ASN (S6)

Année 2020

Asservissements Numériques

Labo 4 : Simulation d'une Régulation de Température

In [1]:
```python
import numpy as np
import json
import plotly
import matplotlib.pyplot as plt
from control import tf, c2d, feedback, bode
from ENIB_control import step, impulse, nichols, rlocus, damp, pole, stepinfo, figure
```

1. But de l'étude

On se propose d'étudier la commande d'un four selon des contraintes de réponse en poursuite et en régulation. La synthèse d'un correcteur de type Proportionnel Intégral (PI) sera basée sur un modèle linéaire du système physique autour d'un point de fonctionnement ; dans le but de valider l'étude, une simulation sera effectuée dans l'environnement `jupyter notebook`.

2. Description du Système

La température $\theta(t)$ d'un four électrique est commandée en agissant sur la tension d'alimentation $V_1(t)$ d'une résistance chauffante R. La puissance fournie, $P(t) = V_1^2(t)/R$, provoque l'échauffement de la résistance et de l'air dans le four selon les lois de la thermodynamique (capacité calorifique, échanges par conduction, rayonnement et convection). La figure 1 représente le bloc-diagramme du dispositif constitué des éléments suivants :

Fig.1 - Système.

Amplificateur de puissance :

- Gain $A = 20$; dynamique négligeable (temps de réponse très faible).

Fonction de transfert de l'élément chauffant :

- Le gain K_1 (Watts/Volt) est obtenu en linéarisant l'expression de $P(t)$ au voisinage du point de fonctionnement $V_1 = 110$ V ; par ailleurs $R = 10\Omega$ (une linéarisation correspond à une dérivation autour du point de fonctionnement).
- La constante de temps $\tau_1 = 10$ s.

Fonction de transfert de l'enceinte thermique (le four) :

- Gain K_2 (Degrés/Watt) = $6.10^{-2}{}^\circ/W$.
- Constante de temps $\tau_2 = 60$s.

Sonde de température :

- Sonde de type thermocouple associée à une électronique de traitement analogique : échelle 0-10 V pour $\theta(t)$ variant de 0 à 300^oC ; temps de réponse négligeable.

Système avec en chaine direct:

- un correcteur de fonction de transfert $C(p)$
- un système de fonction de transfert $F(p)$

Système est à retour unitaire.

La fonction de transfert en boucle fermée :

$$H(p) = \frac{C(p)F(p)}{1 + C(p)F(p)}$$

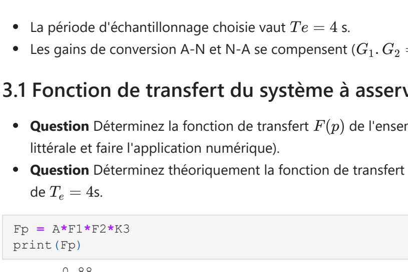
Système précis $\rightarrow$ Gain statique en boucle fermé doit être égal à 1 c-a-d $H(0) = 1$.

$$H(0) = \frac{C(0)F(0)}{1 + C(0)F(0)} = 1$$

Pour obtenir un système précis en boucle fermé, il faut que $C(0) \rightarrow \infty$.

In [2]:
```python
A = 20
R = 10
tau1 = 10
K2 = 0.06
tau2 = 60
K3 = 10/300

V1= np.arange(0,200)
P = V1**2/R

# approximation au voisinage de V1
V1_app = 110
K1 = 2*V1_app/R
tau1 = 10
b = (V1_app**2)/R-K1*V1_app
print(b)
P2 = K1*V1+b

plt.plot(V1,V1**2/R,label="P(t)")
plt.plot(V1,P2,label="linearisation");
plt.xlabel("V1")
plt.ylabel("P(V1)")
plt.legend()
F1 = tf([K1],[tau1,1])
F2 = tf([K2],[tau2,1])

print('A={}'.format(A))
print('F1: {}'.format(F1))
print('F2: {}'.format(F2))
print('K3: {}'.format(K3))
```

```
-1210.0
A=20
F1:
   22
--------
10 s + 1

F2:
  0.06
--------
60 s + 1

K3: 0.03333333333333333
```



3. Préparation

On veut commander ce système à l'aide d'un correcteur PI conformément au diagramme de la figure 2 :

Fig.2 - Système en boucle fermée.

