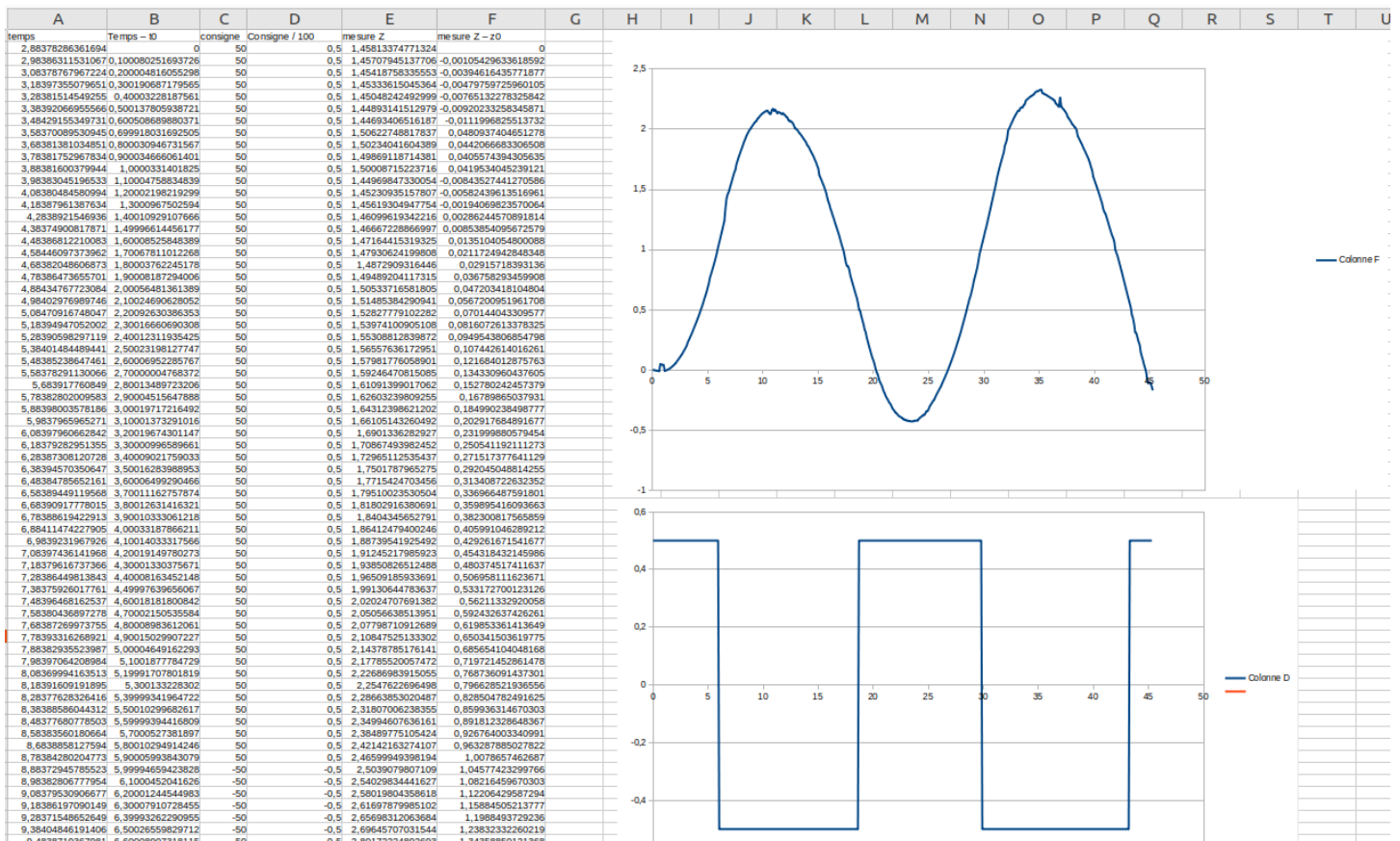


# Étape 1 : acquérir temps, consigne, mesure dans un fichier csv (fonction open et write de python)

## Exemple :

```
f = open("/home/ajuton/Documents/a0_Saphire/Dirigeable/asservissement/dates.txt", "w")
start = time.time()
while time.time() - start < 1.0:
    now = datetime.now().strftime("%Y-%m-%d %H:%M:%S.%f")[:-3]
    f.write(now + "\n")
    f.flush()
    time.sleep(0.01)
```

Étape 2 : tracer sous libreoffice la réponse (attention, il faut remplacer les points par des virgules), supprimer les points inutiles, remettre les valeurs initiales à 0, compléter (par recopie de la valeur précédente, les points abhérants).



