Exercice 1: Simulation d'un MCC

Question 1: Traduire la fonction de transfert du correcteur en une équation différentielle temporelle

$$C(p) = \frac{U(p)}{\varepsilon_{u_c}(p)} = K_p + \frac{K_i}{p}$$

$$U(p) = K_p \varepsilon_{u_c}(p) + \frac{K_i}{p} \varepsilon_{u_c}(p)$$

$$\frac{U(p)}{\varepsilon_{u_c}(p)} = \frac{K_p p + K_i}{p}$$

$$pU(p) = K_p p \varepsilon_{u_c}(p) + K_i \varepsilon_{u_c}(p)$$

$$\frac{du(t)}{dt} = K_p \frac{d\varepsilon_{u_c}(t)}{dt} + K_i \varepsilon_{u_c}(t)$$

$$\frac{U(p)}{\varepsilon_{u_c}(p)} = K_p + \frac{K_i}{p}$$

$$U(p) = K_p \varepsilon_{u_c}(p) + K_i \frac{\varepsilon_{u_c}(p)}{p}$$

$$u(t) = K_p \varepsilon_{u_c}(t) + K_i \int_0^t \varepsilon_{u_c}(u) du$$

CIN : La primitive $\Sigma_{u_c}(t) = \int_0^t \varepsilon_{u_c}(u) du$ s'annule en 0

$$u(t) = K_p \varepsilon_{u_c}(t) + K_i \Sigma_{u_c}(t)$$

En dérivant l'équation :

$$\frac{du(t)}{dt} = K_p \frac{d\varepsilon_{u_c}(t)}{dt} + K_i \frac{d\Sigma_{u_c}(t)}{dt}$$

$$\frac{du(t)}{dt} = K_p \frac{d\varepsilon_{u_c}(t)}{dt} + K_i \varepsilon_{u_c}(t)$$

Il faut donc faire apparaitre des dérivées (multiplications par p) pour obtenir les équations différentielles attendues rapidement.

$$\frac{du(t)}{dt} = K_p \frac{d\varepsilon_{u_c}(t)}{dt} + K_i \varepsilon_{u_c}(t)$$

Question 2: Proposer le système de 3 équations différentielles du premier ordre permettant de simuler cet asservissement avec la méthode d'Euler

$$e(t) = U_c - K_{capt}\, \omega(t)$$

$$\frac{du(t)}{dt} = -K_p K_{capt} \frac{d\,\omega(t)}{dt} + K_i \left(U_c - K_{capt}\, \omega(t)\right)$$

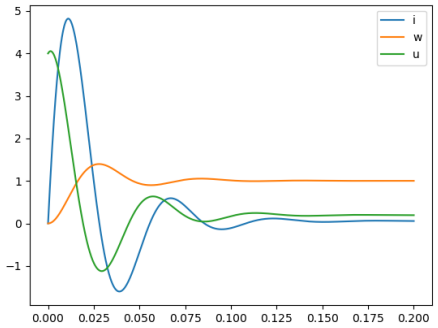
$$\begin{cases} \dfrac{di(t)}{dt} = \dfrac{u(t) - k_e \omega(t) - Ri(t)}{L} \\ \dfrac{d\omega(t)}{dt} = \dfrac{k_c i(t) - f\omega(t) - c_r(t)}{J} \\ \dfrac{du(t)}{dt} = -K_p K_{capt} \dfrac{d\,\omega(t)}{dt} + K_i \left(U_c - K_{capt}\, \omega(t)\right) \end{cases}$$

Question 3: Programmer la résolution Euler explicite des équations pour un échelon de vitesse de 1 rd/s sur 0,2 secondes avec 1000 points (on ne manipulera que des listes et « odeint » ne sera pas utilisé)

```
def Euler_Explicite(f,y0,t0,t1,dt):
    t = t0
    y = y0
    T = [t]
    Y = [y]
    i = 0
    while t < t1: # Important: < et non <=
        yp = f(y,t)
        y = [y[i] + yp[i]*dt for i in range(len(y))]
        i += 1
        t = t0 + i*dt # Important: Eviter les erreurs de t+=dt
        T.append(t)
        Y.append(y)
    return T,Y

def F(Y,t):
    R = 0.45
    L = 0.0046
    ke = 0.169
    kc = 0.17
    f = 0.01
    J = 0.01
    cr = 0
    i,w,u = Y
    di = (u-ke*w-R*i)/L
    dw = (kc*i-f*w-cr)/J
    du = -Kp*Kcapt*dw+Ki*(Uc-Kcapt*w)
    Sol = [di,dw,du]
    return Sol

wc = 1
Kcapt = 10
Ka = Kcapt
Kp = 0.4
Ki = 7.29
Uc = wc*Ka
U = Kp*Uc # CIN (d./dt et w nulles au départ)
i0 = 0
w0 = 0
U0 = U
V0 = [i0,w0,U0]
t0 = 0
t1 = 0.2
N = 1000

dt = (t1-t0)/(N-1)
Lt,Y = Euler_Explicite(F,V0,t0,t1,dt)
Li = [Y[i][0] for i in range(len(Y))]
Lw = [Y[i][1] for i in range(len(Y))]
Lu = [Y[i][2] for i in range(len(Y))]
```

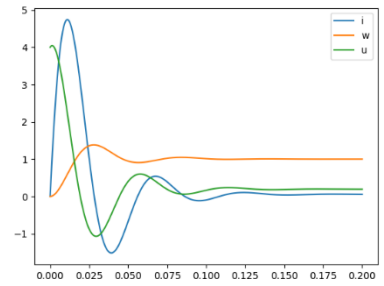
Question 4: Tracer l'évolution de la vitesse moteur et de son intensité en fonction du temps

```
import matplotlib.pyplot as plt
plt.close('all')
def f_Affiche(fig,Lx,Ly,Legende):
    plt.figure(fig)
    plt.plot(Lx,Ly,label=Legende)
    plt.legend()
    plt.show()
    plt.pause(0.0001)

f_Affiche(1,Lt,Li,'i')
f_Affiche(1,Lt,Lw,'w')
f_Affiche(1,Lt,Lu,'u')
```



Question 5: Réaliser la même résolution en utilisant « odeint »

```
dt = (t1-t0)/(N-1)
Lt = [t0+i*dt for i in range(N)]
from scipy.integrate import odeint
Sol = odeint(F,V0,Lt)
Li = [Sol[i][0] for i in range(len(Sol))]
Lw = [Sol[i][1] for i in range(len(Sol))]
Lu = [Sol[i][2] for i in range(len(Sol))]
f_Affiche(2,Lt,Li,'i')
f_Affiche(2,Lt,Lw,'w')
f_Affiche(2,Lt,Lu,'u')
```



Question 6: Réaliser le modèle XCOS de l'asservissement et comparer les résultats obtenus