TD – Servocommande d'hélicoptère

POINT METHODE :

- Détermination graphique des coefficients d'un premier ordre (**Q5/Q8/Q11**) :
  $K = \frac{s_\infty}{e_0}$
  $\tau \rightarrow t_{5\%} \simeq 3.\tau$
  ou
  $\tau \rightarrow Abscisse\ du\ point\ d'intersection\ entre\ la\ tangent\ \text{à}\ l'origine\ et\ l'asymptote\ s_\infty$

ELEMENTS DE CORRECTION :

Q1 :

$z_e > 0 \rightarrow z_s > 0 \rightarrow sortie$

$z_e < 0 \rightarrow z_s < 0 \rightarrow rentrée$

Q2 :

Grandeurs = z_e et z_S

Perturbations = fuites de pression / frottement lors du déplacement / effort extérieur sur la tige.

Q3 :

$$f: y \rightarrow \frac{-y + 1{,}1}{0{,}01}$$

Q4 :

```python
def f(y,t): # Définition de la fonction f
    return (-y+1.1)/0.01  # A COMPLETER

def Euler(f, y0, Listet): # Méthode d'Euler
    t0 = Listet[0]
    y = y0
    Listey = [y]
    for k in range(1,len(Listet)):
        t1 = Listet[k]  # A COMPLETER
        y = y + (t1-t0)*f(y,t0) # A COMPLETER
        Listey.append(y)    # A COMPLETER
        t0 = t1 # A COMPLETER
    return Listey

(a,b) = (0,0.05) # Définition de l'intervalle d'étude
n = 100 # Nombre de points
y0 = 0 # Condition initiale

Listet = [a+i*(b-a)/n for i in range(n+1)] # Liste des temps
Listey = Euler(f, y0, Listet) # Liste des positions

plt.plot(Listet,Listey,'b-',linewidth=6,label="Euler Eq Diff 1er Ordre")
plt.legend(loc=2,fontsize=30)
plt.xlim(a,b)
plt.ylabel('Position relative Zs/Ze',fontsize=30)
plt.xlabel("Temps (s)",fontsize=30)
plt.title("Réponse temporelle de la position relative Zs/Ze",fontsize=30)
plt.grid(which="both")
plt.tick_params(axis = 'both', labelsize = 30)
plt.show()
```

Q5 :

Ecart statique non nul (10 %) → ~~OK~~

Stabilité → OK

Rapidité ($t_{5\%} = 0{,}03 > 10^{-2}\ s$ → ~~OK~~

Amortissement → ~~dépassement~~ → OK

Q6 :

$$F: Y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \rightarrow \begin{pmatrix} y_2 \\ \dfrac{-10^{-5}.y_2 - y_1 + 1}{3.10^{-3}} \end{pmatrix}$$

Q7 :

```
def F(Y,t): # Définition de la fonction F
    [y1,y2]=Y                                                   # A COMPLETER
    return np.array([y2,(-(10**-5)*y2-y1+1)/(3*10**-3)])        # A COMPLETER

def Euler(F, Y0, Listet): # Méthode d'Euler (attention, on travaille ici sur des vecteurs!)
    t0 = Listet[0]
    Y = Y0
    ListeY = [Y]
    for k in range(1,len(Listet)):
        t1 = Listet[k]           # A COMPLETER
        Y = Y + (t1-t0)*F(Y,t0) # A COMPLETER
        ListeY.append(Y)         # A COMPLETER
        t0 = t1                  # A COMPLETER
    return np.array(ListeY)

(a,b) = (0,0.04) # Définition de l'intervalle d'étude
n = 100 # Nombre de points
Y0 = [0,0] # Conditions initiales

Listet = [a+i*(b-a)/n for i in range(n+1)] # Liste des temps
Listey = [Euler(F, Y0, Listet)[i][0] for i in range(n+1)] # Liste des positions : 0 --> y et 1--> y'

plt.plot(Listet,Listey,'b-',linewidth=6,label="Euler Eq Diff 2nd Ordre")
plt.legend(loc=2,fontsize=30)
plt.xlim(a,b)
plt.ylabel('Position relative Zs/Ze',fontsize=30)
plt.xlabel("Temps (s)",fontsize=30)
plt.title("Réponse temporelle de la position relative Zs/Ze",fontsize=30)
plt.grid(which="both")
plt.tick_params(axis = 'both', labelsize = 30)
plt.show()
```

Q8 :

Ecart statique nul → OK

Stabilité → OK

Rapidité ($t_{5\%} = 0{,}02 > 10^{-2}\ s$ → ~~OK~~

Amortissement → dépassement > 5% → ~~OK~~

Q9 :

$$\searrow Osicllations \rightarrow \begin{cases} \searrow amortissement \\ \searrow t_{5\%} \end{cases}$$

Q10 :

$\varepsilon(p) = U_c(p) - U_{capt}(p) = Z_c(p).K_c - Z_s(p).K_v$

Asservissement parfait → $\varepsilon(p) = 0 \quad et \quad Z_c(p) = Z_s(p)$

Donc $\boldsymbol{K_c = K_v}$

Q11 :

Ecart statique nul → OK

Stabilité → OK

Rapidité ($t_{5\%} = 0{,}009 < 10^{-2}\ s$ → OK

Amortissement → dépassement $D_1 \approx 1\ \% < 5\ \%$ → OK