Étape 3 : Vérifier que le fichier csv n'a pas de caractères étranges (ouvrir avec un éditeur de texte). Identifier le système dans Matlab.

Exemple avec sys\_tf (Emile a utilisé SystemIdentification depuis la console matlab, c'est plus user friendly) :

```
>> data = readmatrix('/home/ajuton/Documents/a0_Saphire/Dirigeable/asservissement/data_essai_indiciel_3.csv');
>> t = data(:,1);
>> consigne = data(:,2);
>> z = data(:,3);
>> data2 = iddata(z, consigne, 0.1)

data2 =

Time domain data set with 454 samples.
Sample time: 0.1 seconds

Outputs      Unit (if specified)
   y1

Inputs      Unit (if specified)
   u1

>> sys_tf = tfest(data2, 3, 2);
>> sys_tf

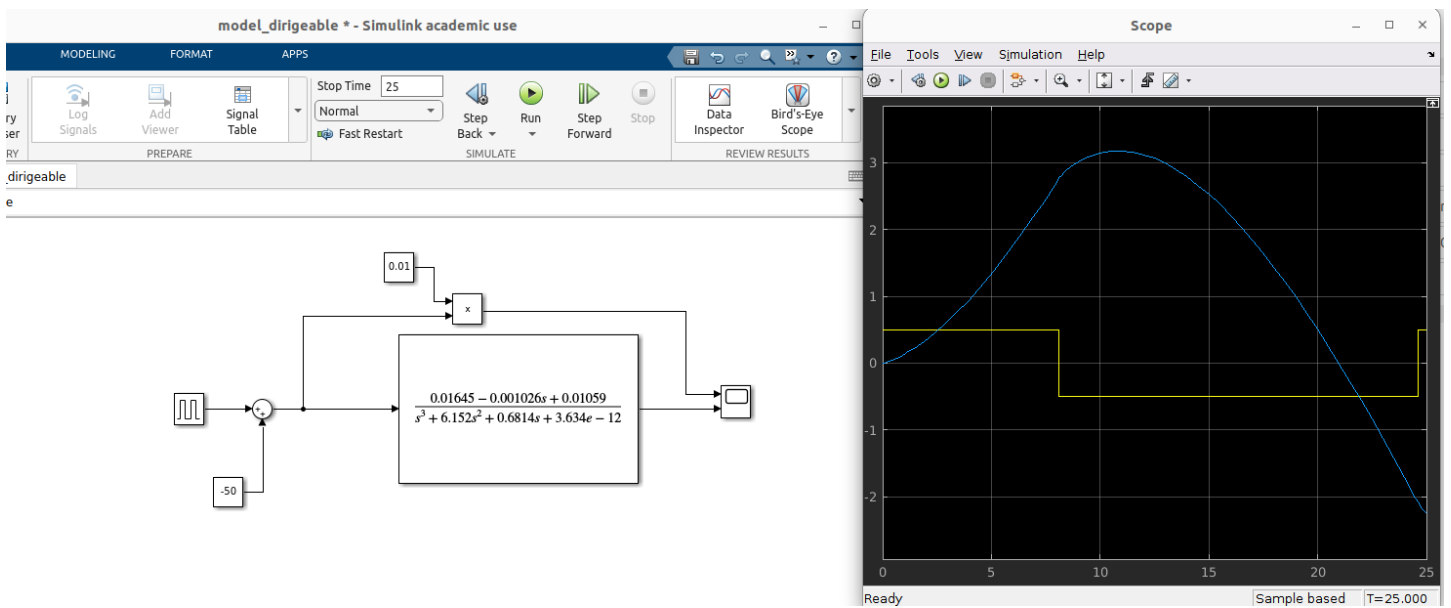
sys_tf =
From input "u1" to output "y1":
    0.01645 s^2 - 0.001026 s + 0.01059
-----
    s^3 + 6.152 s^2 + 0.6814 s + 3.634e-12

Continuous-time identified transfer function.

