Asservissement (S6)

Année 2020 - 2021

Comportement des Systèmes Continus

But de l'étude.

La première partie de ce TP concerne l'analyse du comportement temporel et fréquentiel des systèmes continus. En plus d'un rappel sur les Fonctions de Transfert (FT) du 1er ordre et 2ème ordre, cette partie permet de prendre progressivement connaissance de l'environnement Jupyter et plus particulièrement de la librairie `Python Control` concernant les F.T.

Dans une deuxième partie, diverses synthèses de correcteurs analogiques sont réalisées. Le calcul de correcteurs et la validation des résultats se feront en utilisant au maximum les fonctionnalités de la librairie `Python Control`.

> La documentation de la librairie Python Control est disponible à l'adresse: documentation2.ipynb

In [1]:
```
import numpy as np
from ENIB_control import *
from control import tf
```

1. Analyse en temps et en fréquence des fonctions de transfert

La fonction `tf` de la librairie `Python Control` permet de créer une fonction de transfert $F(p)$. Il s'agit d'une classe particulière de variables avec ses propres propriétés et ses propres opérations.

Pour nous entrainer, nous allons considérer les fonctions de transfert suivantes:

$$F_1(p) = \frac{2}{1+3p}$$

$$F_2(p) = \frac{2}{1+10p+2p^2}$$

$$F_3(p) = \frac{2}{1+0.5p+8p^2}$$

Saisie du Système

La commande `tf` permet la saisie de la F.T. sous forme polynomiale.

Question : Saisissez les fonctions de transfert continue $F_1(p)$, $F_2(p)$ et $F_3(p)$ sous forme polynomiale.

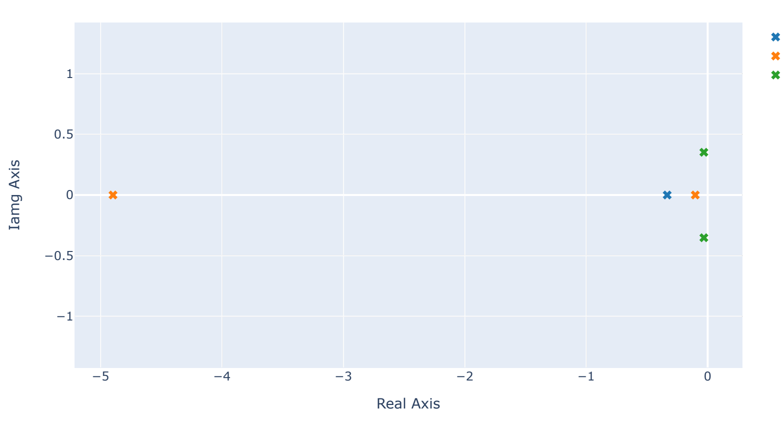
In [2]:
```
f1 = tf([2],[3,1])
f2 = tf([2],[2,10,1])
f3 = tf([2],[8,0.5,1])

sys_list = [f1,f2,f3]
```

Question : Question : Affichez les pôles et les zeros des 3 fonctions de transfert en utilisant `figure("pzmap")`

In [3]:
```
fig = figure("pzmap")
for indice in range(len(sys_list)):
    label = "sys_{}".format(indice)
    fig.plot(sys_list[indice],label=label)

fig.show()
```