- La période d'échantillonnage choisie vaut $Te = 4$ s.
- Les gains de conversion A-N et N-A se compensent ($G_1.G_2 = 1$).

3.1 Fonction de transfert du système à asservir

- **Question** Déterminez la fonction de transfert $F(p)$ de l'ensemble à asservir (amplificateur-résistance- four-sonde). (Ecrire l'expression littérale et faire l'application numérique).
- **Question** Déterminez théoriquement la fonction de transfert discrète équivalente $F(z)$, avec BOZ, pour une période d'échantillonnage de $T_e = 4$s.

In [3]:
```python
Fp = A*F1*F2*K3
print(Fp)
```

```
       0.88
------------------
600 s^2 + 70 s + 1
```

In [4]:
```python
Fz = c2d(Fp,Ts=4)
print(Fz)
```

```
  0.01008 z + 0.00863
----------------------
z^2 - 1.606 z + 0.6271

dt = 4
```

3.2 Définition du cahier des charges : $H_m(z)$

On cherche à obtenir pour le système corrigé et bouclé $H(z)$ une fonction de transfert équivalente à un modèle du 2nd ordre appelé $H_m(z)$. On impose le cahier des charges suivant.

Fonction de transfert modèle $H_m(z)$:

- Erreur statique nulle (gain statique $H_m(z)$ égal à 1)
- Dépassement relatif de $D = 15\%$ sur la réponse indicielle.

- **Question** Déterminez le facteur d'amortissement m et le facteur de résonance M en dB pour ce modèle $H_m(z)$.

In [5]:
```python
m = 0.52
M = 1
```

3.3 Expressions des fonctions de transfert sortie, commande et erreur

Pour visualiser les signaux de sortie $s[n]$, de commande $u_c[n]$ et d'erreur $\epsilon[n]$ pour un signal d'entrée donné $e[n]$, il est nécessaire de déterminez les fonctions de transfert suivantes :

$$H(z) = \frac{S(z)}{E(z)} \qquad (1)$$

$$cmd(z) = \frac{U_c(z)}{E(z)} \qquad (2)$$

$$err(z) = \frac{\epsilon(z)}{E(z)} \qquad (3)$$

- **Question** Déterminez en fonction de $C(z)$ et $F(z)$ (pas d'applications numériques)
  - l'expression de la fonction de transfert en boucle fermée : $H(z)$,
  - l'expression du rapport $err(z)$ où $\epsilon(z)$ désigne signal d'erreur,
  - l'expression du rapport $cmd(z)$ où $U_c(z)$ désigne le signal de commande.

$$H(z) = \frac{S(z)}{E(z)} = \frac{C(z)F(z)}{1 + C(z)F(z)} \qquad (4)$$

$$cmd(z) = \frac{U_c(z)}{E(z)} = \frac{U_c(z)}{S(z)} \times \frac{S(z)}{E(z)} = \frac{H(z)}{F(z)} \qquad (5)$$

$$err(z) = \frac{\epsilon(z)}{E(z)} = \frac{E(z) - S(z)}{E(z)} = 1 - H(z) \qquad (6)$$

4. Analyse d'un correcteur Proportionnel Intégral

Le correcteur série $C(z)$ sur la figure 2 doit apporter à $F(z)$ tout ce qui manque pour que la fonction de transfert corrigée $H(z)$ respecte le cahier des charges.

Le correcteur proportionnel intégral est composé d'un gain et d'une intégration :

- Action proportionnelle : apporte du gain sur toute la bande fréquentielle.
- Action intégrale : apporte du gain sur les basses fréquences, donc de la précision sur la boucle fermée.

Ces deux actions se retrouvent dans un correcteur PI analogique :

$$C(p) = K_i\left(1 + \frac{1}{T_i p}\right)$$

- K_i : action proportionnelle,
- T_i : constante d'intégration,
- $\omega_c = \frac{1}{T_i}$: pulsation de coupure.