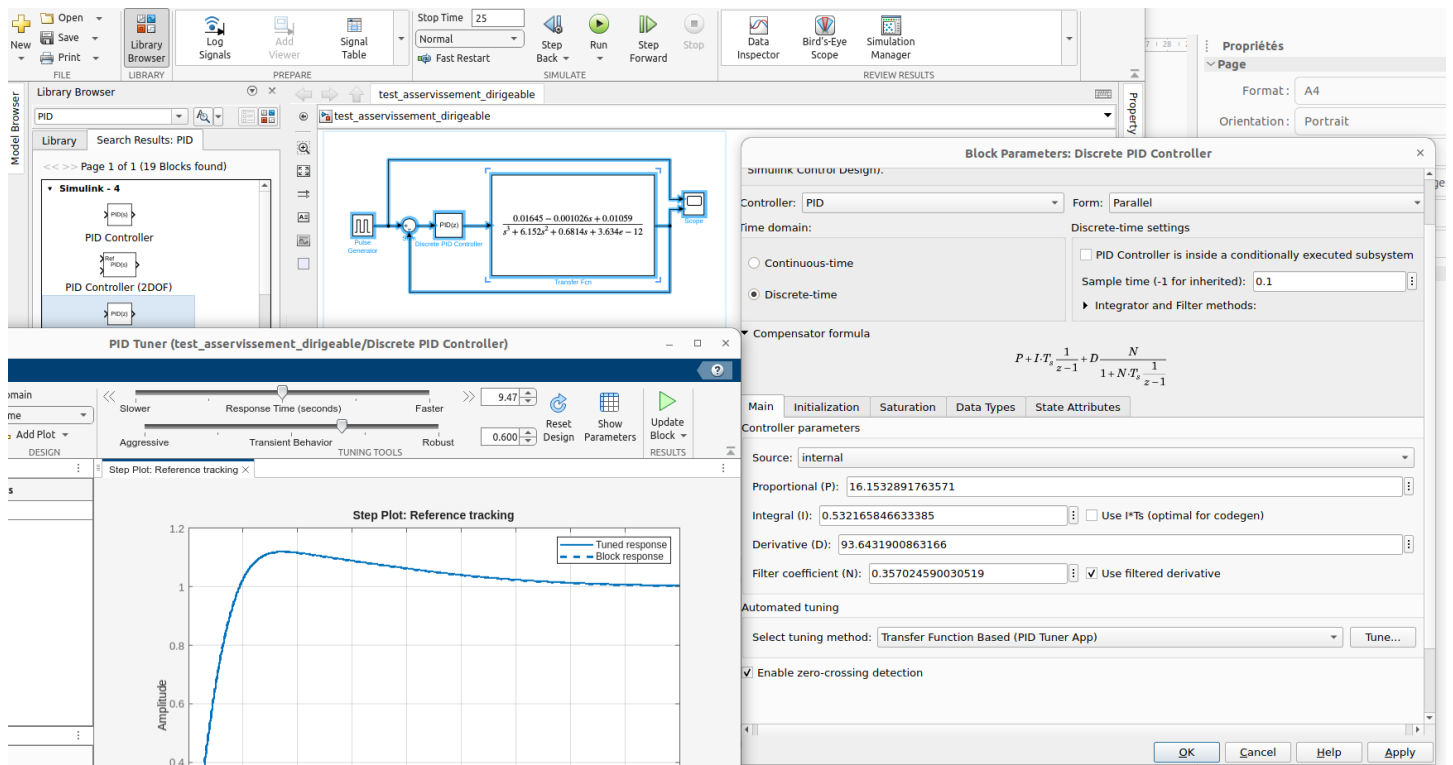
Parameterization:
    Number of poles: 3    Number of zeros: 2
    Number of free coefficients: 6
    Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "data2".
Fit to estimation data: 49.14%
FPE: 0.2219, MSE: 0.2133
```

Étape 4 : Dans Simulink, tester le système obtenu lors de l'identification, vérifier que c'est correct.



Étape 5 : Ajouter un contrôleur PID, mettre la bonne période d'échantillonnage et utiliser la fonction Tune de Matlab pour avoir des valeurs.



## Étape 6 : Intégrer le correcteur dans le dirigeable

The discrete-time PID controller with a filtered derivative term can be expressed in the **difference equation** form:

$$u(k) = a_0 e(k) + a_1 e(k-1) + a_2 e(k-2) - b_1 u(k-1) - b_2 u(k-2)$$

$$a_0 = P + DN$$

$$a_1 = -2(P + DN) + PNT_s + IT_s$$

$$a_2 = (P + DN) - PNT_s - IT_s + IT_s^2 N$$

$$b_0 = 1$$

$$b_1 = NT_s - 2$$

$$b_2 = 1 - NT_s$$