2.1 Analyse temporelle

Réponse à une impulsion

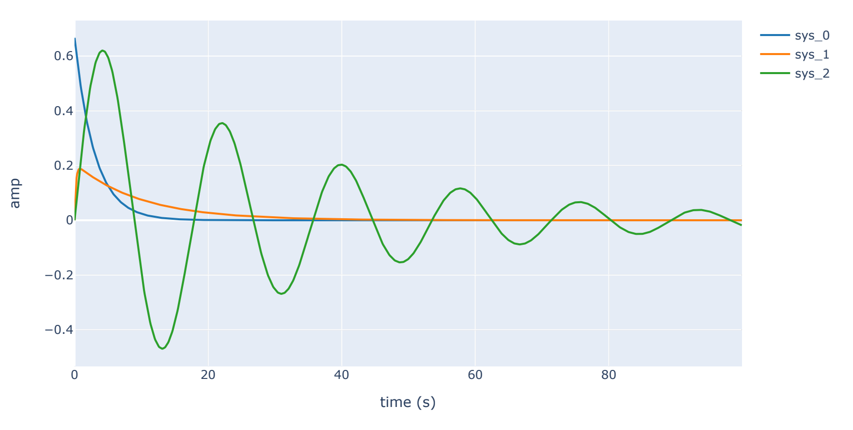
On appelle réponse impulsionnelle, la réponse temporelle à une impulsion en entrée pour un système linéaire initialement au repos, elle permet de déterminer expérimentalement la stabilité du système. Un système linéaire est stable si la réponse impulsionnelle f(t) tend vers 0 lorsque t tend vers l'infini. L'étude théorique explique qu'un système linéaire est stable si tous ses pôles (racines du dénominateur) sont à partie réelle négative.

Question :

- Tracez la réponse impulsionnelle de $F_1(p)$ via la fonction `figure("time")`
- Retrouvez par le calcul l'expression théorique de la réponse impulsionnelle.

In [4]:
```
fig = figure("time")
T = np.arange(0,100,0.1)
for indice in range(len(sys_list)):
    label = "sys_{}".format(indice)
    fig.plot(sys_list[indice],T=T,type="impulse",label=label)

fig.show()
```



Réponse à un échelon

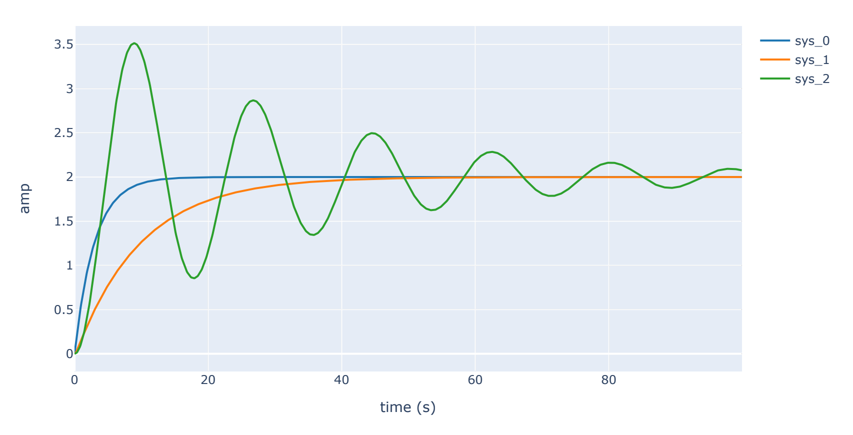
On appelle réponse indicielle, la réponse temporelle à un échelon en entrée pour un système linéaire initialement au repos, elle permet de déterminer expérimentalement la rapidité du système.

Question :

- Tracez la réponse indicielle des différentes fonction de transfert.
- Retrouvez par le calcul l'expression théorique de la réponse indicielle.

In [5]:
```
fig = figure("time")
T = np.arange(0,100,0.1)
for indice in range(len(sys_list)):
    label = "sys_{}".format(indice)
    fig.plot(sys_list[indice],T=T,label=label)

fig.show()
```



Question :

Déterminez, en utilisant la fonction `stepinfo`, le temps de réponse à 5%, le temps de montée à 90% et la valeur finale de la réponse indicielle de la fonction $F_1(p)$.

In [6]:
```
print("Systeme 1")
stepinfo(f1,display=True);
```

```
Systeme 1
RiseTime :       6.64824
SettlingTime :  8.96985
SettlingMin :   1.80377
SettlingMax :   1.99818
Overshoot :     0.00000
Undershoot :    0.00000
Peak :  1.99818
PeakTime :      21.00000
SteadyStateValue :      1.99818
```

Question :

En utilisant la fonction "damp", déterminez pour l'amortissement pour les systèmes $F_1(p)$, $F_2(p)$ et $F_3(p)$.