La structure d'un correcteur PI numérique est identique à la structure du correcteur PI analogique. En discrétisant on obtient :

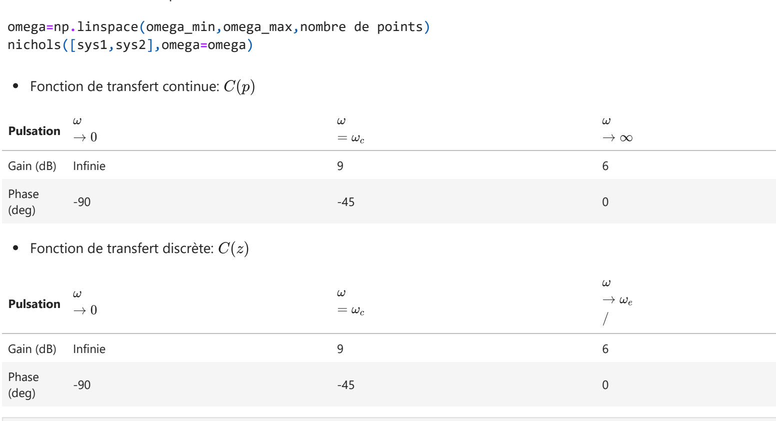
$$C(z) = K_i\left(1 + \frac{T_e}{T_i}\frac{z}{z-1}\right)$$

Remarque Dans cette expression l'intégration continue est remplacée par une intégration numérique (cf cours chap. 7): $\frac{1}{p} \rightarrow \frac{T_e z}{z-1}$.

- **Question** Vérifiez l'équivalence des deux fonctions de transfert en traçant leur diagramme de Bode (on prendra : $K_i = 2$, $T_i = 50$s et $T_e = 4$s).(commande : `[mag, phase, omega] = bode(sys)`)

In [6]:
```python
Ki = 2
Ti = 50
Te = 4

Cp = tf([Ki*Ti,Ki],[Ti,0])
print(c2d(Cp,Te))
Cz=Ki*(1+tf([Te/Ti,0],[1,-1],Te))
print(Cz)

w = np.logspace(-4,0,100)
fig = figure("bode")
fig.plot(Cp,w=w)
fig.plot(Cz,w=w)
fig.show()
```

```
2 z - 1.84
----------
  z - 1

dt = 4


2.16 z - 2
----------
  z - 1

dt = 4
```



- **Question** Complétez les tableau suivants : (à l'aide de la commande `nichols`). Vous pouvez définir votre plage d'étude en définissant un vecteur pulsations :

```python
omega=np.linspace(omega_min,omega_max,nombre de points)
nichols([sys1,sys2],omega=omega)
```

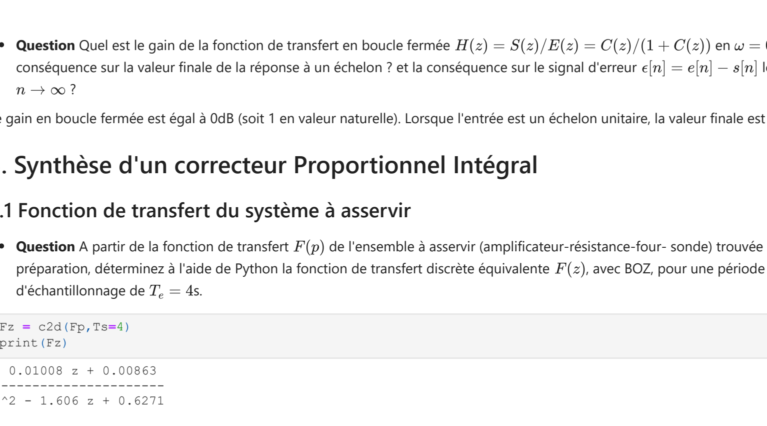
- Fonction de transfert continue: $C(p)$

Pulsation	$\omega \rightarrow 0$	$\omega = \omega_c$	$\omega \rightarrow \infty$
Gain (dB)	Infinie	9	6
Phase (deg)	-90	-45	0

- Fonction de transfert discrète: $C(z)$

Pulsation	$\omega \rightarrow 0$	$\omega = \omega_c$	$\omega \rightarrow \omega_e/2$
Gain (dB)	Infinie	9	6
Phase (deg)	-90	-45	0

In [7]:
```python
wc = 1/Ti
Fe = 1/Te
we=2*np.pi*Fe
w=np.logspace(-4,2,100)


fig = figure("nichols")
fig.plot(Cp,w=w)
fig.plot(Cz,w=w)
fig.show()
```



- **Question** Comment peut-on approximer la fonction $C(z)$ pour $\omega \rightarrow 0$ et $\omega = \omega_e/2$. Regardez comment se comporte le lieu de Nichols de $C(z)$.

- **Question** Quel est le gain de la fonction de transfert en boucle fermée $H(z) = S(z)/E(z) = C(z)/(1 + C(z))$ en $\omega = 0$. Quel est la conséquence sur la valeur finale de la réponse à un échelon ? et la conséquence sur le signal d'erreur $\epsilon[n] = e[n] - s[n]$ lorsque $n \rightarrow \infty$?

Le gain en boucle fermée est égal à 0dB (soit 1 en valeur naturelle). Lorsque l'entrée est un échelon unitaire, la valeur finale est égale à 1.

5. Synthèse d'un correcteur Proportionnel Intégral

5.1 Fonction de transfert du système à asservir

- **Question** A partir de la fonction de transfert $F(p)$ de l'ensemble à asservir (amplificateur-résistance-four- sonde) trouvée en préparation, déterminez à l'aide de Python la fonction de transfert discrète équivalente $F(z)$, avec BOZ, pour une période d'échantillonnage de $T_e = 4$s.

In [8]:
```python
Fz = c2d(Fp,Ts=4)
print(Fz)
```

```
  0.01008 z + 0.00863
----------------------
z^2 - 1.606 z + 0.6271

dt = 4
```

5.2 Synthèse du correcteur

Pour répondre au cahier des charges vu dans le 3.2 ($D\% = 15\%$ et erreur statique nulle), on développe un correcteur proportionnel intégral $C(z)$:

$$C(z) = K_i\left(1 + \frac{T_e}{T_i}\frac{z}{z-1}\right)$$

La synthèse de ce correcteur consiste à déterminer les paramètres T_i et K_i qui permettent d'obtenir une FTBF corrigée :

$$H(z) = \frac{C(z)F(z)}{1 + C(z)F(z)}$$

ayant les mêmes caractéristiques que le cahier des charges défini par $H_m(z)$.

Synthèse d'un correcteur PI On recherche les paramètres du correcteur par une technique graphique d'**approches successives**. On approche un ordre de grandeur de K_i et de T_i en opérant de la façon qui suit dans le plan de Black Nichols :

1. Tracez la FTBO du système non corrigé $F(z)$.
2. Translatez le lieu de transfert par l'apport d'un gain K_1 pour venir tangenter approximativement la courbe d'équigain en boucle fermée égale à la résonance M_{dB} désirée. Attention, tangentez directement la courbe d'équigain égale à M_{dB}. Notez le gain K_1 et la pulsation de résonance ω_r en boucle fermée.
   - La valeur du gain K_i en première approximation est alors donnée par K_1.
   - Pour ne pas modifier le point de résonance obtenu, on choisit la pulsation de cassure $1/T_i$ suffisamment loin de ω_r (pulsation de résonance au point de tangence) de telle manière à ne pas introduire un déphasage supplémentaire en ce point. On prend en général $1/Ti \ll \omega_r/5$.
   - Vous prendrez ici en première approximation $1/Ti = \omega_r/10$
   .
3. Saisissez sous Python votre correcteur $C(z)$ à l'aide de la commande :tf. L'équivalent numérique $C(z)$ du correcteur continu P.I. peut s'écrire avec K_1 et T_i:

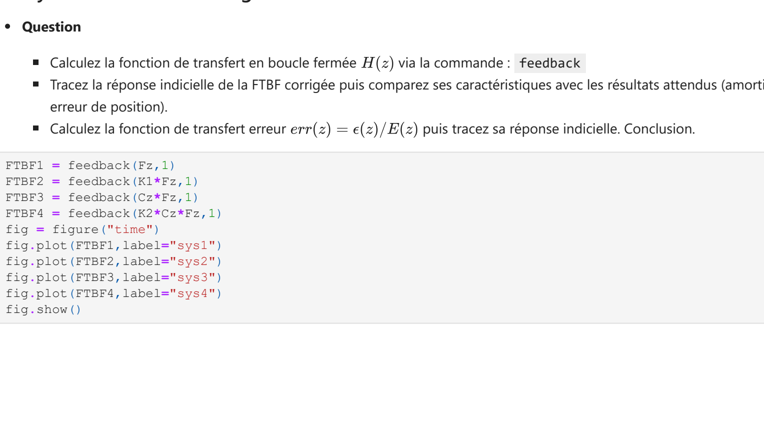
$$C(z) = K_i\left(1 + \frac{T_e}{T_i}\frac{z}{z-1}\right)$$

4. Tracez la FTBO corrigée $C(z).F(z)$ dans le plan de Black Nichols. Si le facteur de résonance a bougé, ajustez la FTBO corrigée $C(z)F(z)$ par un gain K_2 pour obtenir le facteur de résonance M souhaité. Le gain final du correcteur est alors donné par $K_i = K_1.K_2$.

- **Question** En suivant la procédure ci-dessus, déterminez les paramètres du correcteur ($K_i = K1.K2$) et T_i et son expression finale.

In [9]:
```python
w=np.logspace(-4,2,200)
wr = 0.078
K1 = 10**(17.5/20)
K2 = 0.85
Ti = 10/wr
print(K1)
print(K1*K2)
print(Ti)
Cz=K1*(1+tf([Te/Ti,0],[1,-1],Te))

fig = figure("nichols")
fig.plot(Fz,w=w)
#fig.plot(K1*Fz,w=w)
#fig.plot(Cz*Fz,w=w)
fig.plot(K2*Cz*Fz,w=w)
fig.grid(cm=np.array([0,1,3,6,-6]),show_phase=False)
fig.xlim([-250,0])
fig.show()
```

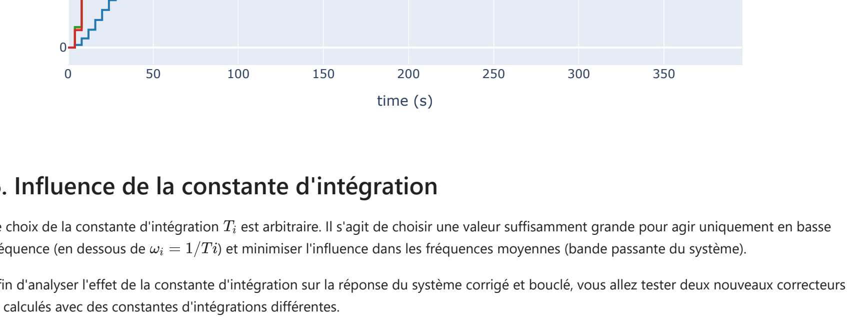
```
7.498942093324558
6.374100779325874
128.2051282051282
```



5.3 Système bouclé et corrigé - validation du correcteur

- **Question**
  - Calculez la fonction de transfert en boucle fermée $H(z)$ via la commande : `feedback`
  - Tracez la réponse indicielle de la FTBF corrigée puis comparez ses caractéristiques avec les résultats attendus (amortissement, erreur de position).
  - Calculez la fonction de transfert erreur $err(z) = \epsilon(z)/E(z)$ puis tracez sa réponse indicielle. Conclusion.

In [10]:
```python
FTBF1 = feedback(Fz,1)
FTBF2 = feedback(K1*Fz,1)
FTBF3 = feedback(Cz*Fz,1)
FTBF4 = feedback(K2*Cz*Fz,1)
fig = figure("time")
fig.plot(FTBF1,label="sys1")
fig.plot(FTBF2,label="sys2")
fig.plot(FTBF3,label="sys3")
fig.plot(FTBF4,label="sys4")
fig.show()
```



6. Influence de la constante d'intégration

Le choix de la constante d'intégration T_i est arbitraire. Il s'agit de choisir une valeur suffisamment grande pour agir uniquement en basse fréquence (en dessous de $\omega_i = 1/Ti$) et minimiser l'influence dans les fréquences moyennes (bande passante du système).

Afin d'analyser l'effet de la constante d'intégration sur la réponse du système corrigé et bouclé, vous allez tester deux nouveaux correcteurs PI calculés avec des constantes d'intégrations différentes.

La méthode utilisée est celle décrite dans le paragraphe 4.2.

- **Question**
  - Calculez les correcteurs PI $C_1(z)$ et $C_2(z)$ en prenant respectivement $1/T_{i1} = \omega_r/5$ et $1/T_{i2} = \omega_r/20$
  - Calculez les fonctions de transfert en boucle fermée et corrigée $H_1(z)$ et $H_2(z)$.
  - Tracez les réponses indicielles des trois fonctions de transfert $H(z)$ (§4.3), $H_1(z)$ et $H_2(z)$. Observez l'influence de la constante d'intégration sur la réponse à un échelon.
  - Proposez un choix de correcteur pour cette application.

In [ ]: