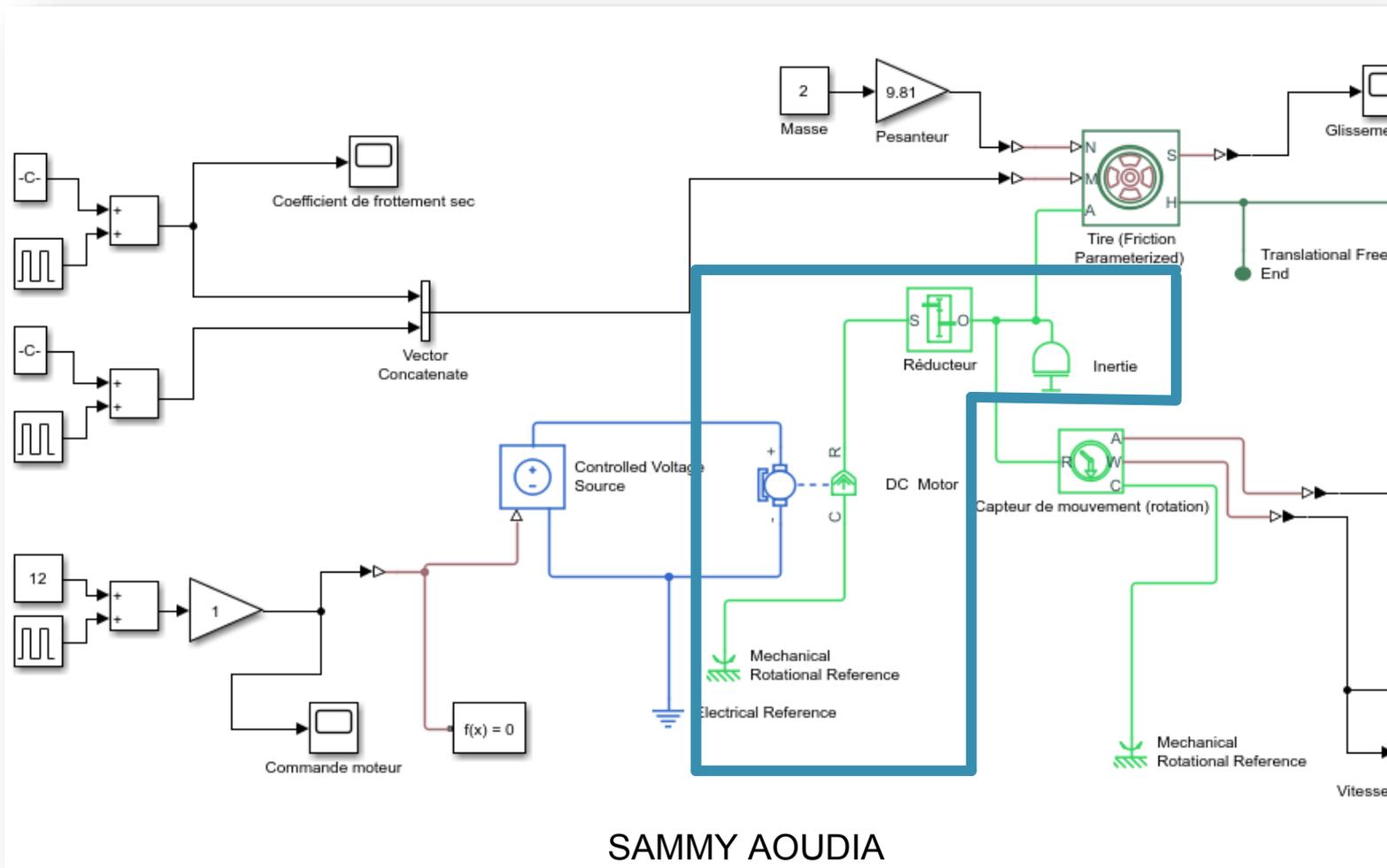


Ready

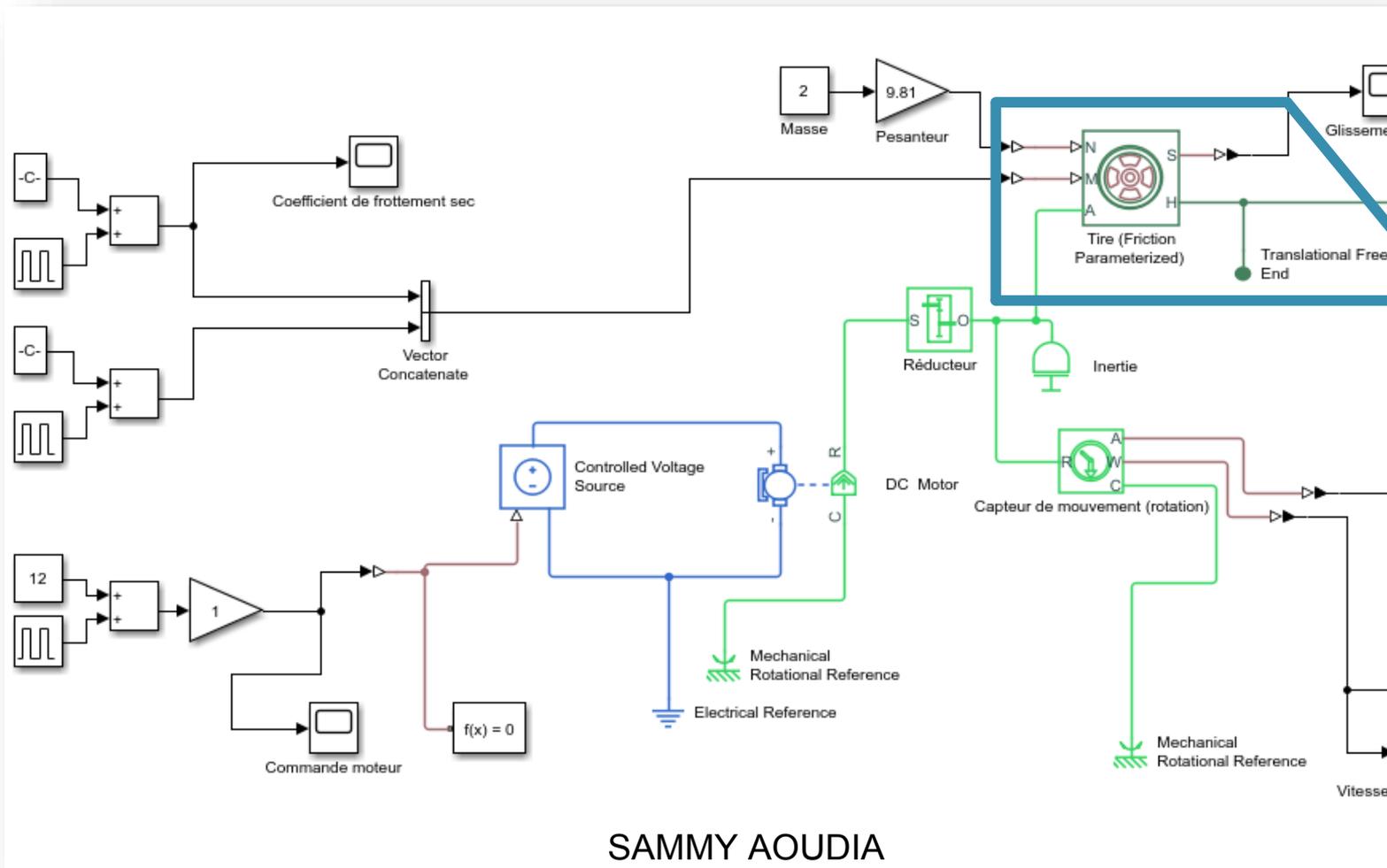
Sample based T=100.000



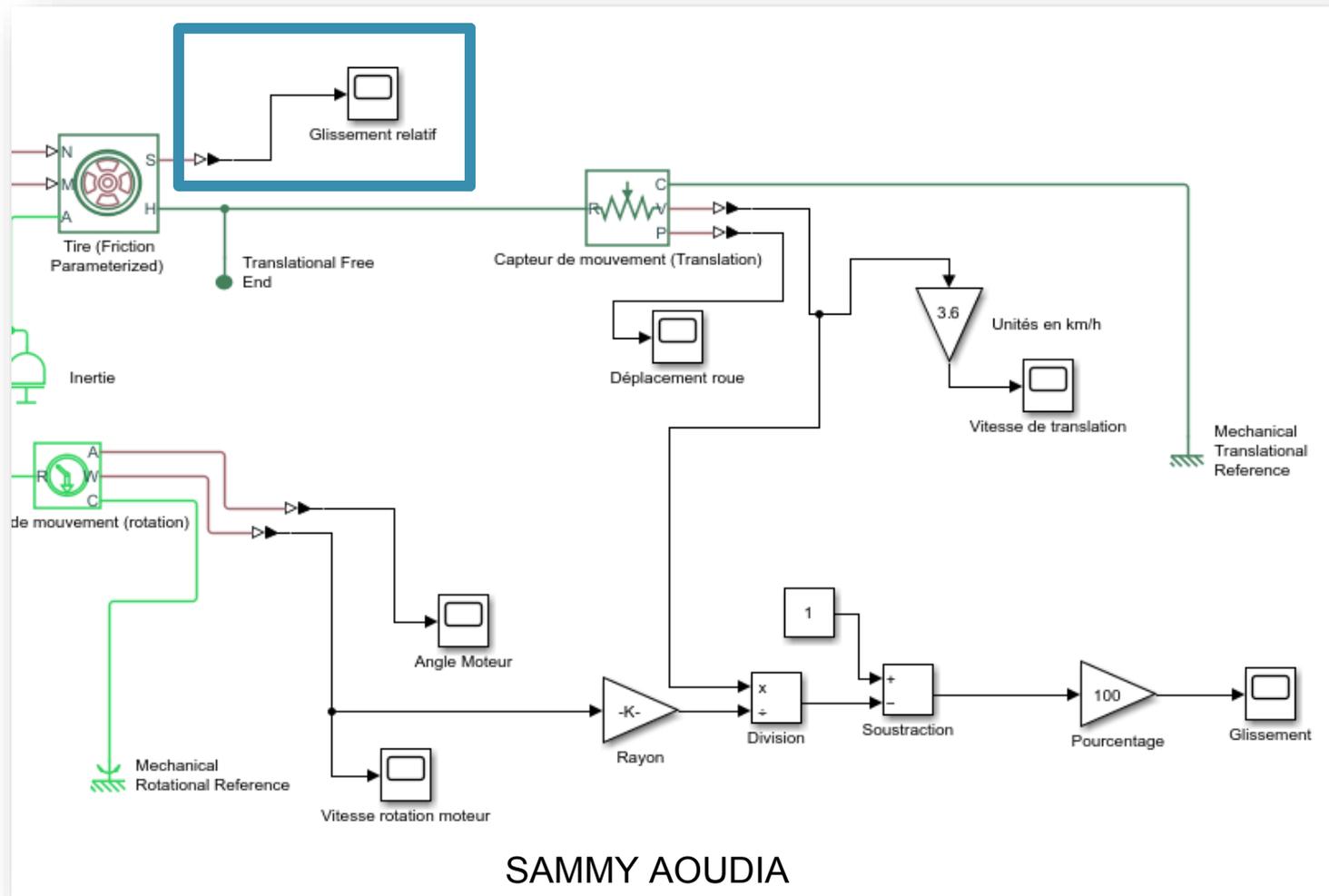
III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB



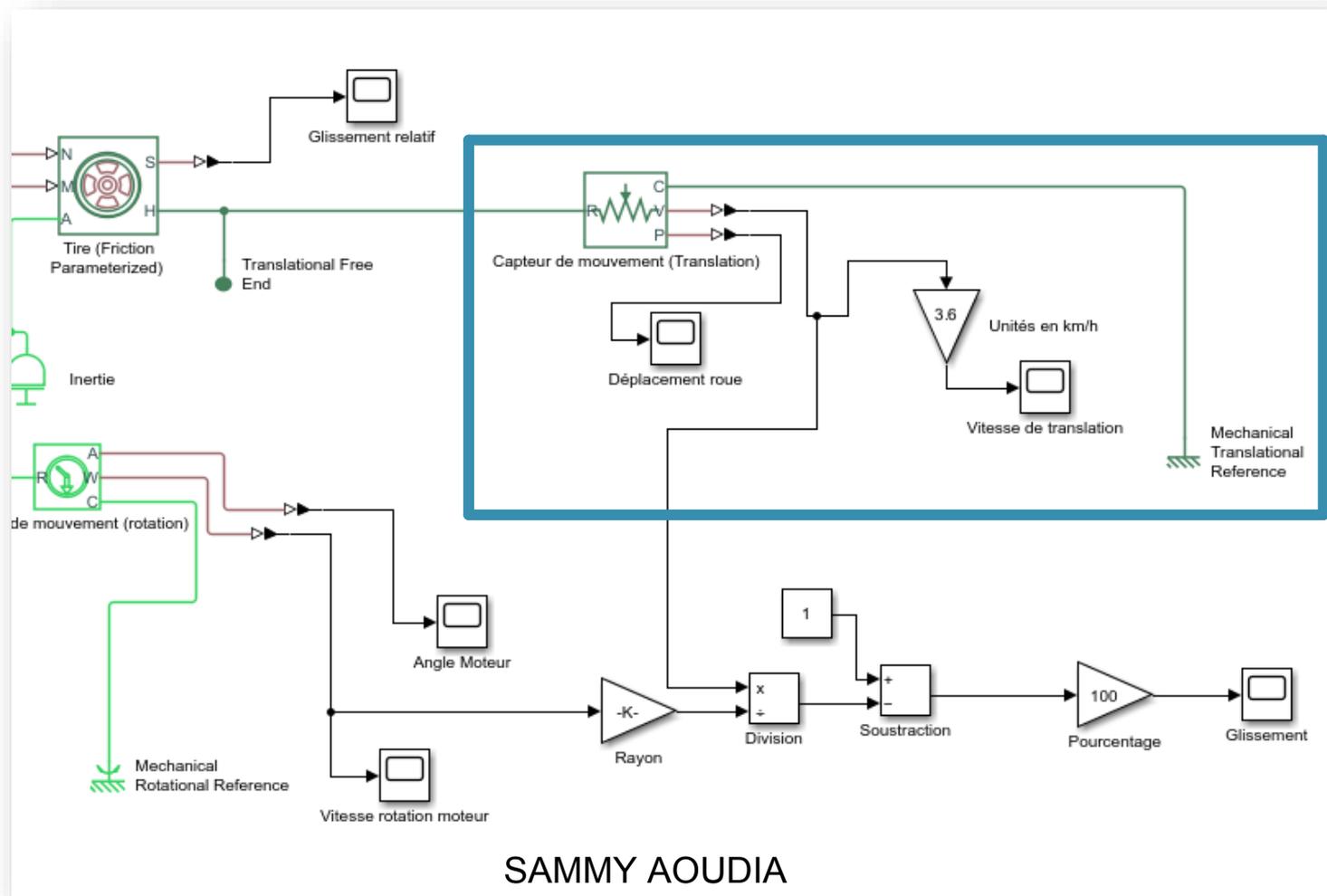
III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB



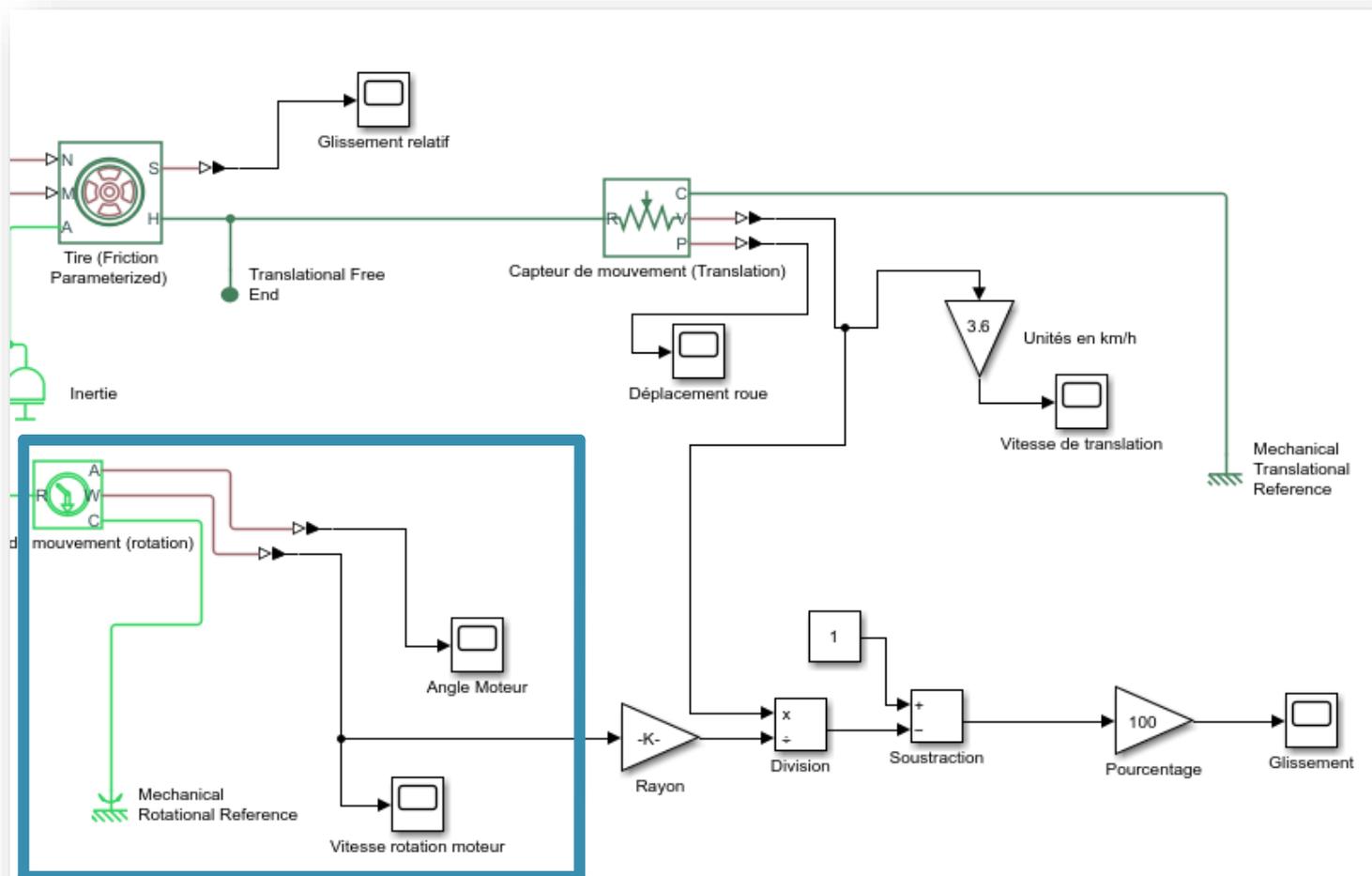
III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB



III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB

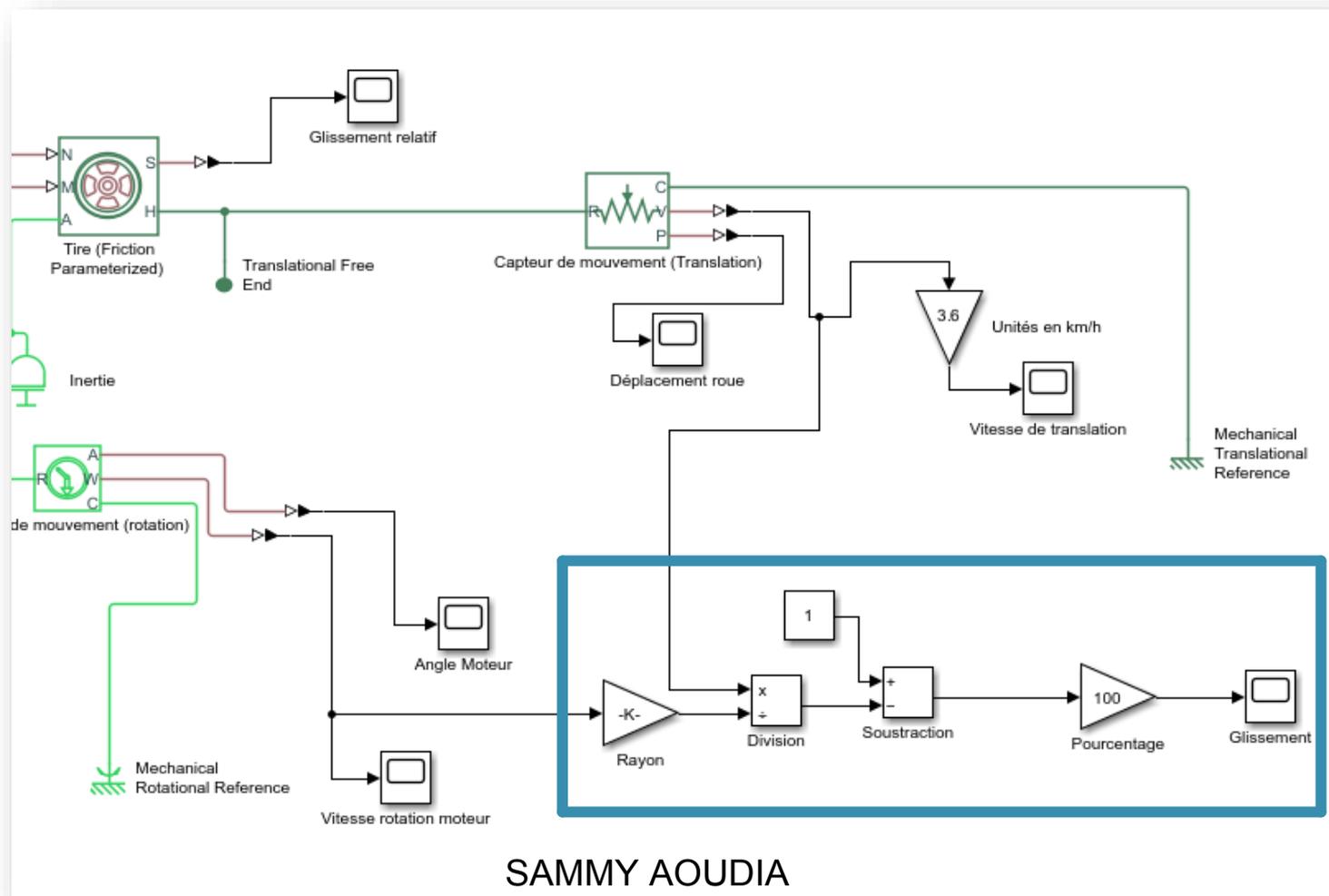


III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB



SAMMY AOUDIA

III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB



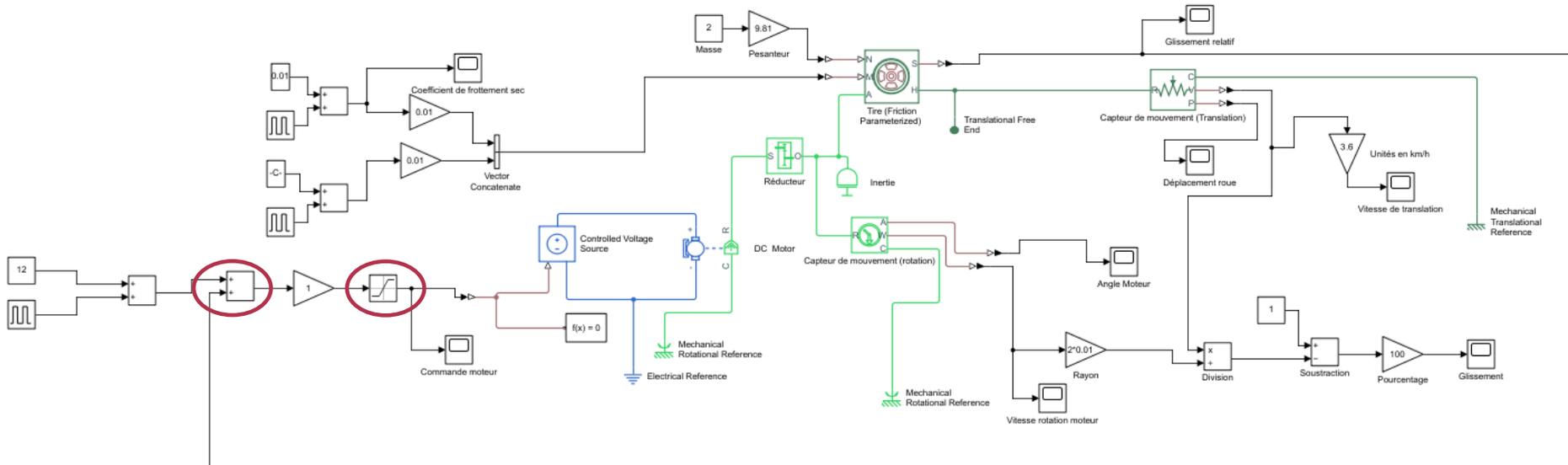
III - MODÉLISATION INFORMATIQUE – 2ND ESSAI : MATLAB

- Glissement (%)



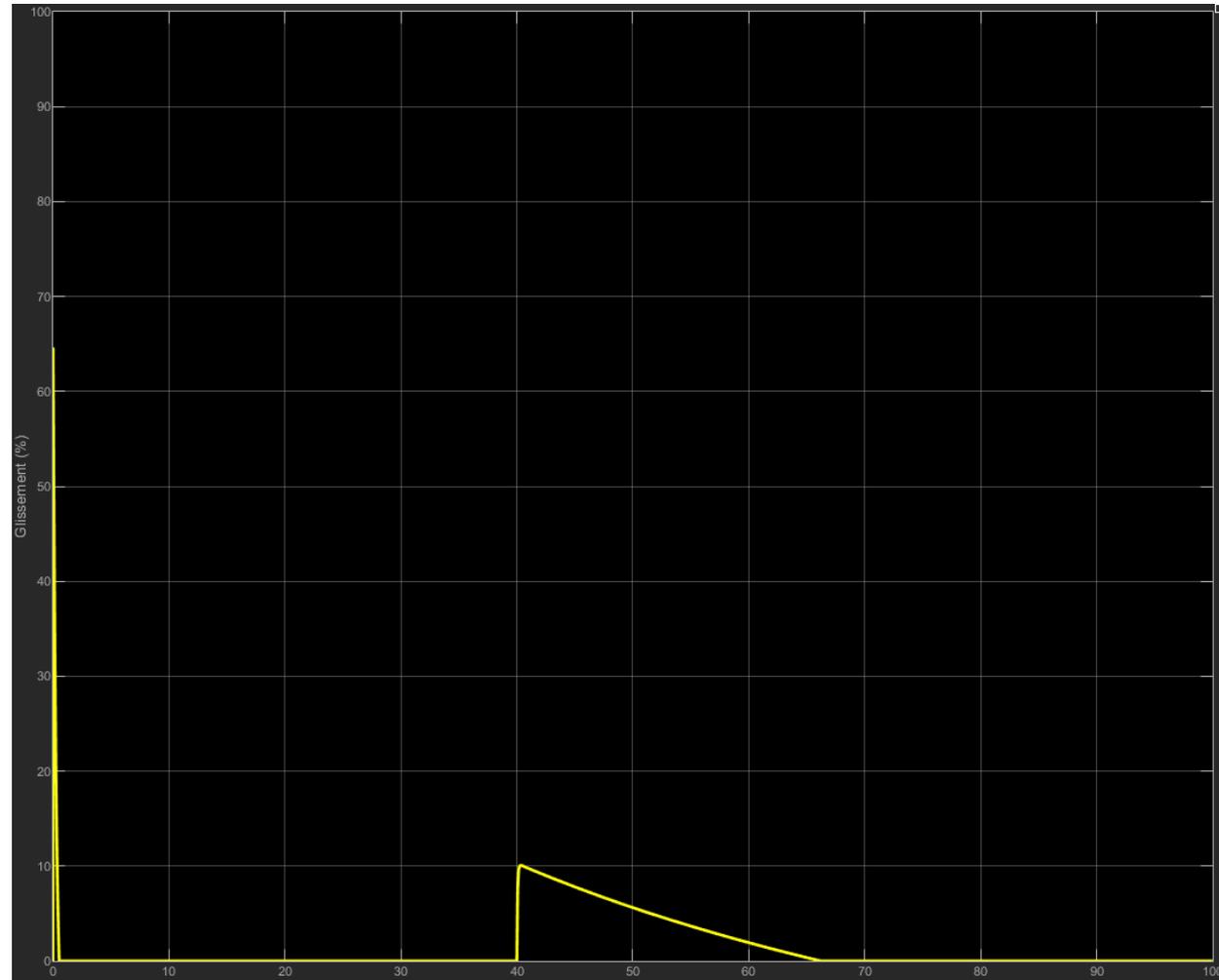
III -
MODÉLISATION
INFORMATIQUE –
3^{ÈME} ESSAI : MATLAB
(AVEC
ASSERVISSEMENT)

III - MODÉLISATION INFORMATIQUE – 3^{ÈME} ESSAI : MATLAB (AVEC ASSERVISSEMENT)



III - MODÉLISATION INFORMATIQUE – 3^{ÈME} ESSAI : MATLAB (AVEC ASSERVISSEMENT)

- Glissement (%)
- Réduction de 30% du glissement
- Pas de correcteur



III - MODÉLISATION INFORMATIQUE – 3^{ÈME} ESSAI : MATLAB (AVEC ASSERVISSEMENT)

- Glissement (%)
- Forte réduction du glissement
- Correcteur proportionnel

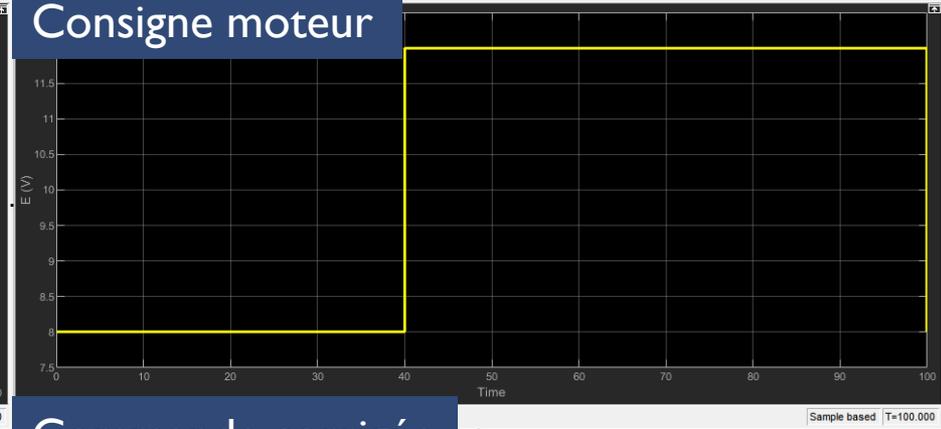


III - MODÉLISATION INFORMATIQUE – 3^{ÈME} ESSAI : MATLAB (AVEC ASSERVISSEMENT)

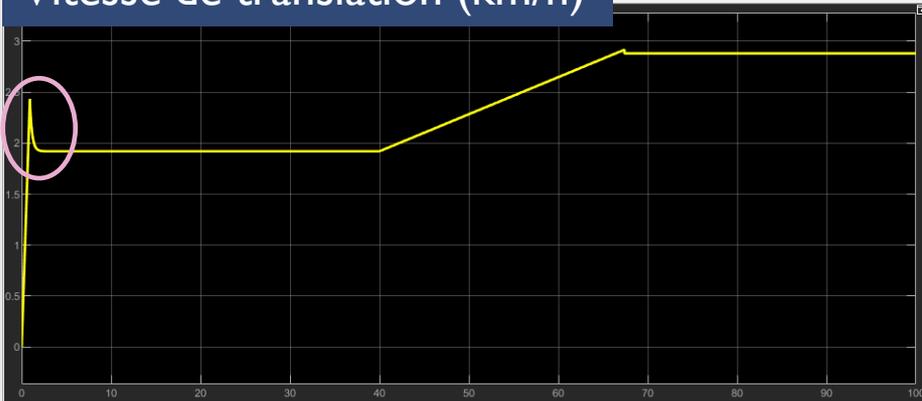
Glissement



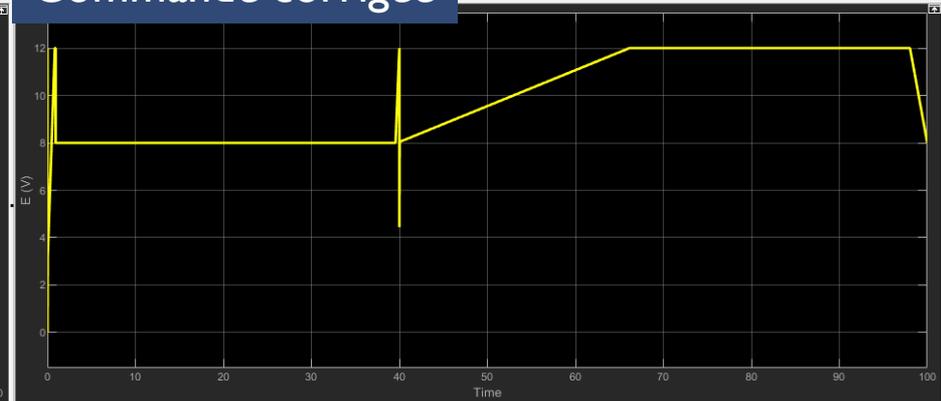
Consigne moteur



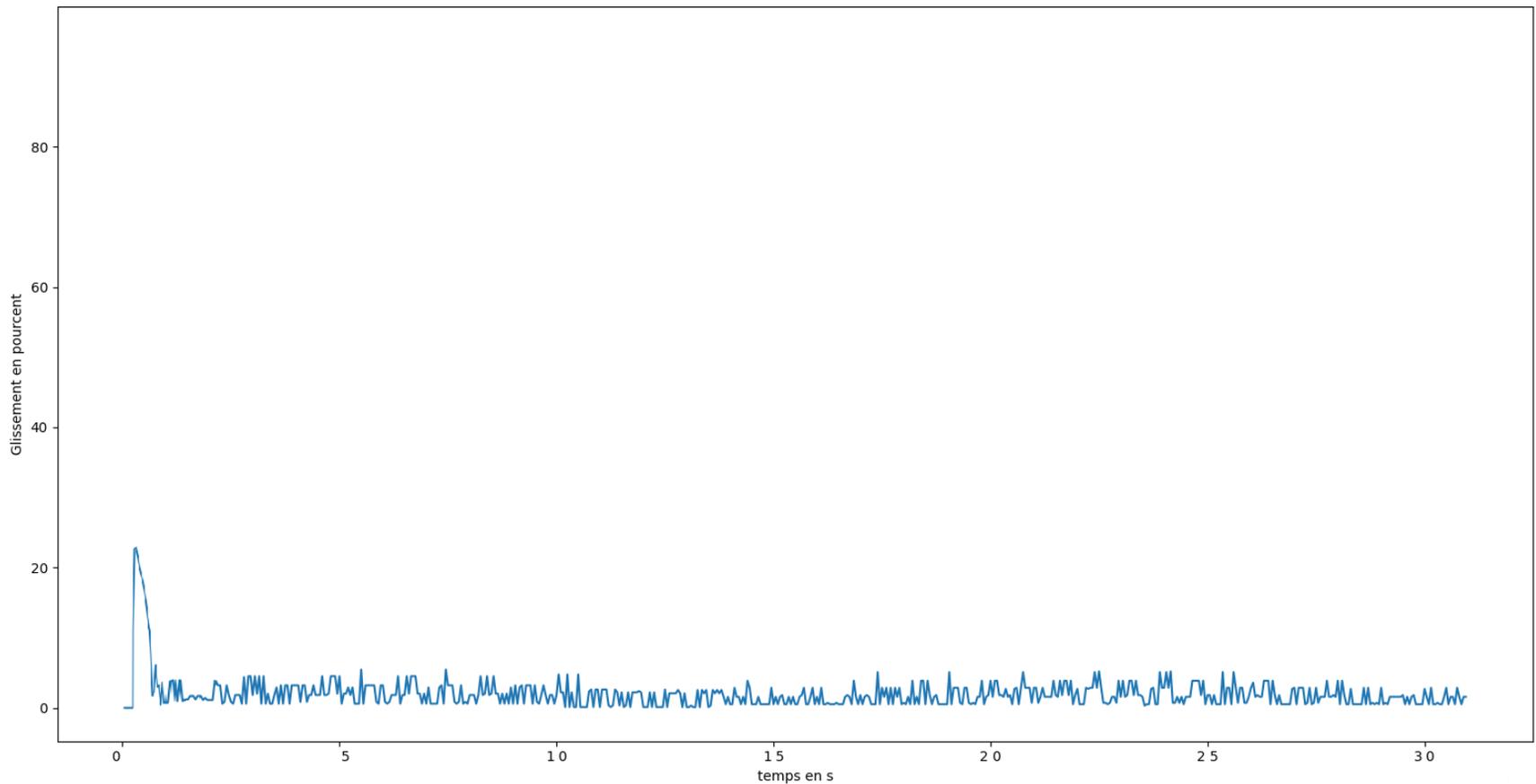
Vitesse de translation (km/h)



Commande corrigée



IV – APRÈS L'AJOUT DU CORRECTEUR



SYNTHÈSE

- Asservissement respecte les objectifs recherchés
- Initialisation nécessaire
- Dépassement de la vitesse de consigne (compromis nécessaire)

ANNEXE CODE ARDUINO

```
#include <Encoder.h> //Bibliothèque chargée

Encoder codeurMot(18, 19);
Encoder codeurStatique(20, 21);
const int motorPin1 = 9;
const int motorPin2 = 10;

long dt = 5;
long Pos=0;
long Pos2=0;
long t=0.0;
int speed =0;

void setup() {
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  analogWrite(motorPin1, speed);
  Serial.begin(9600);
  codeurMot.write(0); //on initialise le codeur à 0
  codeurStatique.write(0);
  Serial.print("t;Pos(stat);Pos(mot);consigne");
  Serial.print("\n");
}
```

```
void loop() {
  Pos=codeurMot.read(); //on lit la valeur du codeur
  Pos2=codeurStatique.read();
  t+=dt;
  Serial.print(t);
  Serial.print(";");
  Serial.print(Pos);
  Serial.print(";");
  Serial.print(Pos2);
  Serial.print(";");
  Serial.print(speed);
  Serial.print("\n");
  delay(dt);
}
```

ANNEXE : CODE PYTHON

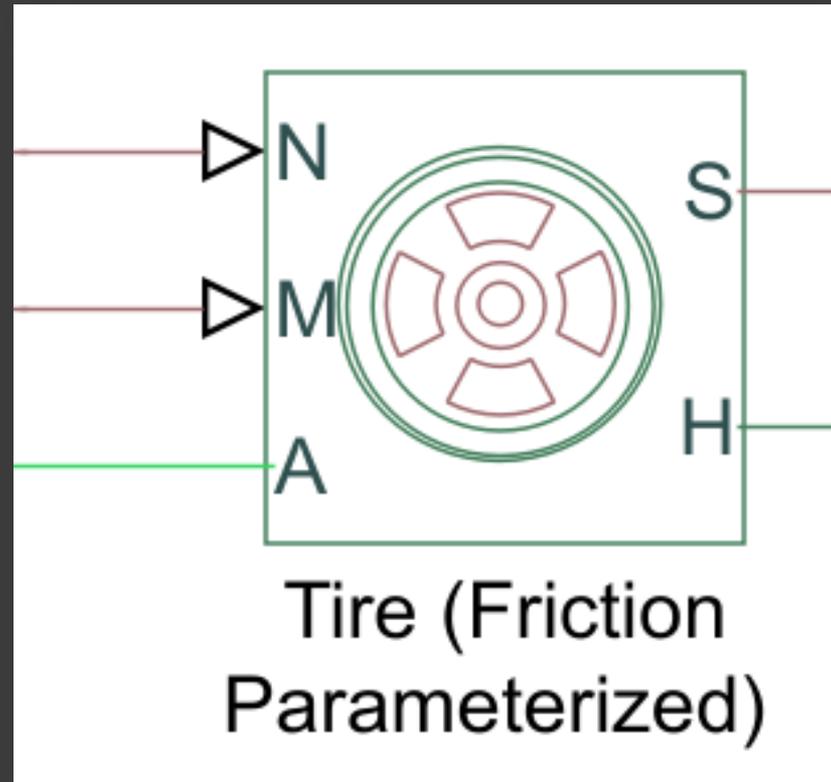
```
1 from numpy import *
2 from matplotlib.pyplot import *
3 import csv
4
5 data=[]
6 incrmax=1440
7 t,vs,vm,cg=[],[],[],[]
8 r=0
9 c=0
10 dt=5*10**-3
11 Rm=3.406*10**-2
12 Rs=2.25131*10**-2
13
14 name= input("Nom_de_l_essai_?")
15
16 with open('C:/Users/samma/OneDrive/Tipe 2021_2022/Sam/'+name+'.txt', newline='') as f:
17     reader = csv.reader(f, delimiter=';')
18     for row in reader:
19         data.append(row)
20 for i in range(1,len(data)-1):
21     for j in range(len(data[0])):
22         data[i][j]=float(data[i][j])
23
24 n=len(data)
25
26 def vitesses(data):
27     data[0].append("wstat")
28     data[0].append("wmot")
29     wstat,wmot=0.0,0.0
30     for i in range(1,len(data)-2):
31         data[i].append(wstat)
32         data[i].append(wmot)
33         wstat=(data[i][1]-data[i+1][1])/incrmax*dt
34         wmot=(data[i][2]-data[i+1][2])/incrmax*dt
35     data[len(data)-1].append(wstat)
36     data[len(data)-1].append(wmot)
37     return data
38
39 data=vitesse(data)
```

ANNEXE : CODE PYTHON

```
39 data=vitesses(data)
40
41 for i in range(1,len(data)-2):
42     t.append(data[i][0]*10**-3)
43     vs.append(data[i][4])
44     vm.append(data[i][5])
45
46
47 for i in range(len(vs)):
48     if vm[i]!=0:
49         cg.append(abs((( vs[i]*Rs-vm[i]*Rm) / (vs[i]*Rs) ))*100))
50     else:
51         cg.append(0.0)
52
53
54 for i in range(2*len(t)//3,len(t)-1):
55     c+=1
56     r+=vs[i]/vm[i]
57 r=r/c
58 print(r)
59
60
61 vs=array(vs)
62 t=array(t)
63 vm=array(vm)
64 plot(t,vs)
65 plot(t,vm)
66 ylabel('Vitesses en m/s')
67 xlabel('temps en s')
68 savefig('Vitesse_en_fct_du_tps'+name+'.png')
69 plot(t,cg)
70 plt.savefig('Condition_glissement'+name+'.png')
```

ANNEXE MATLAB : PNEU (PARAMÉTRÉ PAR LES COEFFICIENTS DE FROTTEMENT)

- Modélise le contact avec le sol d'un pneu.
- Paramètres d'entrée :
 - Coefficients de frottement
 - Force normale appliquée au pneu
- Renvoie le glissement relatif du pneu



ANNEXE MATLAB : MOTEUR À COURANT CONTINU

- Modélise un moteur à courant continu
- Paramètres :
 - Constante de couple
 - Résistance
 - Inertie