In [7]:
```
for indice in range(len(sys_list)):
    print("--- Système {}---".format(indice))
    damp(sys_list[indice])
```

```
--- Système 0---
poles -0.333+0.000j : wn=0.333 rad/s, m= 1.000
--- Système 1---
poles -4.898+0.000j : wn=4.898 rad/s, m= 1.000
poles -0.102+0.000j : wn=0.102 rad/s, m= 1.000
--- Système 2---
poles -0.031+0.352j : wn=0.354 rad/s, m= 0.088
poles -0.031-0.352j : wn=0.354 rad/s, m= 0.088
```

Question :

Que pouvez-vous conclure sur les pôles de ces FT à partir du coefficient d'amortissement ou du dépassement ?

2.2 Analyse Fréquentielle

La réponse harmonique ou réponse fréquentielle permet d'analyser la réponse temporelle du système à une entrée sinusoïdale en régime permanent par une lecture directe du module (coefficient d'amplification ou d'atténuation) et de la phase (avance ou retard) pour toutes les fréquences du signal d'entrée. Cette analyse permet d'évaluer les performances en régime dynamique telles que la bande passante, le filtrage, le déphasage du processus.

On rappelle que la réponse harmonique est obtenue en étudiant le gain et la phase de la fonction complexe $F(j\omega)$ correspondant à $F(p)$ où on remplace $p = j\omega$ (ω : pulsation réelle du signal d'entrée). La réponse harmonique peut être visualisée sous différentes formes : Bode, Nyquist, ou Black-Nichols en utilisant la fonction `figure()`

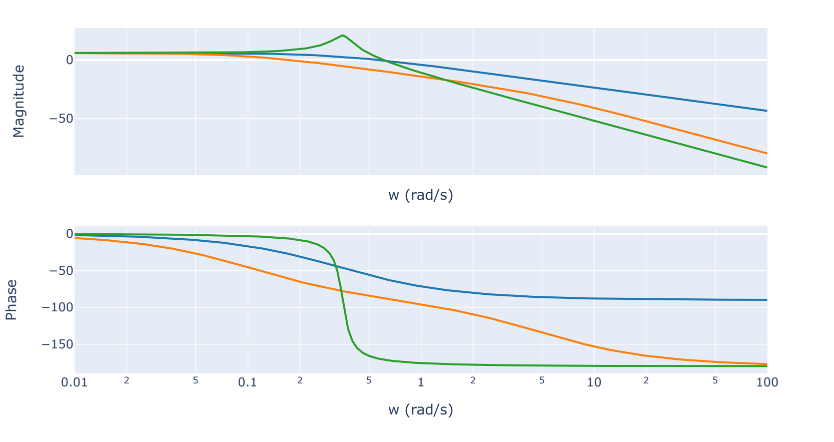
Diagramme de Bode

Visualisez les diagrammes de Bode des 3 F.T. Analysez le comportement de chacune d'elle et complétez le tableau suivant à l'aide de MATLAB

In [8]:
```
fig = figure("bode")
w = np.logspace(-2,2,1000)

for indice in range(len(sys_list)):
    label = "sys_{}".format(indice)
    fig.plot(sys_list[indice],w=w,label=label)

fig.show()
```



Question : Complétez le tableau suivant à l'aide de Python :

Fonction de Transfert	$F_1(p)$	$F_2(p)$	$F_3(p)$
Gain Statique (dB)			
Phase à l'origine (deg)			
Gain pour $\omega \to \infty$ (dB)			
Phase pour $\omega \to \infty$ (deg)			
Fréquence de résonance (Hz)			
Coefficient de résonance (dB)			
Fréquence naturelle (Hz)			
Coefficient de qualité (dB)			

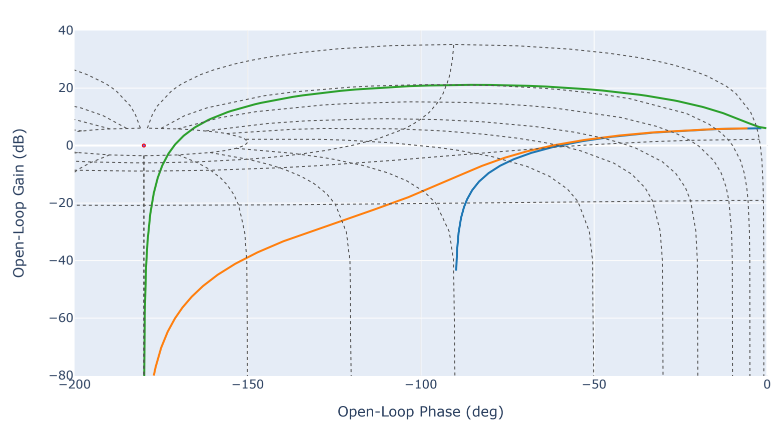
Diagramme de Black NIchols

Question : Visualisez les diagrammes de Black des 3 F.T. Retrouvez les caractéristiques du tableau précédent.

In [9]:
```
fig = figure("nichols")
w = np.logspace(-2,2,1000)

for indice in range(len(sys_list)):
    label = "sys_{}".format(indice)
    fig.plot(sys_list[indice],w=w,label=label)

fig.grid(cm=np.array([6,0,-5,-20]))
fig.xlim([-200,0])
fig.ylim([-80,40])
fig.show()
```



2. Correction d'un système du 2nd ordre.

2.1 Analyse du système en boucle fermée

On considère un système du second ordre continu incorporé dans une boucle d'asservissement avec un correcteur proportionnel Kc.

avec la FT.

$$F(p) = \frac{8}{(1+0.1p)(1+0.01p)}$$

Analyse temporelle

Question :

- Déterminez la fonction de transfert en boucle fermée (FTBF) $H(p)$ avec la fonction `feedback` pour un retour unitaire et pour $Kc = 1$.
- Vérifiez que vous retrouvez H(p) théoriquement

In [10]:
```
sys1 = tf([8],[0.1,1])
sys2 = tf([1],[0.01,1])
ftbo = sys1*sys2
Kc = 1
ftbf = feedback(Kc*ftbo,1)

print(ftbf)
```

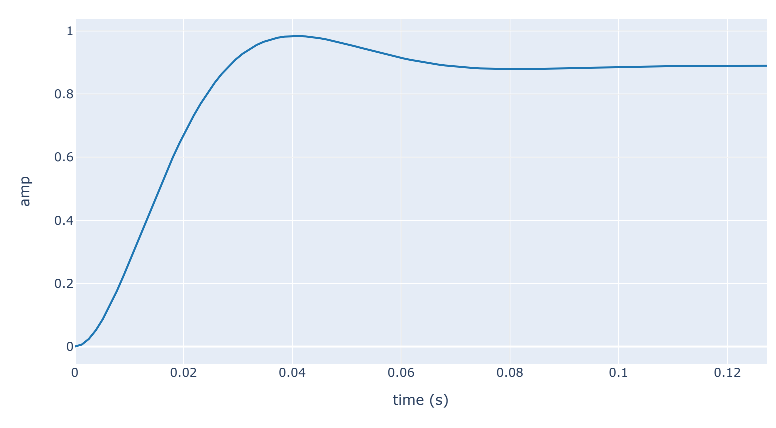
```
          8
----------------------
0.001 s^2 + 0.11 s + 9
```

Question :

- Simulez la réponse en boucle fermée à un échelon unité puis complétez le tableau suivant :

$H(p)$	t_r	m	$D\%$	$s(\infty)$
???	???	???	???	???

In [11]:
```
fig = figure("time")
fig.plot(ftbf)
fig.show()
```



In [12]:
```
print("FTBF")
stepinfo(ftbf,display=True)
damp(ftbf);
```

```
FTBF
RiseTime :       0.01855
SettlingTime :  0.05564
SettlingMin :   0.80129
SettlingMax :   0.98393
Overshoot :     10.57044
Undershoot :    0.00000
Peak :  0.98393
PeakTime :      0.04093
SteadyStateValue :      0.88986
poles -55.000+77.298j : wn=94.868 rad/s, m= 0.580
poles -55.000-77.298j : wn=94.868 rad/s, m= 0.580
```

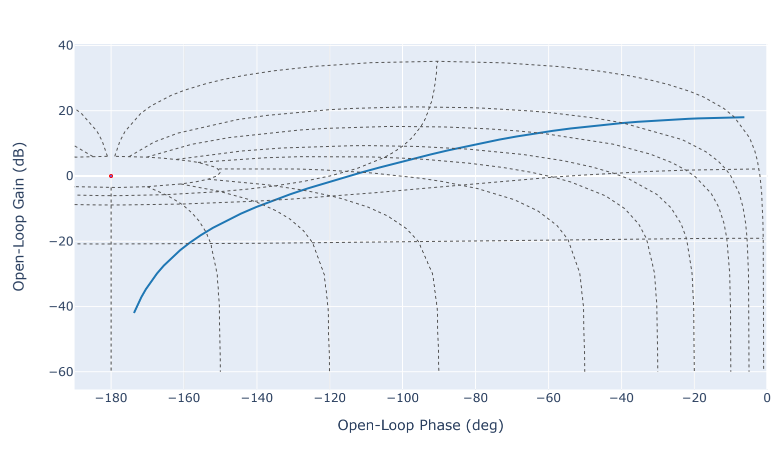
Réponse fréquentielle de H(p), (Kc=1) : diagramme de Black-Nichols

Question :

- Tracez la réponse fréquentielle dans le plan de Black-Nichols
- Complétez le tableau suivant pour la FTBO $F(p)$ et la FTBF $H(p)$: (vérifiez que vous obtenez pour $F(p)$ les mêmes résultats que lors de l'analyse de Bode)

	ω (rad/s)	0	5	10	20	30	50	100	500	1000	∞
$F(p)$	gain (dB)										
$F(p)$	Phase (deg)										
$H(p)$	gain (dB)										
$H(p)$	Phase (deg)										

In [13]:
```
fig = figure("nichols")
fig.plot(ftbo)
fig.grid(cm=np.array([6,0,-5,-20]))
fig.xlim([-190,0])
fig.show()
```



Stabilité de $H(p)$ (Kc=1)

Question :

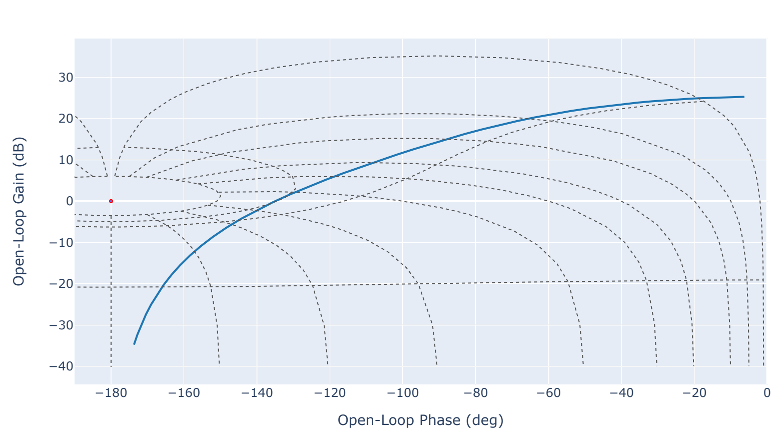
- Énoncez le critère de stabilité dans ce lieu.
- Concluez quant à la stabilité de $H(p)$ à partir du lieu de Black-Nichols.

2.2 Correction proportionnel du système à asservir

Question :

- Déterminez par essais successifs le gain Kc à introduire dans la chaîne pour obtenir un comportement de la fonction de transfert en boucle fermée comparable à celui d'un système du 2ème ordre de facteur d'amortissement $m = 0.4$.
- Déterminez les caractéristiques temporelles de la fonction de transfert corrigée et bouclée par le gain Kc.

In [14]:
```
#m=0.4 -> M = 2.69dB
Kc = 2.3
fig = figure("nichols")
fig.plot(Kc*ftbo)
fig.grid(cm=np.array([6,2.2,-0.5,-20]))
fig.xlim([-190,0])
fig.show()
```



In [15]:
```
ftbf2 = feedback(Kc*ftbo,1)
stepinfo(ftbf2,display=True);
```

```
RiseTime :       0.01023
SettlingTime :  0.05564
SettlingMin :   0.86498
SettlingMax :   1.19412
Overshoot :     25.77845
Undershoot :    0.00000
Peak :  1.19412
PeakTime :      0.02430
SteadyStateValue :      0.94938
```

In [ ]: